

E-Prime **Extensions for** **Net Station**

User Manual

For Research and Education Only

*Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone 412.449.0078
Fax: 412-449-0079
E-mail: info@pstnet.com*

E-Prime Extensions for Net Station User Manual
PST-101838
Rev 1

Copyright

Copyright 2018 Psychology Software Tools, Inc. All Rights Reserved.

The information in this document is subject to change without notice. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced, or distributed in any form or by any means, or stored in a database or retrieval system, without prior written permission of Psychology Software Tools, Inc.

Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone: 412-449-0078
Fax: 412-449-0079
E-mail: info@pstnet.com
Web: www.pstnet.com

For questions or comments regarding this manual or installation assistance:
Please e-mail us at support@pstnet.com or visit us at <https://support.pstnet.com>

Software Notice: The enclosed software is provided for use by a single user who has purchased the software. The software MAY NOT be reproduced or distributed to others. Unauthorized reproduction and/or sales of the enclosed software may result in criminal and civil prosecution (17 USC 506).

Trademark

Psychology Software Tools, Inc. (PST), the Psychology Software Tools, Inc. logo, E-Prime® Extensions for Net Station, the E-Prime® Extensions for Net Station logo, E-Prime® and the E-Prime® logo are trademarks or registered trademarks of Psychology Software Tools, Inc. Net Station™ is a trademark of Electrical Geodesics, Inc. (EGI). Windows® and Excel® are registered trademarks of Microsoft Corporation in the United States and other countries. Macintosh and Mac are trademarks of Apple Inc., registered in the United States and other countries.

*This manual describes the installation procedure for the E-Prime Extensions for Net Station.
Please review the manual completely and thoroughly before beginning the system installation.*

E-Prime Extensions for Net Station Software is for research and educational purposes only.

Table of Contents

1. Getting Started	6
1.1. Product Overview.....	6
1.2. System Requirements.....	6
1.3. Network Configuration	7
1.4. Product Service and Support	9
1.5. Resources.....	10
2. Installing E-Prime Extensions for Net Station	11
2.1. Before Installation	11
2.2. During Installation	12
2.3. After Installation	15
3. How to Create a Net Station Experiment.....	21
3.1. Overview.....	21
3.2. Experiment Design.....	21
3.3. Tutorial 1: Adding Net Station Support to an E-Prime Experiment.....	25
3.4. Tutorial 2: Adding User Defined Keys.....	53
3.5. Tutorial 3: Adding a Practice Block	61
3.6. Tutorial 4: Timing Test	76
4. Troubleshooting.....	80
4.1. Cannot locate E-Prime Samples and Tutorials	80
4.2. Missing Parameter Errors	81
4.3. Running an Experiment Without Net Station Connected	82
4.4. Net Station Communication Errors.....	83
4.5. Net Station Non-Communication Errors.....	85
5. Appendix A: Samples and Tutorials	87
6. Contact Information	88

1. Getting Started

1.1. Product Overview

E-Prime Extensions for Net Station (EENS) is a set of software routines that allow communication between Net Station and E-Prime during the run of an experiment. It allows you to create E-Prime experiments or update existing E-Prime experiments to function with Net Station. The extensions also include a set of sample experiments that can be run directly or used as a basis from which to create new experiments. This manual is intended for use with the TCP/IP (Ethernet) and NTP based Experimental Control Interface methods. If you have issues regarding the legacy serial communications or earlier synchronization methods, please contact EGI's [Knowledge Center](#).

1.2. System Requirements

E-Prime Extensions for Net Station is compatible with **E-Prime 3.0 only**. E-Prime 2.0 is not supported with the current software. The E-Prime 3.0 file extension is .es3. This manual uses the .es3 extension. Prior to the E-Prime Extensions for Net Station installation, you need to determine which version of E-Prime you currently have on your machine.

NOTE: *If permitted by your research institution, we recommend updating to the latest version of E-Prime for new features and bug fixes.*

Software Compatibility Requirements

E-Prime 3.0 and E-Prime Extensions for Net Station are compatible with machines running Microsoft® Windows® 7 SP1, 8, 10.

The remainder of this document shows images of E-Prime 3.0 running on a Windows 10 system. Your screen images may vary slightly depending on operating system and E-Prime version.

Computer Requirements for the machine running E-Prime

Additionally, the computer must comply with the minimum E-Prime 3.0 requirements:

[E-Prime 3.0 Requirements](#)

NOTE: *For further computer requirements for your Net Station software, please refer to your Net Station documentation.*

1.3. Network Configuration

TCP/IP Configuration

This document covers the configuration of EGI's NA400 amplifier. For information on configuring the NA300 amplifier, contact EGI's [Knowledge Center](#). Interactions between E-Prime and Net Station use TCP/IP communications. There are two separate components that require distinct connections to Net Station software: ECI and NTP. The Experimental Control Interface (ECI) uses TCP/IP to control Net Station recording operations and to send event information to Net Station. This connection is between E-Prime and Net Station directly. Network Time Protocol (NTP) is an industry standard mechanism for communicating clock and timing information between computers or devices. Net Station software employs this mechanism to provide timing information to E-Prime so that events in E-Prime are synchronized with the EEG data collected by EGI's amplifiers.

To configure ECI, set the IP address in the EENS Init PackageCall as described in the Tutorials in this manual (refer to [Task 6: Add the NSInit PackageCall to Initialize the System](#)). To configure NTP, you can edit the file SNTPClockExtension.ini by using the Timing tab of the Experiment Object Properties dialog.

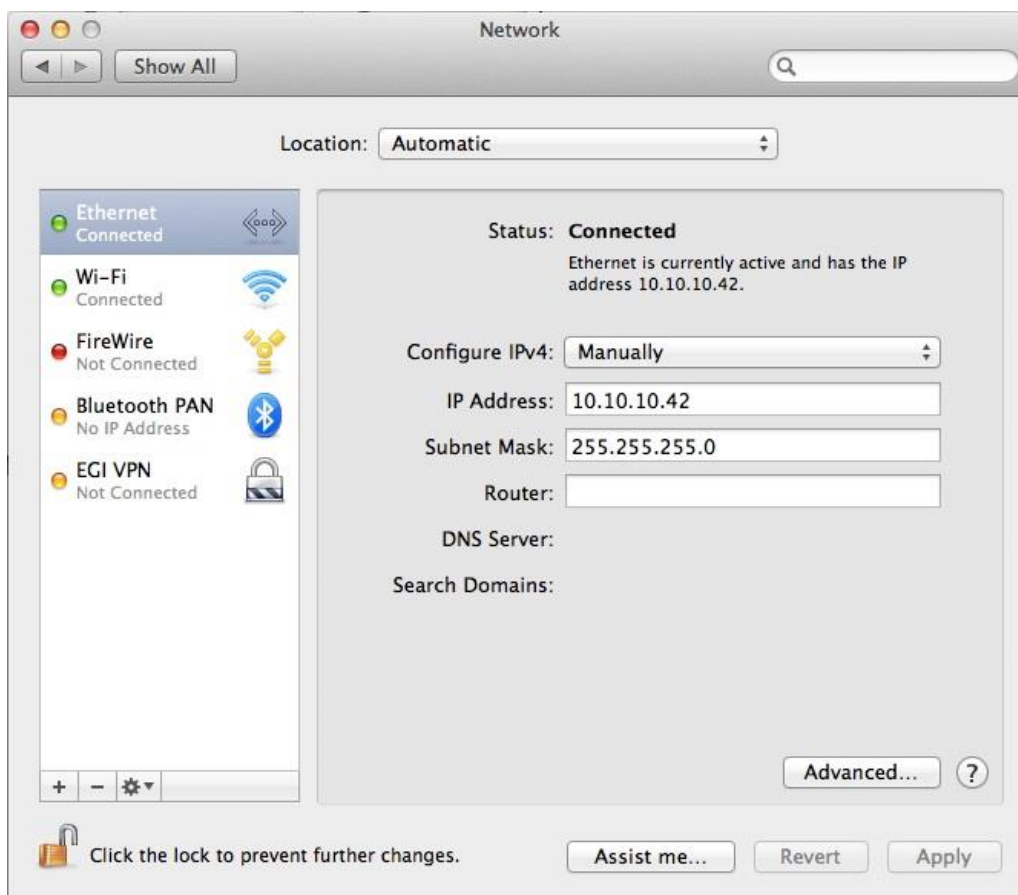
NA400 NTP Configuration:

The NA400 uses Ethernet to communicate data that is collected by the Amp to Net Station running on a Macintosh computer. The system clock on the amplifier provides the NTP service to E-Prime. An Ethernet switch is used to connect the Net Station computer, the E-Prime computer, and the NA400 amplifier together in a local network. This local network should be isolated from any other networks or the internet. To connect the system to external networks for file transfer or support purposes, use the second Ethernet port on the Mac desktop system or the wireless connection on the Mac laptop.

NA400 ECI Configuration:

For the NA400, follow these steps to set a static IP address on the Mac (OS 10.8):

- 1) **Launch System Preferences** from the Apple menu.
- 2) **Click** on the **Network** icon.
- 3) To create a unique network location, **select** the **location** or **create** a **new location** from the “**Location:**” pop up menu or **edit** the **default settings** for the location “**Automatic**”.
- 4) **Select** “**Ethernet**”, from the left panel. If there is more than one Ethernet option, **select** the **one** for the **Ethernet port** to which you have **connected** the **E-Prime computer**.
- 5) **Select** “**Manually**” from the “**Configure IPv4:**” Pop up menu to the right.
- 6) **Type** your **Mac's static IP address** into the “IP Address text field.
For the NA400, the factory IP setting is 10.10.10.42.
- 7) **Type** **255.255.255.0** into the **subnet mask field text box**.



Set a static IP address on your PC (Windows):

- 1) ***Open*** the **Local Area Connection Status** control panel by going to **Start > Control Panel > Network and Sharing Center > Local Area Connection**.
- 2) ***Click*** the **Properties** button to ***open*** the **Local Area Connection Properties** panel.
- 3) ***Select*** “**Internet Protocol (TCP/IPv4)**” from the list, and ***click*** the **Properties** button to ***open*** the **Internet Protocol (TCP/IP) Properties** panel.
- 4) ***Select*** the “**Use the following IP Address**” radio button.
- 5) ***Type*** your **PC’s static IP address** (e.g., 10.10.10.43) into the “**IP address**” text field.
- 6) ***Type*** **255.255.255.0** into the **subnet mask text field**.
- 7) ***Click*** the **OK** button to ***save*** your selection, and ***close*** the **Local Area Connection Properties** panel.

1.4. Product Service and Support

If you have difficulty running an experiment, take note of any error messages and try some of the suggestions in the [Troubleshooting](#) section. If you still have issues, please contact PST's Product Service and Support.

Technical Support

Psychology Software Tools, Inc. provides technical support for E-Prime via the PST Product Service and Support web site. To receive technical support, you must register online at:

<https://support.pstnet.com/>

Registration requires a valid E-Prime serial number, an E-Prime Extensions for Net Station serial number, and an e-mail address. A Knowledge Base is available on our Product Service and Support web site with release notes and frequently asked questions. The site includes E-Prime sample paradigms available for download, How-To videos, and the E-Prime 3.0 Experiment Library.

Our Product Service and Support web site offers an additional support forum you can use to post general questions about E-Prime and E-Prime Extensions for Net Station:

[Psychology Software Tools Community Forum](#)

Additionally, Psychology Software Tools has a YouTube channel with various Live Stream Webinars and How-To videos:

[PSTNET on YouTube](#)

NOTE: *This document contains links to our online Knowledge Base which has articles for E-Prime Extensions for Net Station. If you have trouble opening the link, please search the article number on our Product Service and Support web site.*

1.5. Resources

EGI Resources

- EGI provides technical support for EGI related software:

[EGI Knowledge Center](#)

- EGI product pages:

[EGI Products](#)

Psychology Software Tools Resources

- E-Prime Extensions for Net Station product page:

[EENS Product Page](#)

- E-Prime Extensions for Net Station download link:

[EENS Download Link](#)

NOTE: *You must be a registered E-Prime and E-Prime Extensions for Net Station user to access downloads.*

2. Installing E-Prime Extensions for Net Station

2.1. Before Installation

E-Prime Extensions for Net Station 3.0 is compatible **with E-Prime 3.0 only**. Prior to installing, determine which version of E-Prime exists on your machine.

For installation, you need either the EENS 3.0 USB Flash Drive or the installation download that we host on our web site. If using the USB Flash Drive, navigate in Windows Explorer, and run the E-Prime Extensions for Net Station 3.0 Setup.exe file. If using an installation download provided by our web site, navigate to the Psychology Software Tools Support page and click on the download link (See [Chapter 1: Section 1.5: Resources](#)).

After you have determined if the correct version of E-Prime is installed on your machine, you can install E-Prime Extensions for Net Station 3.0. Before continuing, be sure that you have administrative rights to install the software on your machine. The software cannot be installed without administrative rights. If you are unaware of your rights, please contact your system administrator.

If during the installation process you encounter issues that prohibit installation, contact your system administrator or contact PST's Product Service and Support at:

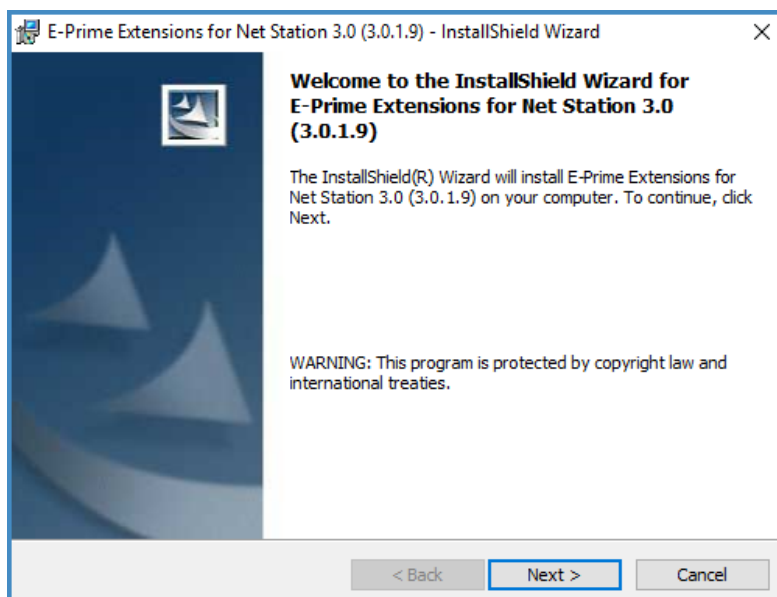
<https://support.pstnet.com/>

NOTE: Please retain the packaging for your EENS 3.0 Installation; it contains a label with your serial number which is required for installation. Additionally, register your serial number on our Product Service and Support web site. Once your serial number is registered you can access our Downloads section of our web site.

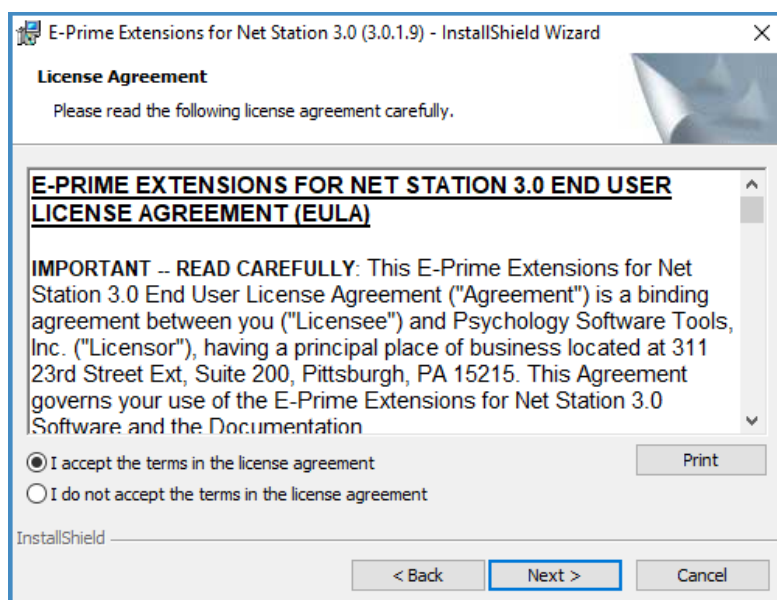
2.2. During Installation

NOTE: The version number on the following images may not correspond to the version you are installing.

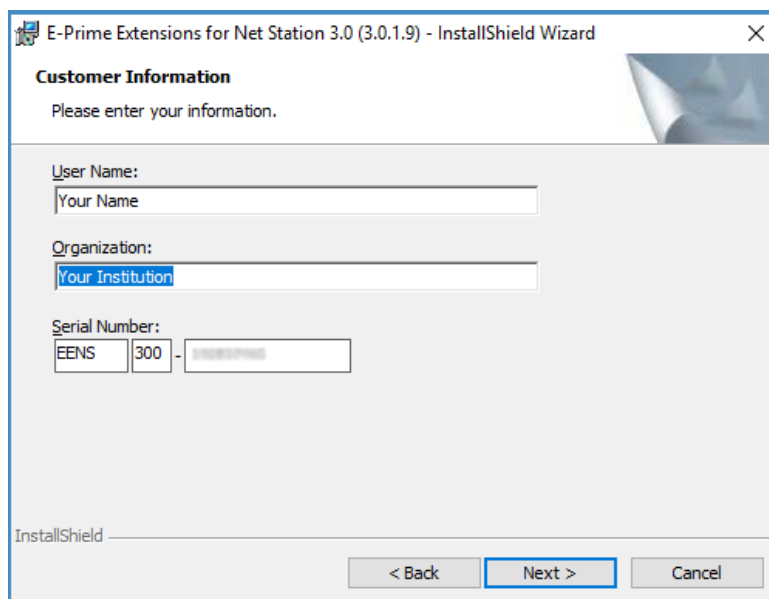
- 1) **Insert** the **E-Prime Extensions for Net Station 3.0 USB Flash Drive** into your USB port. If the Setup program does not automatically start, **navigate** to your **USB drive** in Windows Explorer and **double click Setup.exe** to launch the installer. **Click Next** to continue after the InstallShield Wizard begins.



- 2) **Read**, and if you agree, **Accept** the **Terms and Conditions**. **Click Next** to continue.



3) **Specify** your **User Name**, **Institution**, and **Serial Number**. **Click Next** to continue.



E-Prime Extensions for Net Station 3.0 (3.0.1.9) - InstallShield Wizard

Customer Information

Please enter your information.

User Name:
Your Name

Organization:
Your Institution

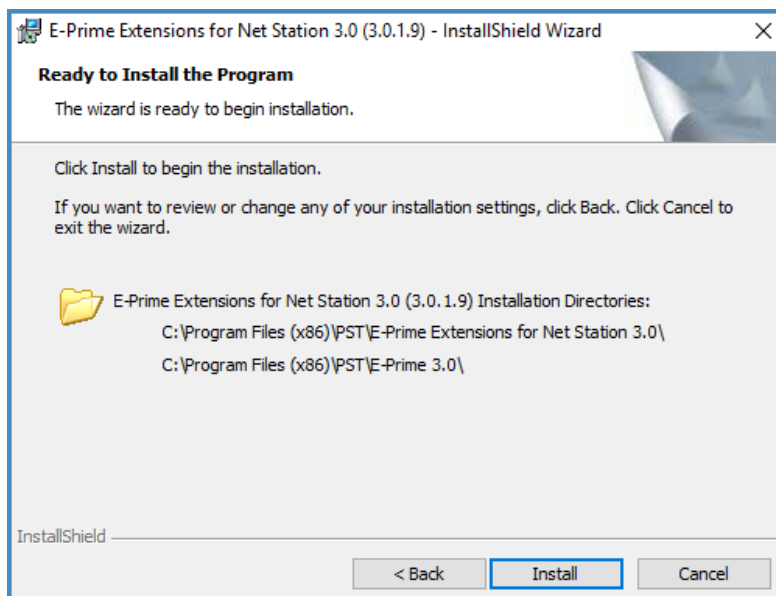
Serial Number:
EENS 300 -

InstallShield

< Back Next > Cancel

NOTE: Please retain the packaging for your EENS 3.0 Installation; it contains a label with your serial number which is required for installation.

4) **Review** the installation **folder**. **Click Install** to continue.



E-Prime Extensions for Net Station 3.0 (3.0.1.9) - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

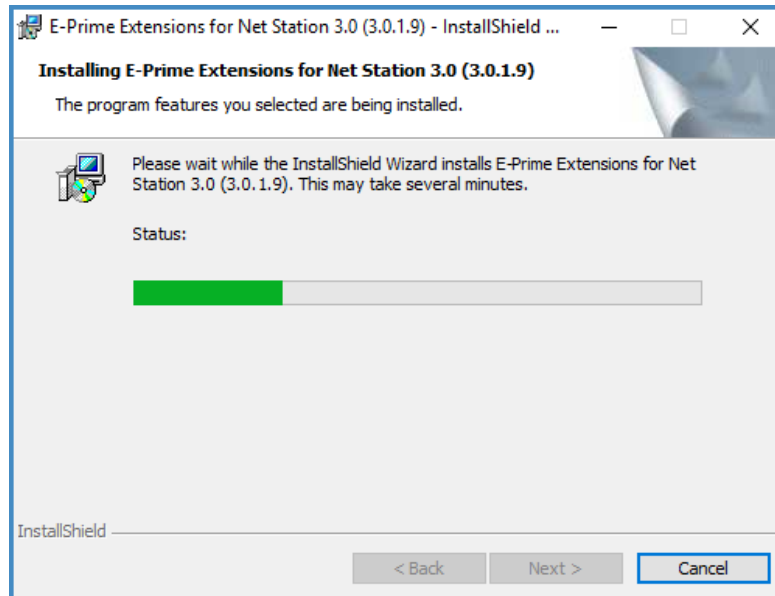
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

E-Prime Extensions for Net Station 3.0 (3.0.1.9) Installation Directories:
C:\Program Files (x86)\PST\E-Prime Extensions for Net Station 3.0\
C:\Program Files (x86)\PST\E-Prime 3.0\

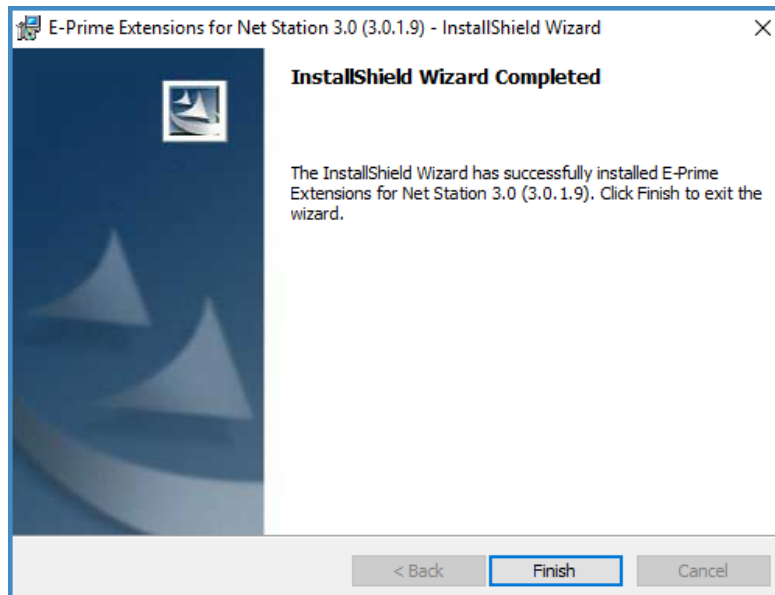
InstallShield

< Back Install Cancel

5) **Wait** while the software installs.



6) **Wait** for the confirmation message that the software installs successfully.
Click Finish to complete the installation process.



2.3. After Installation

Finding the My Experiment Folder

E-Prime 3.0 is compatible with machines running Microsoft® Windows® 7 SP1, 8, 10. The “**My Experiments**” folder is frequently used and is the default location to store new experiments created with E-Studio. E-Prime creates the “My Experiments” folder in your personal documents folder on your PC. This folder also contains the “**Samples**” and “**Tutorials**” folders which stores the experiments discussed in [Tutorial 1: Adding Net Station Support to an E-Prime Experiment](#).

While the E-Prime documentation directs you to the “My Experiments” folder, it does not include the full path to the folder. Instead, the documentation refers to “My Experiments”, where the “...” indicates the full path up to your personal documents folder. When you see this notation in the documentation, replace the “...” with the path to your personal documents folder.

NOTE: *The path on your machine may have been modified by your administrator.*

The table below shows the default paths to your personal documents folder:

Folder	Path to your personal documents folder (“My Documents” or “Documents”)
*Windows 7/8/10	<drive>\Users\<user name>\Documents\My Experiments\3.0\

**Windows 7 and 8 go to “Documents”, but it is visible as “My Documents”.*

E-Prime Extensions for Net Station Samples and Tutorials

The E-Prime Extensions for Net Station 3.0 folder is located within the “My Experiments” folder, as shown below. The Net Station folder contains the Samples and Tutorials subfolders:

Folder Type	Path to Samples or Tutorials
Samples	...My Experiments\3.0\Net Station\Samples
Tutorials	...My Experiments\3.0\Net Station\Tutorials

For a listing of all installed Samples and Tutorials please refer to [Appendix A: Samples and Tutorials](#).

NOTE: *This User Manual contains Tutorials. Both the starting point for the Tutorial and a finished copy of the Tutorial are provided for you (see [Tutorial 1: Adding Net Station Support to an E-Prime Experiment](#)).*

Opening E-Studio

After E-Prime Extensions for Net Station 3.0 installs, launch E-Studio. When E-Studio launches, it detects that the E-Prime Extensions for Net Station 3.0 Sample and Tutorials files need to be updated. By specifying ‘Yes,’ you allow E-Prime to copy the new Samples and Tutorials into your ...My Experiments\3.0 folder.

NOTE: *Copying the Samples and Tutorials upon launching E-Studio overwrites your existing Samples and Tutorials. If you wish to overwrite your files at a later point in time refer to [Troubleshooting, 4.1 Cannot locate E-Prime Samples and Tutorials](#).*

Running a Test Experiment:

Once you completed the software installation and configured the TCP/IP settings on your Mac and PC computers, you should run a test experiment. We recommend using the NSBasicRT.es3 sample experiment provided with the E-Prime Extensions for Net Station 3.0 installer. This test run familiarizes you with setting up other experiments for TCP/IP ECI communications.

Launch the NSBasicRT.es3 Sample experiment:

- 1) **Launch** the **E-Studio** application from the Windows Start menu by **selecting Start > Programs > E-Prime > E-Studio**.
- 2) After **E-Studio** launches, **dismiss** the “**New E-Prime 3.0 Experiment**” template dialog.
- 3) **Use** the **Open File** dialog to **navigate** to appropriate folder, and **load** the **NSBasicRT.es3** experiment. The default file path is:

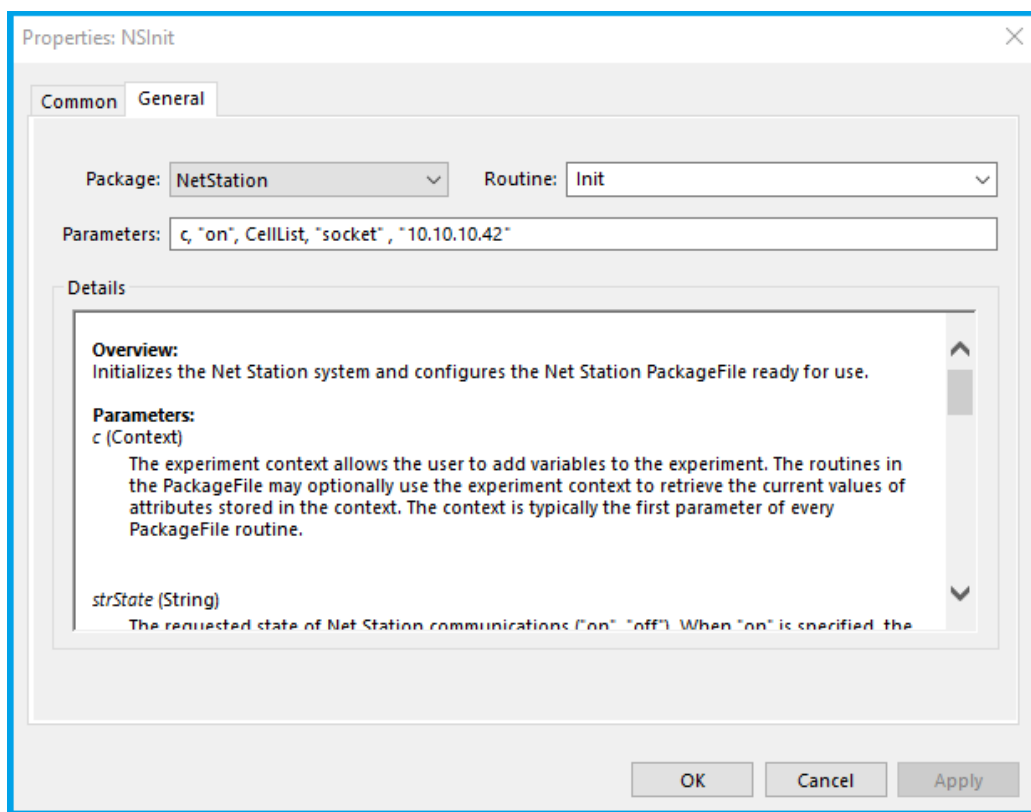
...\\My Experiments\\3.0\\Net Station\\Samples\\NSBasicRT

NOTE: Alternatively, you may also **launch** any **E-Studio** experiment by **double clicking** on the **Experiment file** in Windows Explorer.

Specify the NSInit parameters:

- 1) Once you have **loaded** the **NSBasicRT.es3** experiment, you must set it up to **communicate** using the **static IP address** of the **Net Station Mac**. **Double click** on the icon “NSInit” in the **Structure Window**.
- 2) In the “**Parameters**” text field, **add** the **static IP address** (“XXX.XXX.XXX.XXX”) of your **Net Station Mac**. For example, assuming the Mac IP address is 10.10.10.42, you would enter:

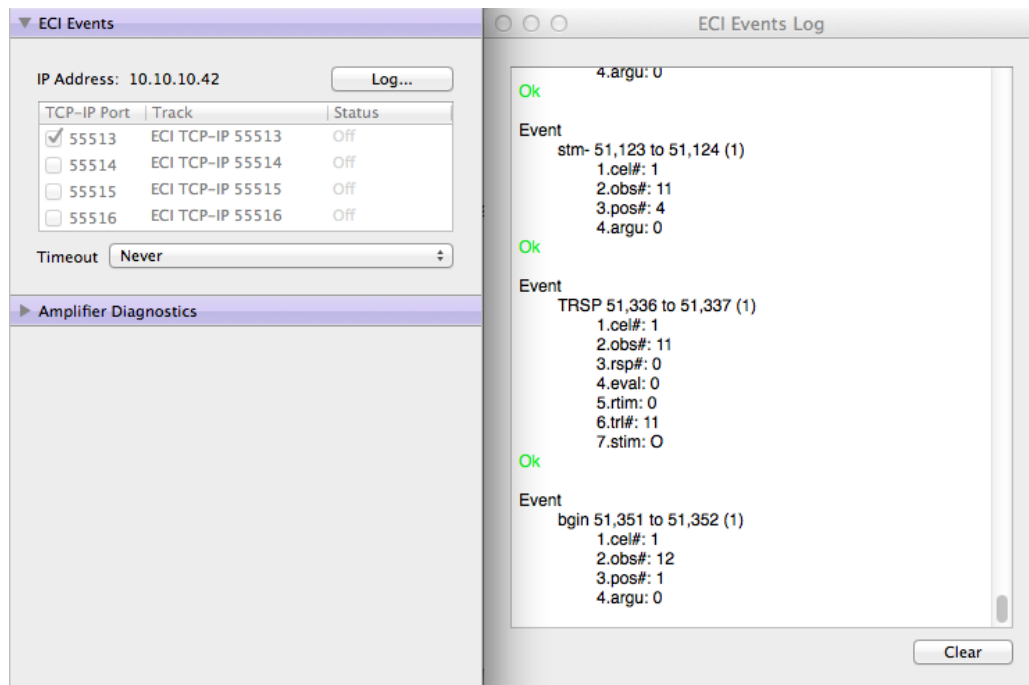
c, "on", CellList, "socket", "10.10.10.42"



- 3) **Click** the **OK** button to **save** your changes and **close** the **NSInit** Object.

Launch Net Station

- 1) **Follow** the **instructions** provided by EGI to **set up** Net Station for acquisition. Please **use** the default ECI port, **55513**, for this test run.
- 2) In the **ECI panel**, **click** the **Log** button to **open** the **Log panel**.

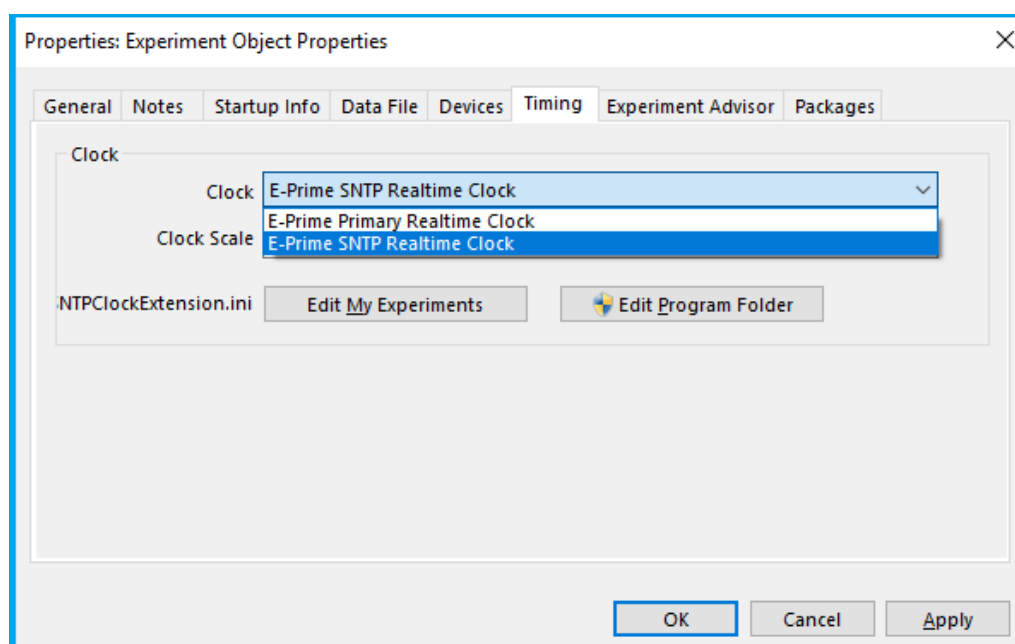


- 3) **Start acquisition** on the Mac. This starts the NTP services in Net Station that allow E-Prime to synchronize with the EGI amplifier.

Configuring the NTP Clock

The E-Prime Realtime clock is the default clock used in E-Prime. However, when running experiments with a EGI's amplifier, we recommend using the SNTP clock for timing purposes. When the SNTP Realtime Clock is selected, you must specify additional settings in a file named **SNTPClockExtension.ini** (e.g., the IP address of the NTP server). Because the default clock is not the SNTP Clock, the clock settings should be switched before running an experiment with EGI's components.

- 1) **Double click** the **Experiment Object** to open it.
- 2) **Navigate** to the **Timing** tab.
- 3) **Switch** the **Clock** to the **E-Prime SNTP Realtime Clock**.



- 4) **Edit** the **SNTPClockExtension.ini** file that is either located in the **My Experiments** or **E-Prime Program** folder.
- 5) **Save** the **SNTPClockExtension.ini** file. **Accept** all changes to the Experiment Object.

NOTE: Typically, you only need to configure the IP property of the *SNTPClockExtension.in* file to be the IP Address of the amplifier (e.g., 10.10.10.51).

Please refer to the following article for more information on the SNTP Clock:

[Article 19471: SNTP Clock Configuration Options](#)

Run the experiment:

- 1) **Click** the **Run** button (**F7**) from **E-Studio** to **generate** and **run** the **experiment**
- 2) **After** the **experiment** begins **executing**, it will present message boxes prompting you to **enter Startup Info**. **Accept** the **default values** or **provide** a **valid value** for each prompt.
- 3) **Verify** in the **experiment** that the **instructions** appear and the **trials** begin.
- 4) **Verify** that **Net Station** displays the **messages** being sent from E-Prime within the ECI Log panel and the “ECI TCP/IP 55513” Event Track. It is recommended that you keep these visible during testing for reference.

NOTE: *If you have trouble running the test experiment, refer to [Troubleshooting](#) and [Product Service and Support](#) sections at the end of this chapter before continuing.*

NOTE: *Prior to collecting any real data with E-Prime and Net Station, it is strongly recommended that you review the critical timing information contained in the [Tutorial 4: Timing Test](#).*

Alternate NSInit Parameters:

E-Prime uses default TCP port 55513 if none is specified in the “Parameters” text field. If you want to use a different port, you may specify it by entering 55514, 55515, or 55516 after the IP address, separated by a colon (“XXX.XXX.XXX.XXX:NNNNN”). For example:

c, "on", CellList, "socket", "10.10.10.42:55514"

Net Station defaults to port 55513. If you want to use a different port, you may specify it in the Configuration settings of the Net Station ECI panel. You may select multiple ports if you wish, and Net Station automatically activates the port on which E-Prime sends events during each experimental run.

3. How to Create a Net Station Experiment

3.1. Overview

This chapter illustrates the creation of an E-Prime/Net Station experiment using one of the sample paradigms provided with EENS, the NSPictureRT paradigm. Additionally, it introduces and explains some programming methods that E-Prime uses.

To complete the Tutorials, you need a computer with E-Prime and the EENS software already installed. If you have not installed E-Prime 3.0, please complete the installation before continuing. If you have not installed E-Prime Extensions for Net Station 3.0, please refer to the installation section in this manual (see [Chapter 2: Installing E-Prime Extensions for Net Station](#)).

While this document includes a basic introduction to programming concepts specific to using E-Prime with Net Station, the Tutorial assumes you are familiar with using E-Prime to build behavioral experiments. If you are new to using E-Prime, we recommend that you work through all the Tutorials provided on our Product Service and Support web site prior to beginning this Tutorial.

[Article 23282: Experiment Design \(Beginner\)](#)

[Article 22741: Experiment Design \(Advanced\)](#)

3.2. Experiment Design

Overview

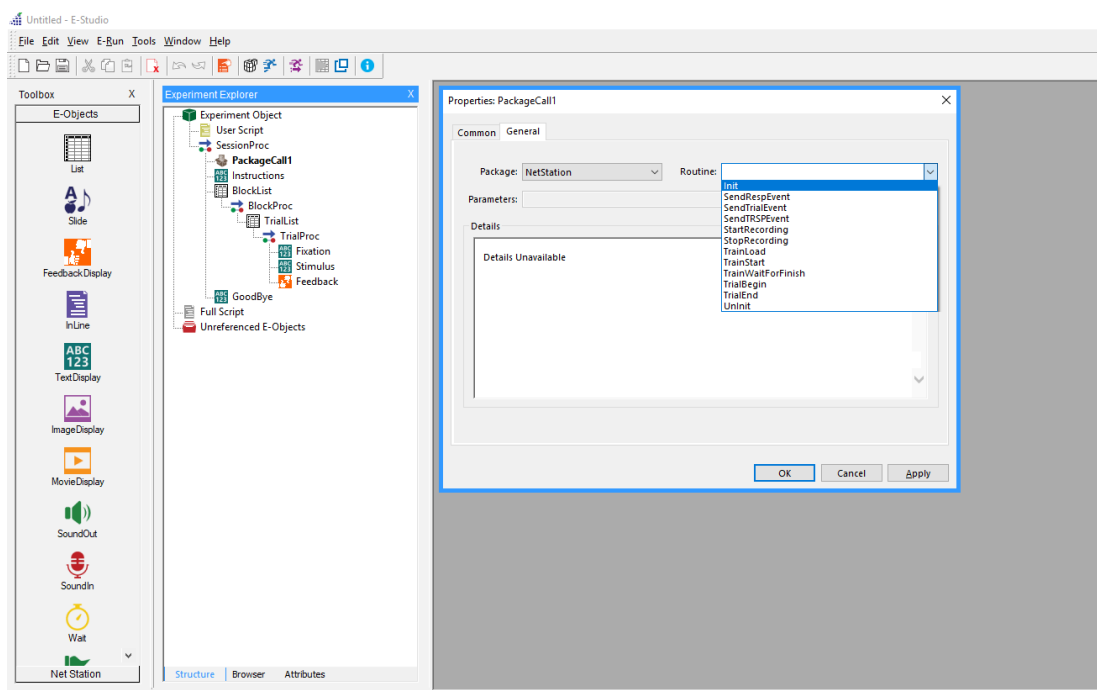
To add support for Net Station to an existing E-Prime experiment, you need to utilize the PackageFile feature of E-Prime. An E-Prime **PackageFile** is a pre-written, cohesive set of E-Basic script routines that are grouped together and distributed in a single file. The PackageFile can be authored and maintained externally and then shared amongst different E-Prime experiments. The Routines contained in the PackageFile serve to expand the default functionality offered by E-Prime for building experiments that interact with third party hardware and software products.

The E-Prime Extensions for Net Station 3.0 PackageFile is included in the E-Prime Extensions for Net Station installation. The Routines in the PackageFile allow you to automatically perform various functions (e.g., hardware initialization, starting/stopping, and configuring/sending event markers). These Routines are invoked via graphical representations in the Experiment Explorer window in E-Studio. Some of the Routines require you to set or modify parameters to enable them to work within the context of your experiment. Once a PackageFile is added to an experiment, there are several methods that can be used to access the Routines. A brief overview of each method is provided below.

PackageCall Method

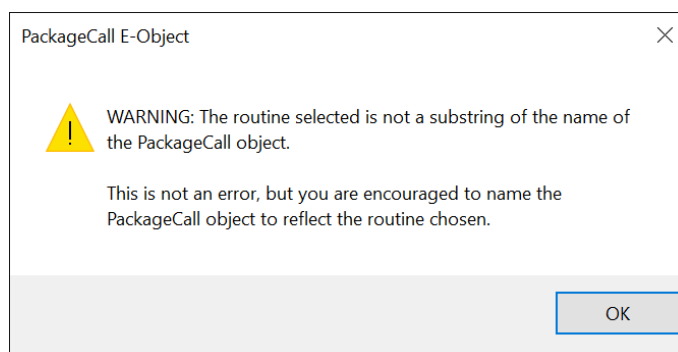
Routines contained within a PackageFile are typically called by dragging a **PackageCall Object** from the E-Studio Toolbox. The PackageCall is then dropped into a Procedure Object at the location within the experiment where the call is to be invoked. Once a PackageCall is added to the experiment, you can double click on the object to configure its properties. Minimally, you must specify which Package and Routine need to be called and what parameters need to be sent during the call.

The PackageCall Properties includes a **Details** section which provides information regarding the purpose of the Routine. A description of any required or optional parameters that need to be specified to properly use the Routine are also listed. When viewing the Details section, it is helpful to note that a **Remarks** section exists at the bottom of the window. The Remarks section offers further insight on how the PackageCall is used in the experiment.



NOTE: In some cases, you may have to scroll down to view Remarks/additional parameters.

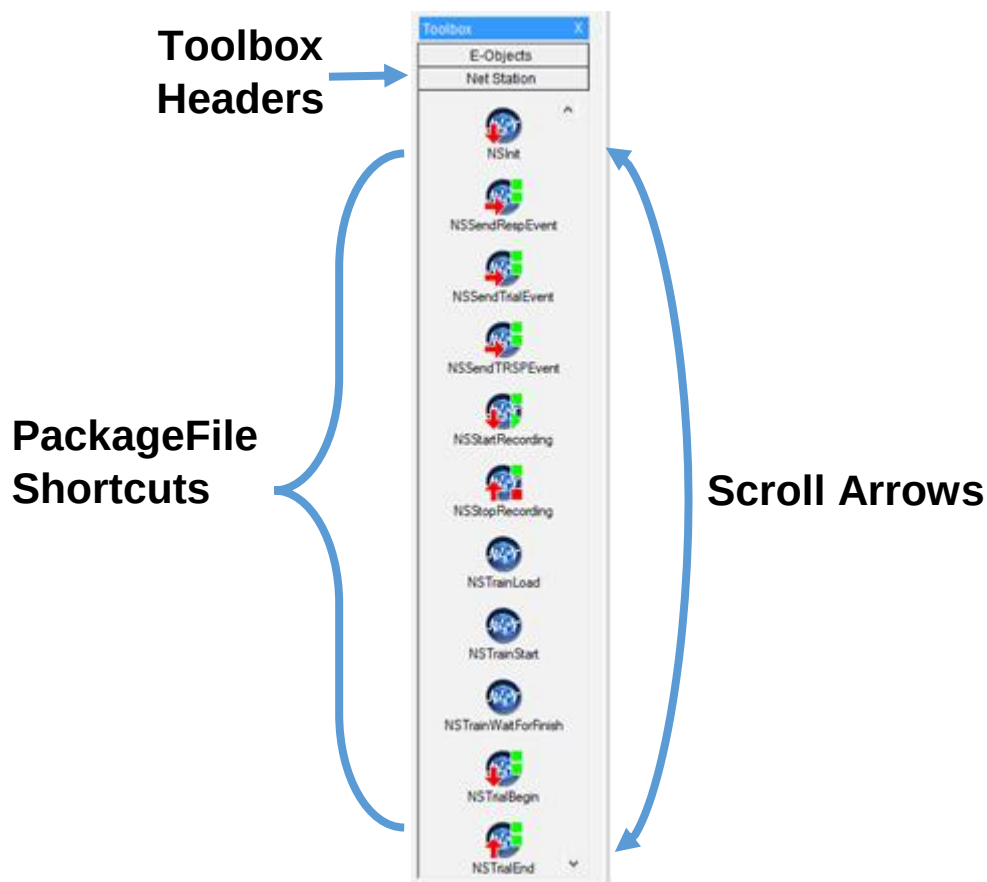
Once the Package and Routine properties of the PackageCall are set and the Properties dialog is closed, the icon associated with the PackageCall Object changes from the default icon to an author-specified icon. If there is no author-specified icon associated then the default icon for the entire PackageFile is used. When dragging and manually configuring a PackageCall, it is strongly recommended that you rename the object to reflect the specific PackageFile and Routines which are being referenced within the object. If you do not follow this recommendation, E-Prime issues the following warning:



With Routines in the Net Station PackageFile, it is a common convention to create a name for each object by abbreviating “Net Station” as “NS” and then concatenating the name of the Routine being called. For example, the “NetStation_Init” routine would be renamed “NSInit”.

PackageFile Toolbox Shortcuts

A PackageFile author can optionally choose to provide a customized Toolbox of PackageFile Shortcuts to make it faster and easier for the experiment author to access and configure the Routines supported by the PackageFile. If a custom Toolbox is included with the PackageFile, a new panel is added to the Toolbox in E-Studio after the PackageFile is added to the experiment. Each panel of the Toolbox includes a clickable header that indicates which PackageFile is associated with the panel. You can click on the header to show the contents of the Toolbox and then drag a Shortcut from the Toolbox and drop it in a Procedure where the call is to be invoked. Scroll Arrows are present on the Toolbox for scrolling through the PackageFile Shortcuts.



Each PackageFile Shortcut contained in the Toolbox is simply a PackageCall Object (as outlined above) which has been predefined by the PackageFile author to save the experimenter time in renaming each PackageCall and configuring its required Properties. It is important to note that PackageFile Shortcuts that have different names in the Toolbox may be configured to reference the same PackageFile routine (e.g., properties may only differ by the list of default parameters that are specified). Providing multiple PackageFile Shortcuts for the same Routine can be useful when a

single common PackageFile routine is used in different scenarios with the experiment. Using PackageFile Shortcuts are the primary programming method used for this Tutorial.

InLine Script Method

An **InLine** is an object used to insert user-defined script into an experiment at a specific point. The user-written script is placed at the corresponding location in the Full Script when the experiment is generated. PackageFile routines can be called on an InLine Object as well as being called by a PackageCall or PackageFile Shortcut. PST does not typically recommend this method of calling PackageFile routines; however, EENS often requires users to call multiple PackageFile routines in a sequence. [Tutorial 1, Task 12: Add script to send trial events to Net Station](#) describes how to call PackageFile routines in script for EENS and why this method of referencing PackageFile routines is recommended.

Calling a PackageFile routine from script is generally the same as calling any other E-Basic Subroutine or Function that could be defined in User Script except that the definition of the Routine is contained with the PackageFile rather than within the User Script section of your experiment.

Due to a common naming convention used by PackageFile authors, a Package Routine can be invoked directly from an InLine Object. This is done by concatenating the name of the Package followed by an underscore (“_”) and then the name of the Routine you want to call. Lastly, any parameters are added that are required for the call.

For example, the Net Station PackageFile includes a call to start recording data. The Routine is named “StartRecording” and takes the Experiment Context (“c”) as a required parameter. To call this Routine from an InLine Script Object, specify the following in the InLine Object:

PackageFile Name	Routine	Parameter
↓	↓	↓
<i>NetStation_StartRecording c</i>		

If you are making direct PackageCalls via E-Basic script please refer to:

[Article 26208: Using PackageCalls via InLine Script](#)

Please view the article mentioned above to see examples of PackageCall parameters and functions invoked by calling the Routine from an InLine.

For specific information on PackageCalls in E-Prime Extensions for Net Station 3.0, please refer to:

[E-Prime Command Reference](#)

3.3. Tutorial 1: Adding Net Station Support to an E-Prime Experiment

Summary:

Net Station-enabled experiments can be run locally from E-Studio during development and testing (i.e., with the Net Station hardware not connected). We recommend testing your experiment locally before using it on the experiment presentation computer connected to the Net Station hardware.

NOTE: *While many errors in your EENS-enabled experiment errors can be detected while testing locally, some errors can only be detected or generated when the Net Station hardware is in use. The final step in developing an experiment is to test with all related hardware and software components.*

To test your experiment without the EGI hardware and software connected, communications to EGI need turned off by changing NSInit from “on” to “off” (refer to [Task 6: Add the NSInit PackageCall to Initialize the System](#)). Also, you need to use the E-Prime Realtime clock rather than the SNTP clock when testing your E-Prime experiment without the EGI components. The E-Prime Realtime clock is the default clock used in E-Prime experiments. When testing experiments with EGI components, consider using the SNTP clock and consulting your Net Station documentation.

NOTE: *All the Tutorials, except for Tutorial 4: Timing Test, build upon each other and must be completed in order. The Sample experiment found in the folder <My Experiments>\3.0\Net Station\Samples\NSPictureRT represents the complete set of changes in Tutorials 1 through 3.*

Goal:

This Tutorial illustrates how to add the Net Station PackageCalls into the PictureRT.es3 Tutorial experiment included with E-Prime Extensions for Net Station 3.0. After this Tutorial, a basic EENS-enabled paradigm is created.

Overview of Tasks:

- Load PictureRT and resave it as NSPictureRT.es3.
- Add the Net Station PackageFile to the experiment.
- Enable the Startup Info prompts required by Net Station.
- Add Net Station PackageCalls to:
 - Initialize and Uninitialize the System.
 - Start and Stop Recording.
 - Initialize and Uninitialize each Trial.
 - Send RESP, Trial, and TRSP Events to Net Station.
- Assign unique four-character codes to each event of interest.
- Add and configure the CellList.
- Assign CellNumber to individual trials.
- Run the experiment.

Recommended readings:

Before beginning the E-Prime Extensions for Net Station Tutorial, we recommend that you have read and worked through the other Tutorials provided with E-Prime. Please refer to the following for other available Tutorials:

[Article 23282: Experiment Design \(Beginner\)](#)

[Article 22741: Experiment Design \(Advanced\)](#)

It is important to understand critical timing in E-Prime to run experiments under the best possible conditions for accurate timing data. Refer to the following articles on timing:

[Article 22867: E-Prime Timing Test Results](#)

[Article 22852: Timing of E-Objects](#)

[Article 22857: Stimulus Preparation Solutions](#)

Additionally, the clock settings in E-Prime are manipulated during this Tutorial. It is crucial to know the differences in the clocks available in E-Prime. Please refer to the following on the clock settings in E-Prime:

[Article 19471: SNTP Clock Configuration Options](#)

[Article 22863: Clock Settings](#)

Estimated Tutorial Time:

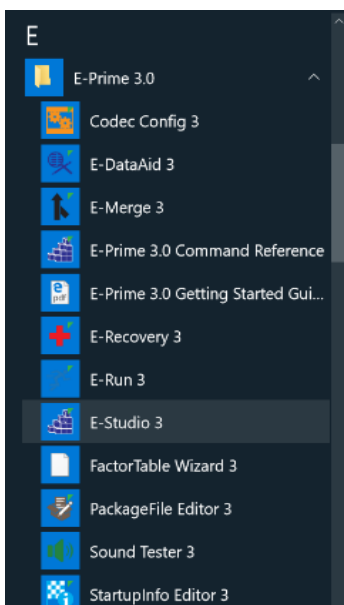
15-20 minutes

Task 1: Open the PictureRT.es3 Experiment in E-Studio

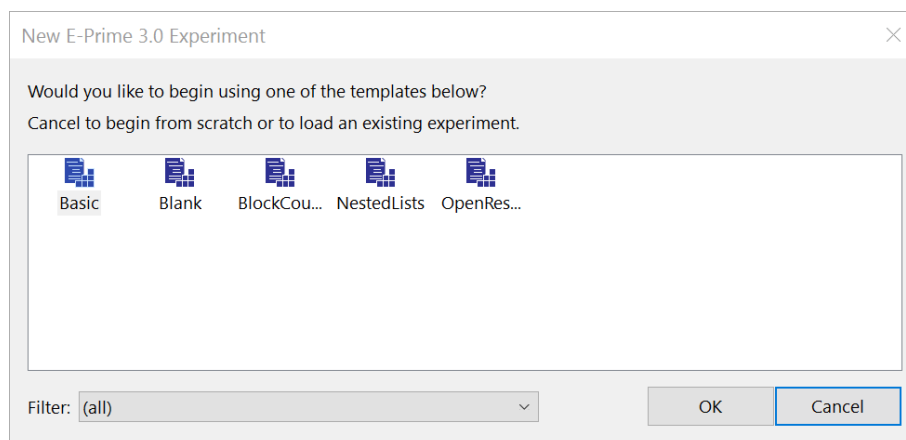
Locate the E-Studio icon in the Start > All Programs > E-Prime 3.0 menu and launch the application by selecting it. Load the PictureRT.es3 experiment.

The E-Studio application is installed as part of the typical E-Prime installation. This application is used to create, modify, and test experiments within E-Prime. Open the E-Studio application, navigate to <My Experiments>\3.0\Net Station\Tutorials, and load the PictureRT.es3 experiment.

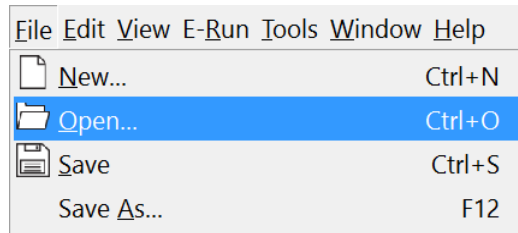
- 1) **Click** on the Windows **Start** menu, **search** for **E-Prime 3.0**. From the menu, **click** on **E-Studio** to launch the application.



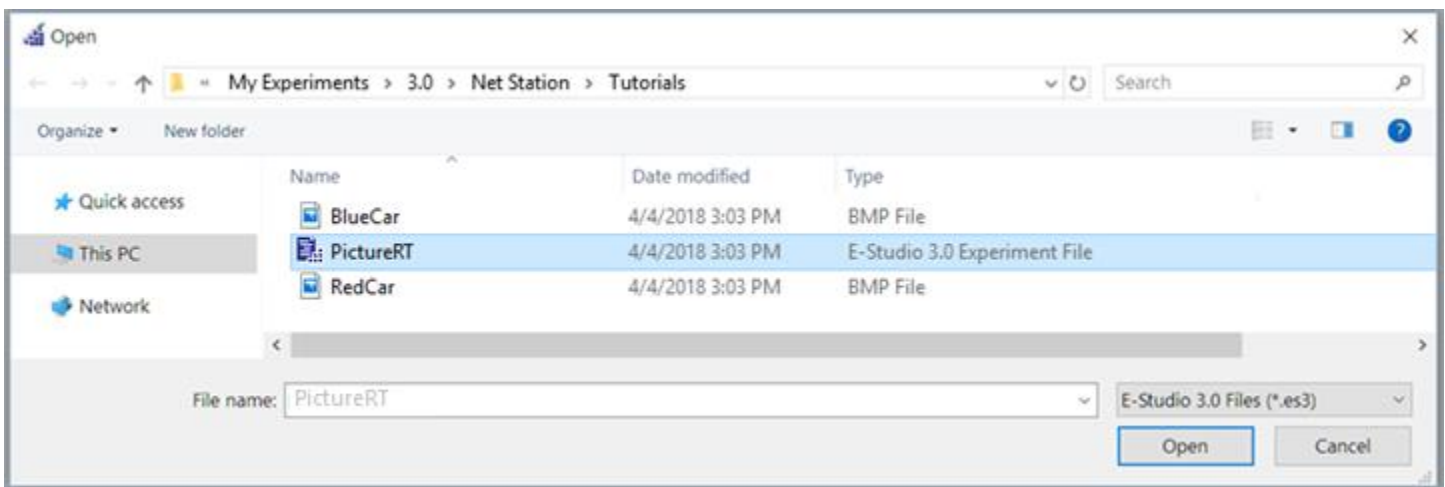
- 2) **Click** the **Cancel** button on the New E-Prime 3.0 Experiment window.



- 3) **Select Open** from the **File** menu. **Navigate** to the **<My Experiments>\3.0\Net Station\Tutorials** folder to load the experiment.



- 4) **Select** the **“PictureRT.es3”** file and then **click** the **Open** button to load the experiment into E-Studio.



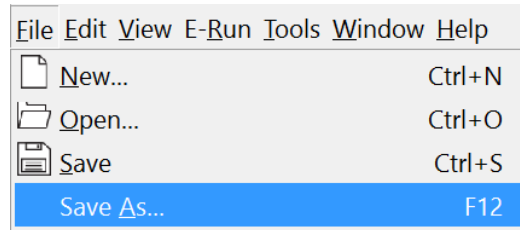
NOTE: If Samples and Tutorials for E-Prime Extensions for Net Station 3.0 do not exist, see [Troubleshooting, 4.1: Cannot locate E-Prime Samples and Tutorials](#).

Task 2: Save the experiment under a new name

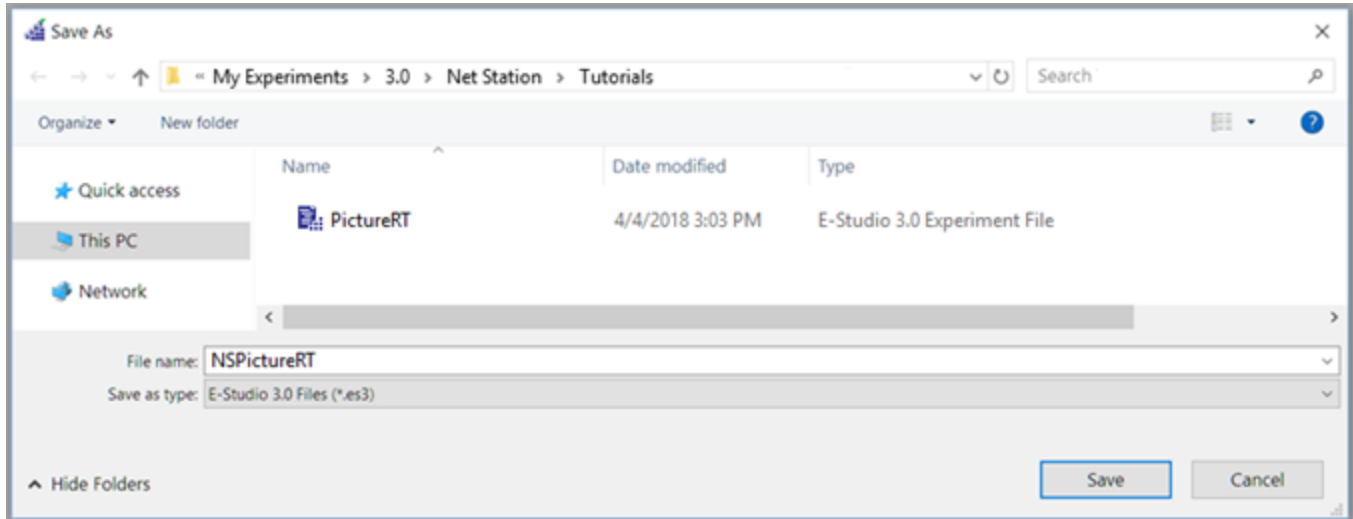
Save the *PictureRT.es3* experiment in the same folder under the new name “*NSPictureRT.es3*.”

Rename the experiment, but be sure to save it in the same folder (<My Experiments>\3.0\Net Station\Tutorials) so that any resources within the experiment remain valid and can be reused.

1) **Select Save As...** from the **File** menu.



2) **Type “NSPictureRT.es3”** as the new name in the **File name** field. **Click** the **Save** button.

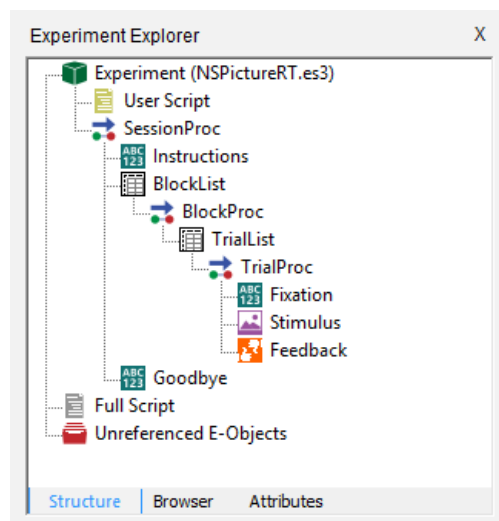


Task 3: Add the Net Station PackageFile to the Experiment

Open the Properties dialog for the Experiment Object and use the Packages tab to add the NS PackageFile to the experiment.

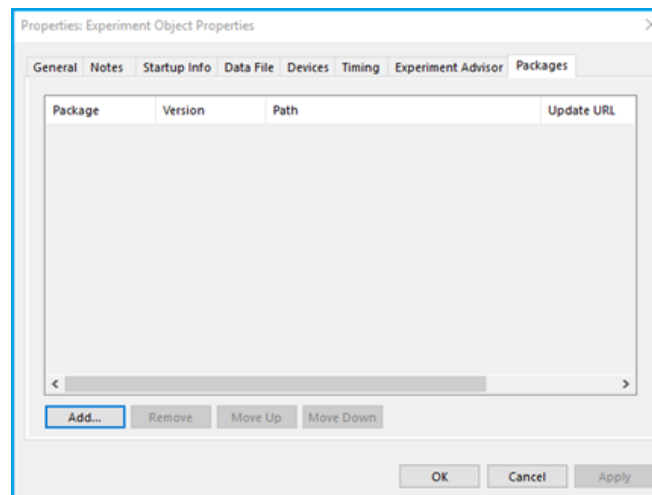
PackageFiles in E-Prime are cohesive sets of E-Basic routines that are grouped together. To gain access to the Routines within a PackageFile you must first add the PackageFile to the experiment. PackageFiles can be added to an experiment using the Packages tab of the Experiment Object Properties dialog. The Routines that are used to communicate with the Net Station hardware/software at runtime are contained within the Net Station PackageFile. The PackageFile also includes a customized Toolbox which contains shortcuts that you can drag into Procedure Objects within your experiment to invoke Routines contained in the PackageFile.

1) **Double click** the **Experiment Object** at the top of the tree in the **Experiment Explorer** window.

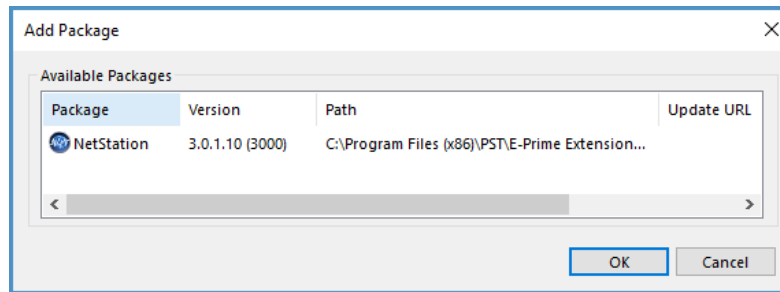


2) **Click** on the **Packages** tab of the **Experiment Object Properties** dialog.

3) **Click** the **Add...** button.

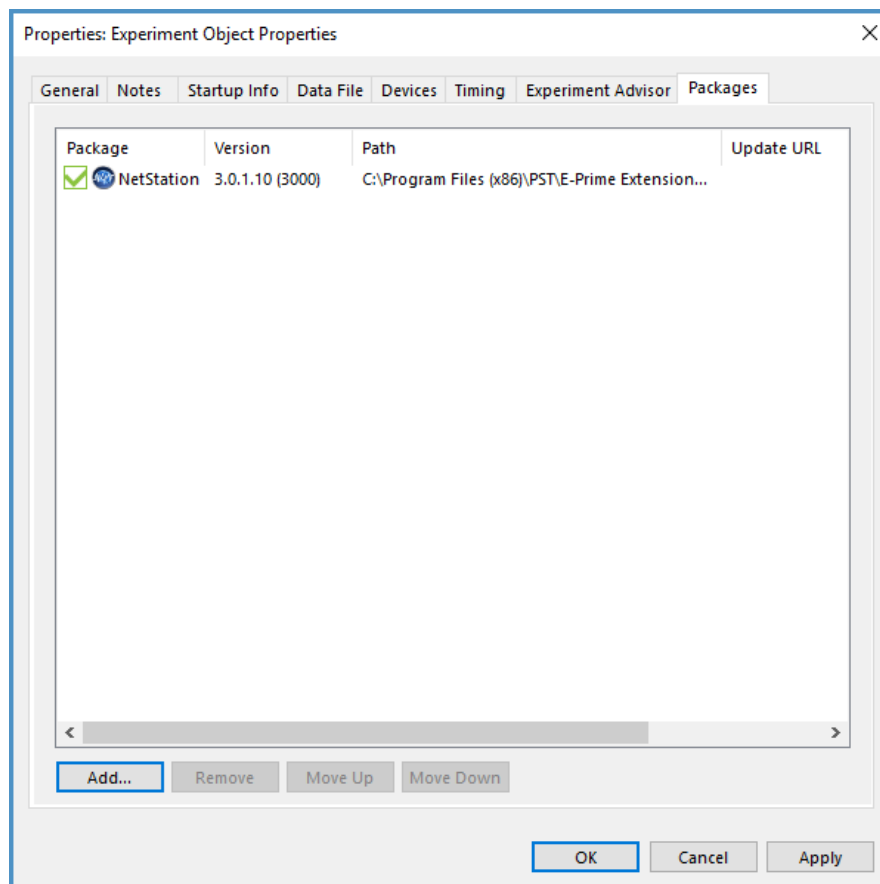


4) **Select** the **NetStation PackageFile** in the **Add Package** dialog.



5) **Click** the **OK** button to accept the dialog.

6) **Verify** the **NetStation PackageFile** has been added and is checked. (**Do not** yet **click OK** or **dismiss** the dialog).



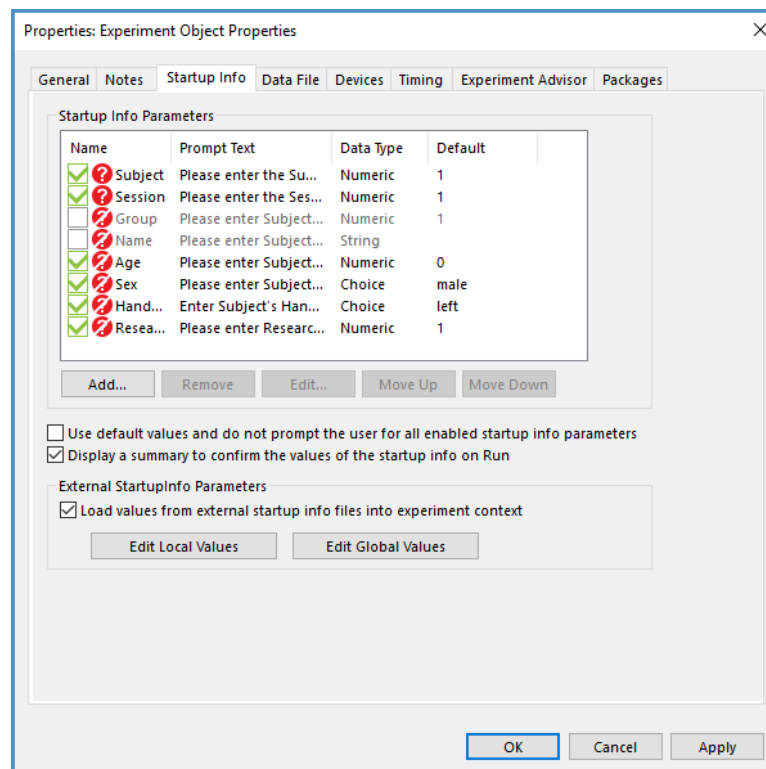
NOTE: The PackageFile version number that is displayed by E-Studio reflects the version of the Net Station PackageFile that is currently installed on your machine and may not match the picture.

Task 4: Configure the Experiment Startup Info

Configure the experiment to prompt for the Startup Info parameters that are required by Net Station.

Net Station requires that experiments provide information related to Subject, Session, Age, Sex, Handedness, and ResearcherID. Failure to configure E-Prime to send these parameters causes a runtime error. Subject and Session are already enabled in E-Prime by default but all the other parameters must be added to the experiment using the Startup Info tab of the Experiment Object Properties dialog. It is typical to prompt the researcher for this information at the beginning of the experiment. If you do not prompt for the information at runtime, you must set the Default value of each parameter to the value you want to use.

- 1) **Click** the **Startup Info** tab on the **Experiment Object Properties** dialog.
- 2) **Click** the **checkboxes** of the **Age**, **Sex**, **Handedness**, and **ResearcherID** parameters so these are included in the experiment.



- 3) **Click** the **icon** beside each of the parameters that were recently included so that the experiment prompts to specify the values at runtime.



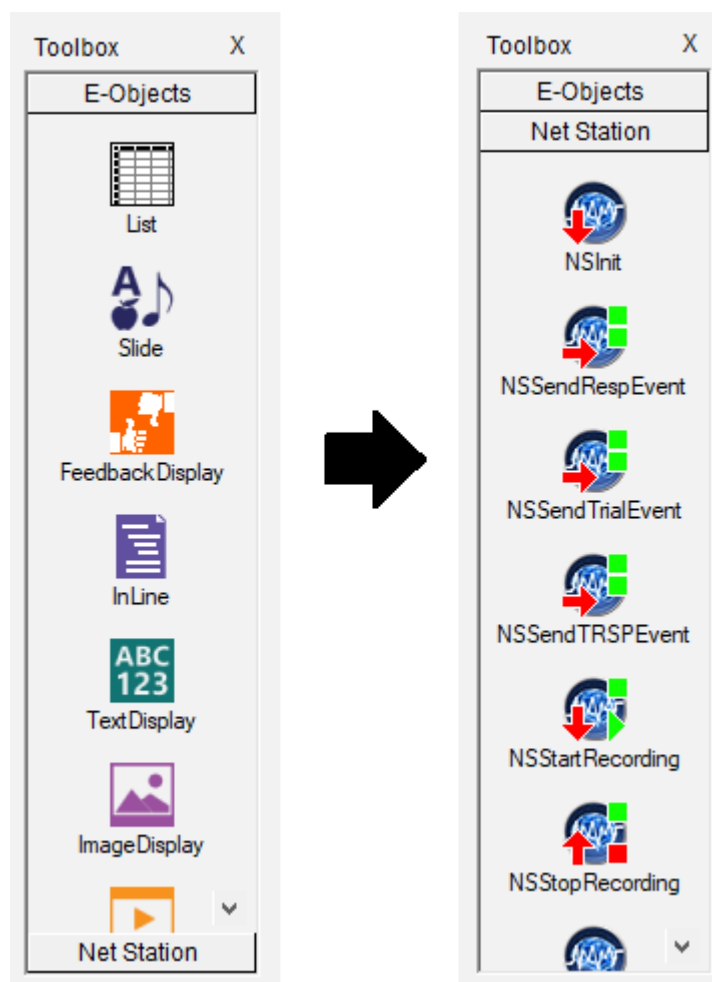
- 4) **Click** the **OK** button to accept and dismiss the dialog.

Task 5: Open the Net Station Toolbox to use the Net Station PackageCall Shortcuts

Click the Net Station heading in the Toolbox to show the Net Station PackageCall Shortcuts.

E-Prime Extensions for Net Station 3.0 utilizes pre-configured PackageCalls for communicating with Net Station. These pre-configured PackageCall Shortcuts can be dragged and dropped to the appropriate Procedure without any further editing. After the Net Station PackageFile is added to the experiment, these PackageCalls can be accessed from the Toolbox. A new heading (i.e., Net Station) exists in the Toolbox once the PackageFile is added.

1) **Click** the **Net Station heading** in the **Toolbox** to display the Net Station PackageCalls.



NOTE: To view the E-Objects again, click the E-Objects heading.

Task 6: Add the NSInit PackageCall to Initialize the System

Add a PackageCall at the beginning of the SessionProc to initialize the system for use with Net Station.

The **NSInit** PackageCall initializes Net Station at the start of the experiment. It is preconfigured to call the NetStation_Init routine. The NSInit PackageCall is added at the beginning of the SessionProc for this purpose.

The first parameter (**c**) is the common to most PackageCall routines and simply passes in the current Experiment Context.

The second parameter (**strState**) is a required parameter. This parameter controls the state of the communications between E-Prime and Net Station. This parameter can either be specified as “on” or “off”. When “on” is specified, the system attempts to establish communications with Net Station and a non-fatal error is produced if the attempt fails. When “off” is specified, the system does not attempt to establish communications with Net Station and no errors are generated. For information on how to run an NS-enabled experiment without the hardware connected, please refer to [Troubleshooting, 4.3: Running an Experiment Without Net Station Connected](#).

The third required parameter (**objCellList**) is the name of the Net Station CellList. The CellList is a List Object in the experiment that contains attributes (minimally CellNumber and CellLabel) which describe and identify the cells that may be used within the experiment. For more information on the CellList, please refer to [Tutorial 1, Task 14: Add a CellList Object](#).

The fourth parameter (**varCommMethod**) is an optional parameter that defines which communication method E-Prime uses to establish communication with Net Station. Currently, “socket” is the only recommend method, although “serial” is a deprecated method used with earlier EENS versions. The communication method can be specified in several places. If not specified as a parameter in the Init routine, it can be specified in an E-Prime global or local StartupInfo file. We recommend using a global StartupInfo file rather than a local file to ensure consistent results with the EENS-enabled experiments run in the same lab. For more information on using StartupInfo files, please refer to the following:

[Article 22727: StartupInfo Editor](#)

Lastly, the communications method can be specified in the NETSTATION.INI file. This file is not installed as part of EENS 3.0 but is a deprecated method that is supported for both EENS 3.0 and EENS 2.0. Please refer to the following for more information on the NETSTATION.INI file:

[Article 27214: The NETSTATION.INI file](#)

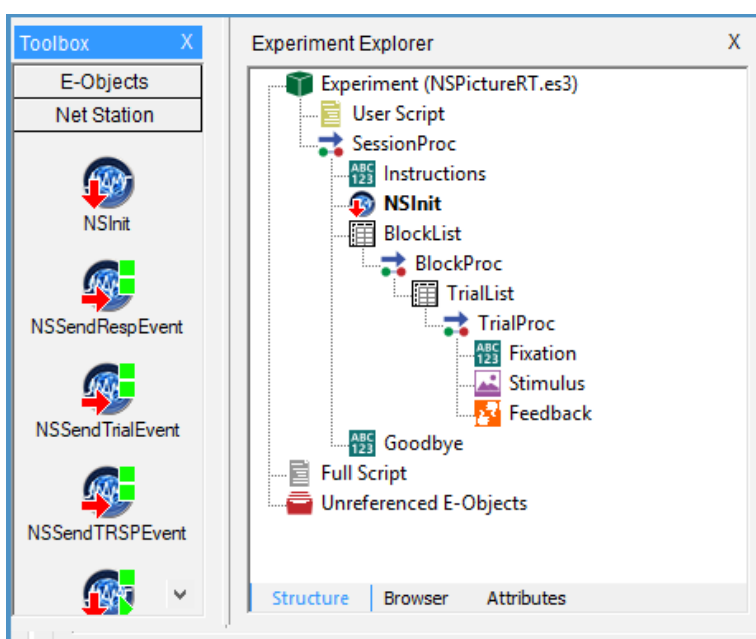
The fifth parameter (**varCommMethodParams**) is an optional parameter associated with the Net Station communications method. Like the communication method parameter described above, if not specified with the Init routine then it can be specified from a StartupInfo file (or the deprecated NETSTATION.INI file).

When varCommMethod is “socket”, this parameter is used to specify the IP address and port of the remote Net Station computer, e.g. “127.0.0.1:55513”. If no port number is specified the default port expected by Net Station is used.

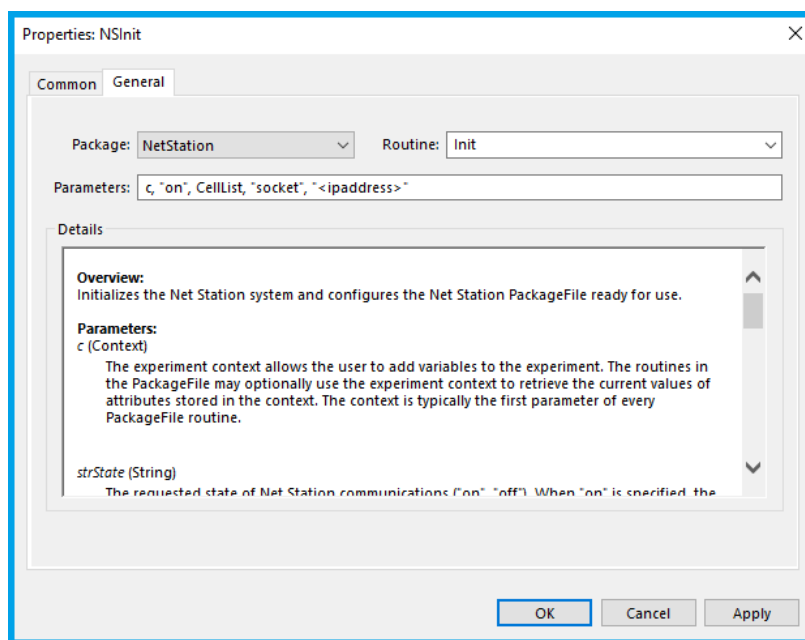
The sixth parameter (**varStopRecording**) is optional and simply indicates if Net Station should stop recording immediately after being initialized. If not specified, this defaults to True.

The final parameter (**vHandlePreRelease**) is an optional parameter that specifies if the HandlePreRelease routine is called prior to the execution of this Routine's script. If not specified, this defaults to True.

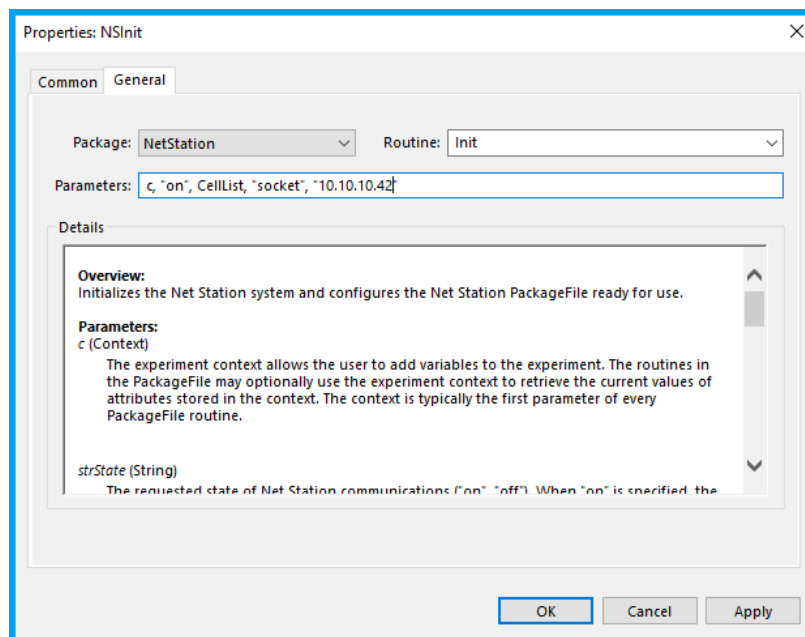
- 1) **Drag** the shortcut **NSInit** from the **Net Station Toolbox** and **drop** it in the **SessionProc** procedure after the TextDisplay **Instructions**.



2) **Double click** on the **NSInit** Object to open its **Properties** dialog.



3) **Edit** the **Parameters** field and **specify** the IP address for the Net Station computer.



NOTE: The recommended parameters are *c, "on", CellList, "socket", "10.10.10.42"*.

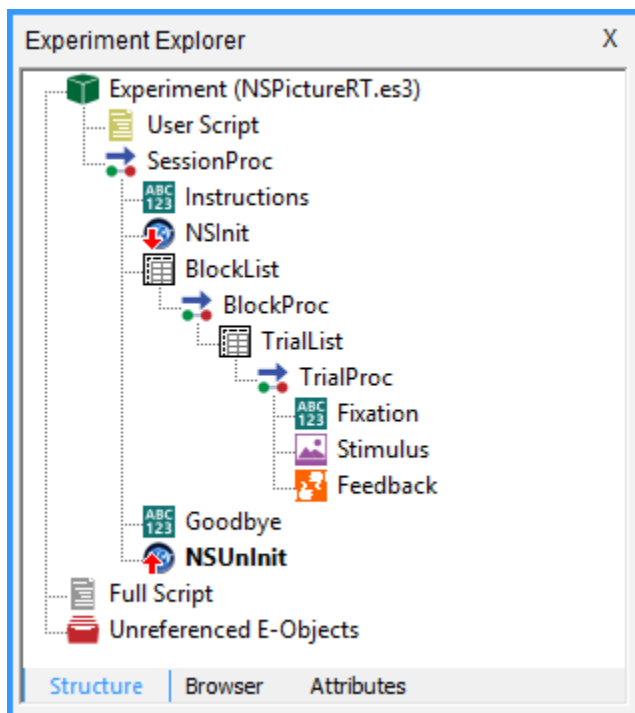
4) **Click** the **OK** button to accept and dismiss the dialog.

Task 7: Add a Net Station PackageCall to Uninitialize the System

Add a PackageCall at the end of the SessionProc to uninitialize the system.

At the end of a Net Station-enabled experiment, the system needs uninitialized. **NSUnInit** sends a message to Net Station indicating that the experiment is completed. This is typically placed at the end of your SessionProc.

- 1) **Drag** the shortcut **NSUnInit** Object from the **Net Station Toolbox** and **drop** it after the **Goodbye** Object in the **SessionProc**.

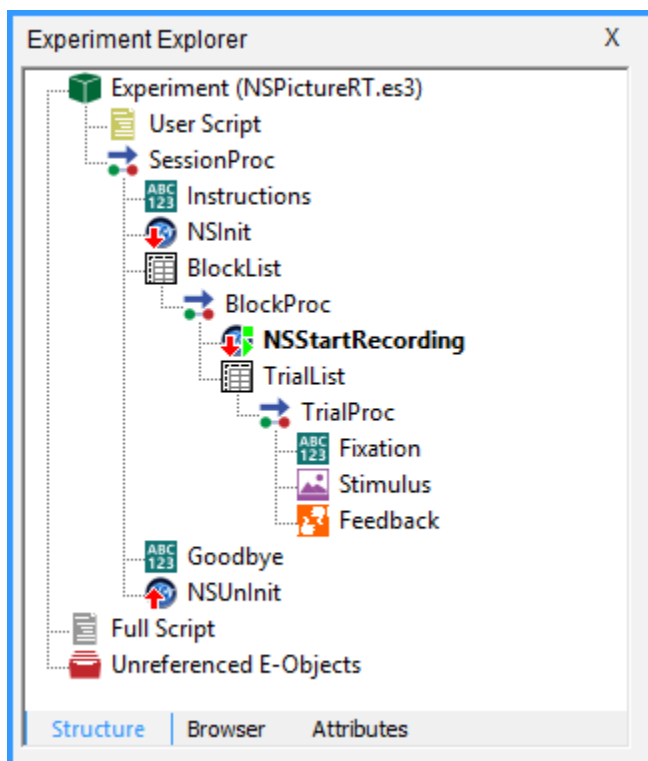


Task 8: Add a Net Station PackageCall to Start Recording

Add a PackageCall to the TrialProc to start the Net Station recording.

The next step is to add the PackageCall to start recording. **NSStartRecording** is placed at the location in the procedure where you want to start recording. This is typically placed at the beginning of a block of trials.

- 1) **Drag** the shortcut **NSStartRecording** Object from the **Net Station Toolbox** and **drop** it as the first object in the **BlockProc** procedure.

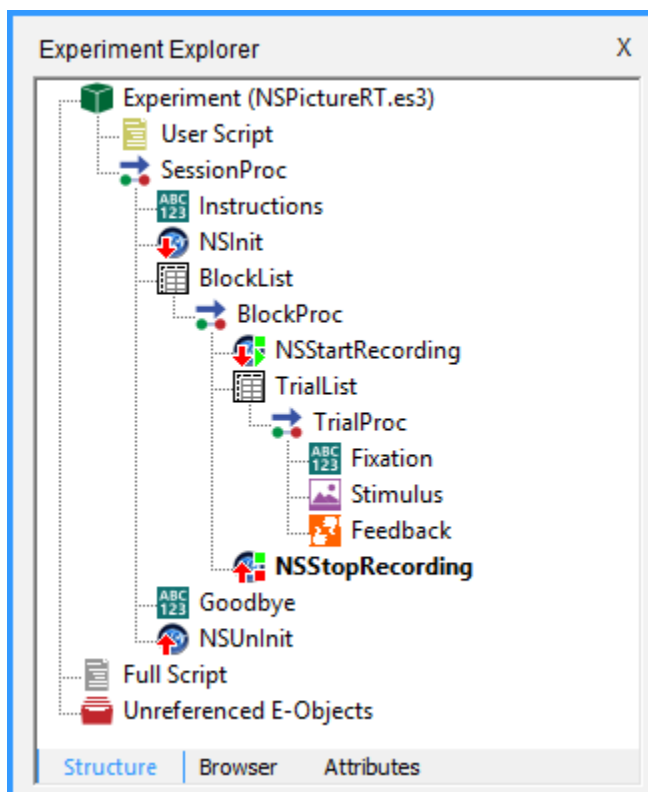


Task 9: Add a Net Station PackageCall to Stop Recording

Add a PackageCall at the end of the BlockProc to direct NetStation to stop recording.

The **NSStopRecording** PackageCall needs to be added to the experiment to stop the Net Station recording. This PackageCall is placed at the location in the procedure where you decide you want to stop recording. This is typically at the end of a block of trials.

- 1) **Drag** the shortcut **NSStopRecording** Object from the **Net Station Toolbox** and **drop** it as the last object in the **BlockProc** procedure.

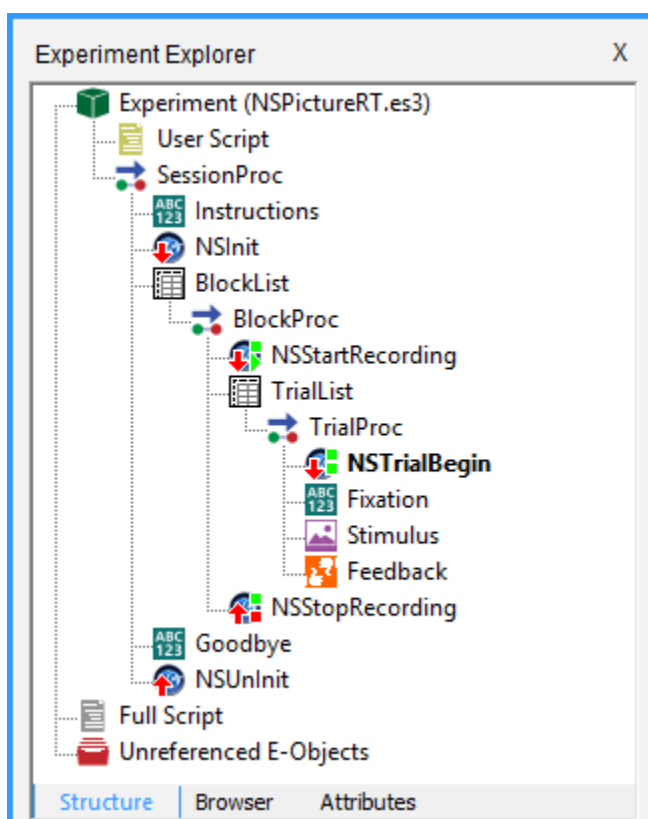


Task 10: Add a Net Station PackageCall to Initialize the Trial

Add a PackageCall at the beginning of the TrialProc to initialize each trial.

Net Station should be notified at the start of each trial with the PackageCall **NSTrialBegin**. This PackageCall is typically placed at the beginning of the TrialProc and sends a "begin" message to Net Station. When this Routine is called, it determines the currently active cell by searching the Experiment Context for an attribute named "CellNumber". Once the current cell is determined, the Routine uses this information to increment the number of observation for the current cell as well as the current Net Station trial number.

- 1) **Drag** the shortcut **NSTrialBegin** Object from the **Net Station Toolbox** and **drop** it as the first object in the **TrialProc** procedure.

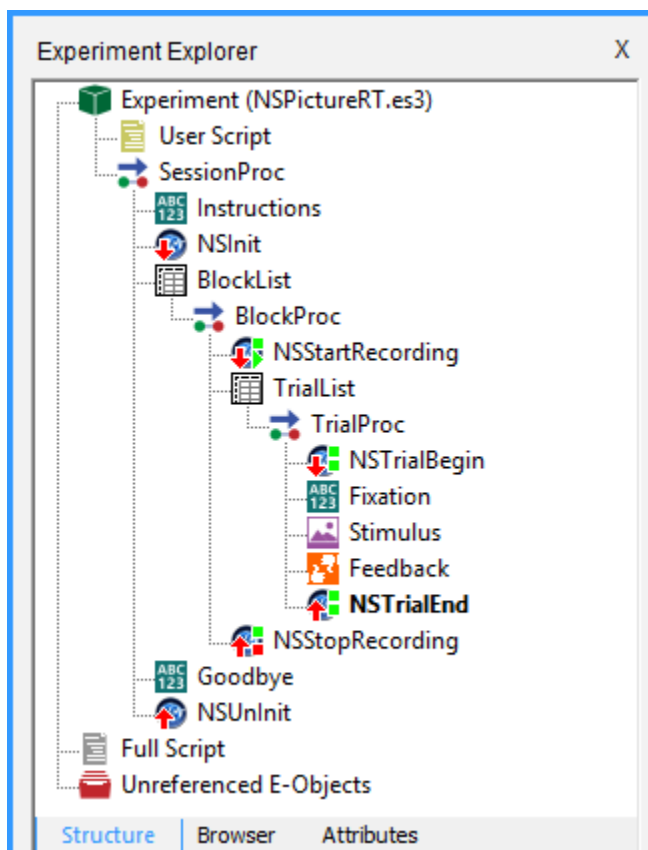


Task 11: Add a Net Station PackageCall to Uninitialize the Trial

Add a PackageCall at the end of the TrialProc to uninitialize each trial.

Although not required, Net Station is typically notified of the end of each trial with **NSTrialEnd**. This PackageCall is placed at the end of the TrialProc Procedure to clear the message buffer and prepare the communications engine for the next trial

- 1) **Drag** the shortcut **NSTrialEnd** Object from the **Net Station Toolbox** and **drop** it as the last object in the **TrialProc** procedure.



Task 12: Add script to send trial events to Net Station

Add an InLine Object named “NSSendTrialEvents” after the Feedback Object. Add script to the InLine to make direct PackageCalls to send the required trial information to Net Station.

Net Station requires that it is notified of all critical trial events (e.g., stimulus events presented or participant responses that were accepted during the trial). While the typical recommendation is to use shortcuts or PackageCalls to call Routines from the Net Station PackageFile, the InLine method is recommended for sending the trial events. Please refer to the section [InLine Script Method](#) for more information on calling PackageCalls from an InLine.

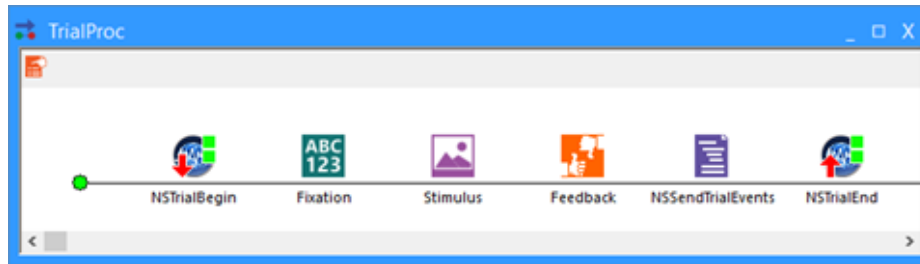
The first trial event that is sent is **NetStation_SendRespEvent**. This Routine notifies Net Station of any participant response that is collected during each trial. This Routine has two required parameters: “c” which is the Experiment Context and “objRteObject.” objRteObject requires that you pass in a reference to the object within the trial sequence that collected the response (e.g., the Stimulus Object in this example).

The second trial event that is sent is **NetStation_SendTrialEvent**. Prior to the end of each trial, all timing information related to any critical stimulus presentation that occurred within the trial is sent to Net Station with this Routine. The syntax of NetStation_SendTrialEvent requires the parameters of the current Experiment Context (c) and a reference to the object that generated the current event (objTrialEvent). For example, in this experiment, three separate NetStation_SendTrialEvents are called with the objTrialEvent specified as the Fixation, Stimulus, and Feedback Objects because they directly alter the visual display.

The last trial event is **NetStation_SendTRSPEvent**. Net Station requires that a TRial SPecific (TRSP) event is sent prior to the end of each trial so that it can be notified of behavioral data related to the critical stimulus and response. The syntax of this Routine call requires the parameters of the current Experiment Context (c) and a reference to the object that collected the participant’s response (objStimulus). objStimulus requires a reference to the object in the trial sequence that is responsible for presenting the critical stimulus and optionally collecting the response data (e.g., the Stimulus object in this example).

- 1) **Click** the **E-Objects heading** in the **Toolbox** to display the E-Prime E-Objects.
- 2) **Drag** an **InLine** Object from the **Toolbox** and **drop** it after the **Feedback** Object in the **TrialProc** Procedure.

3) **Click** on the new **InLine** Object to select it and **press F2** and **rename** it “**NSSendTrialEvents.**”



4) **Double click** on **NSSendTrialEvents** Object to open it in the workspace.

5) **Type** in the following script to call **NetStation_SendRespEvent**:

```
'      Send response using data collected from specified object
NetStation_SendRespEvent c, Stimulus
```

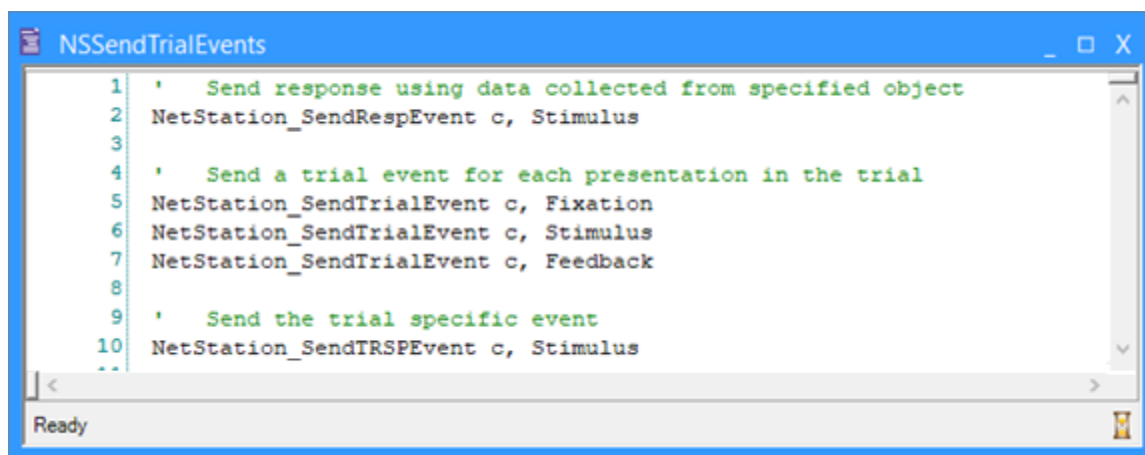
6) **Type** in the following script after the **NetStation_SendRespEvent** routine you previously typed to call **NSSendTrialEvent**:

```
'      Send a trial event for each presentation in the trial
NetStation_SendTrialEvent c, Fixation
NetStation_SendTrialEvent c, Stimulus
NetStation_SendTrialEvent c, Feedback
```

7) **Type** in the following script after the **NetStation_SendTrialEvent** routines you previously typed to call **NSSendTRSPEvent**:

```
'      Send the trial specific event
NetStation_SendTRSPEvent c, Stimulus
```

NOTE: Your final **NSSendTrialEvents InLine** Object should look the same as the image below:



Task 13: Assign a unique four-character code to each object of interest

Assign a four-character code to each stimulus presentation object or event of interest within the Trial sequence by setting the tag property of the Fixation Object.

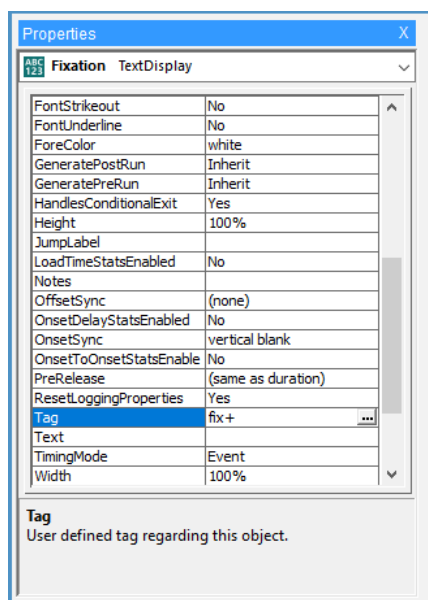
Net Station requires that each event of interest within a trial sequence be identified via a unique four-character code. Within a Net Station-enabled experiment, you assign these unique codes to objects by setting each object's "Tag" property. At runtime, Routines within the Net Station PackageFile examine the assigned Tag property of an object to associate data sent to Net Station with a specific object. Additionally, the tags you set are shown in the Net Station interface as event markers. In this Tutorial, you need to set the Tag property of Fixation, Stimulus, and Feedback to "fix+", "stm+", and "stm-" respectively.

NOTE: The Tag property is viewable via an E-Object's Property Window. For the subsequent steps, verify that the Property Window is viewable before proceeding. If you need to display the Properties Window, navigate to "View" in E-Studio and click on "Properties."

NOTE: Assigned codes must be exactly four-characters in length or an error is generated when the experiment is run with Net Station.

1) **Click** on the **Fixation** Object to show **Fixation's Property window**.

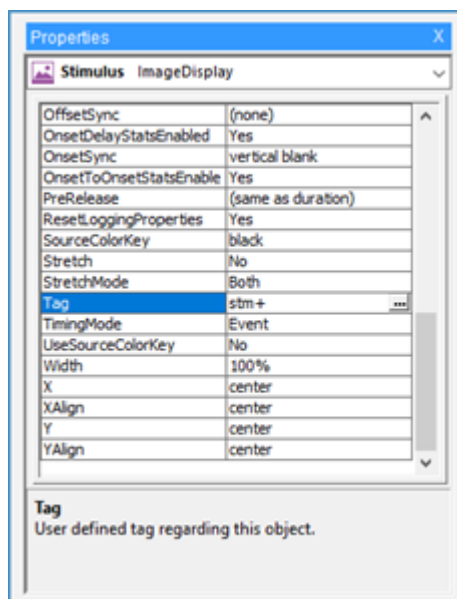
2) **Click** in the **edit box** beside the Tag property and **type "fix+"**.



3) **Press Enter** to accept the change.

4) **Click** on the **Stimulus** Object to show **Stimulus's Property window**.

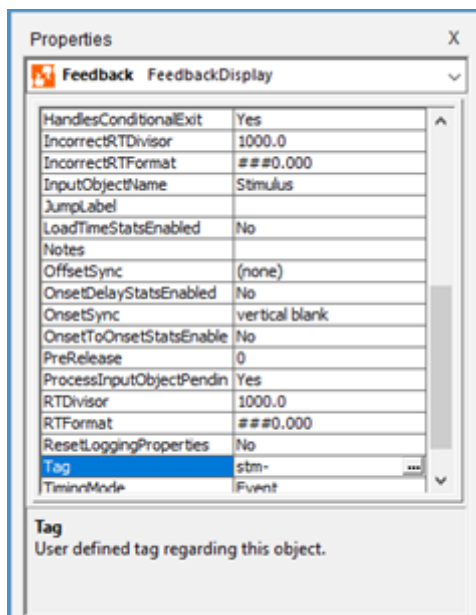
5) **Click** in the **edit box** beside the Tag property and **type** “stm+”.



6) **Press Enter** to accept the change.

7) **Click** on the **Feedback** Object to show **Feedback's Property** window.

8) **Click** in the **edit box** beside the Tag property and **type** “stm-”.



9) **Press Enter** to accept the change.

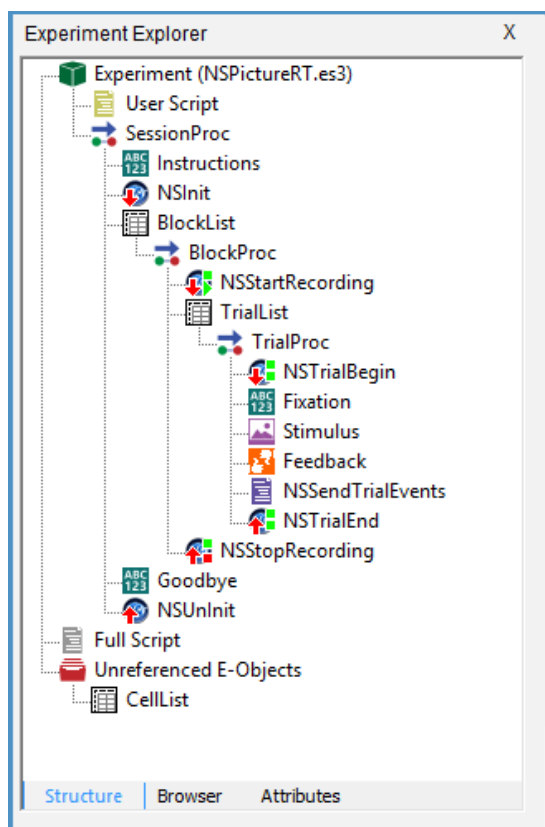
Task 14: Add a CellList Object

Add a new List Object to the Unreferenced E-Objects branch of the tree.

Net Station identifies conditions or “cells” within an experiment by referencing user-defined Cell Numbers and Cell Labels. The Routines within the Net Station PackageFile look up cell information within the E-Prime experiment by examining the attributes of the specific List Object that is identified in the NSInit routine. Specifically, the *objCellList* parameter in the NSInit routine denotes the name of the ListObject that defines the Net Station cells. Back in Step 6 of this Tutorial, the *objCellList* parameter was set to the value CellList (see [Task 6: Add the NSInit PackageCall to Initialize the System](#)). In this step we create a List Object and name it CellList.

NOTE: The *objCellList* Object must be placed in the Unreferenced E-Objects.

- 1) **Drag** a new List Object from the **E-Object's Toolbox** and **drop** it in the **Unreferenced E-Objects**.
- 2) **Click** on the new List (i.e., List1) and **press F2** to **rename** the object “**CellList**”.



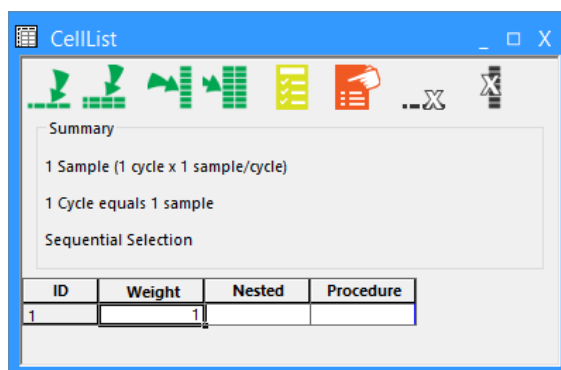
Task 15: Add the required CellNumber and CellLabel attributes to the CellList

Open the CellList Object in the Workspace and add two new attributes and one new level.

Once the List Object is created in Unreferenced Objects, a couple of Attributes and another Level needs to be added to the List Object. The CellList must contain two Attributes named “CellNumber” and “CellLabel”. Additionally, another Level needs to be added to the list.

NOTE: If the Attributes specified above are not added in CellList then the experiment generates an error at runtime.

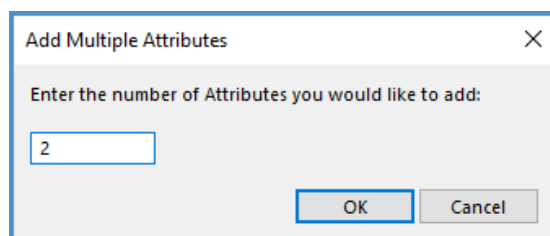
1) **Double click** the **CellList** Object to open it.



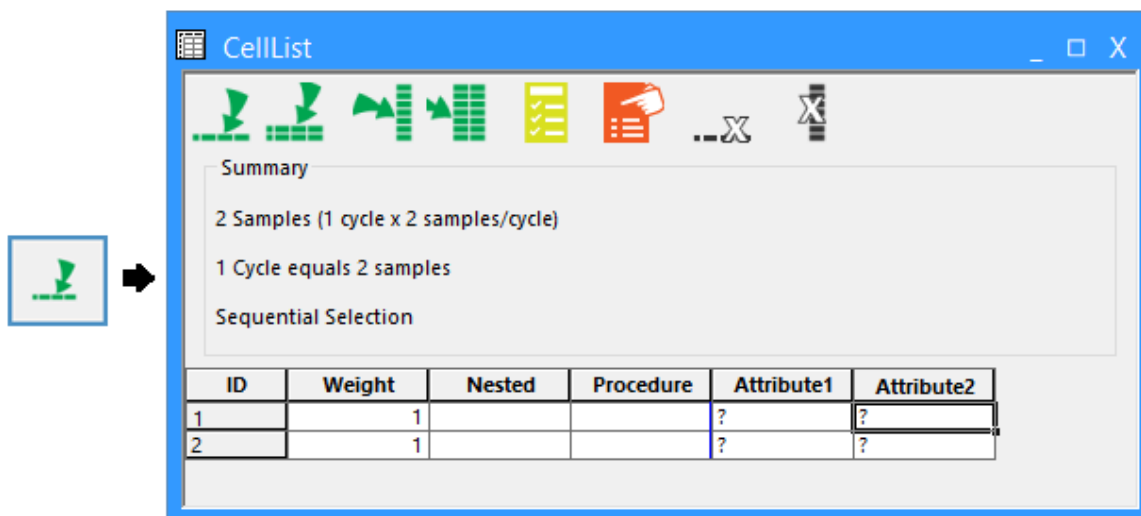
2) **Click** the **Add Multiple Attributes** button.



3) **Type “2”** in the **Add Multiple Attributes** dialog and **click OK** to add the Attributes.

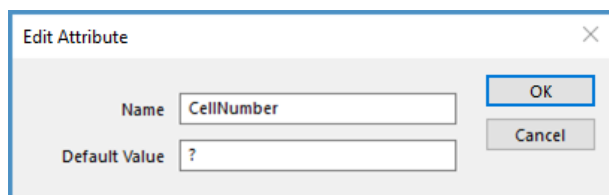


4) **Click** the **Add Level** button to add an additional Level.



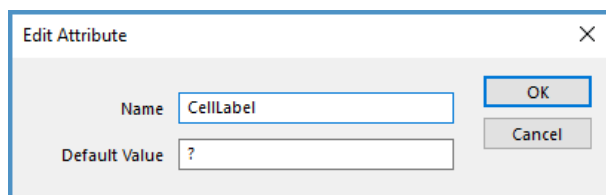
5) **Double click** the **header** of the **Attribute1** column to display the **Edit Attribute** Dialog.

6) **Type** “CellNumber” in the **Name** field and **click OK** to accept the change.

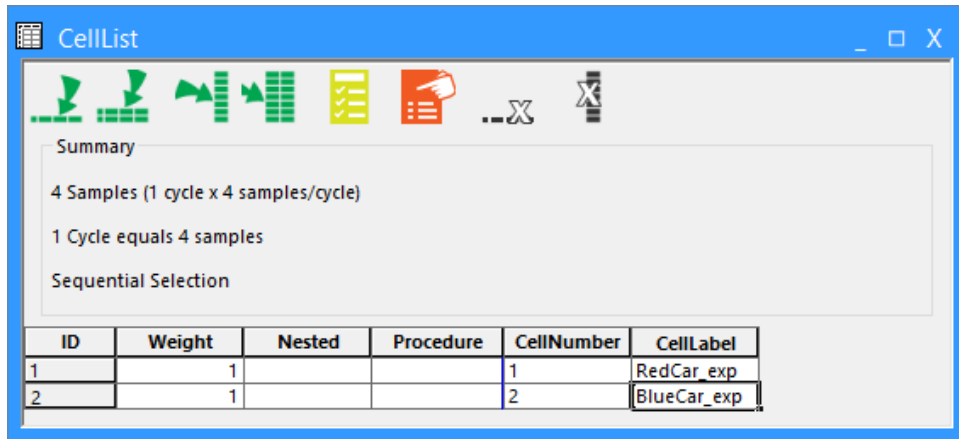


7) **Double click** the **header** of the **Attribute2** column to display the **Edit Attribute** Dialog.

8) **Type** “CellLabel” in the **Name** field and **click OK** to accept the change.



- 9) **Click** in each cell under **CellNumber** and edit the values to be “1” for Level 1 and “2” for Level 2.
Click in each cell under **CellLabel** and edit the values to be “RedCar_exp” for Level 1 and “BlueCar_exp” for Level 2.



The screenshot shows a window titled "CellList" with a toolbar containing icons for file operations, data manipulation, and printing. Below the toolbar is a "Summary" section with the following text:

4 Samples (1 cycle x 4 samples/cycle)
1 Cycle equals 4 samples
Sequential Selection

Below the summary is a table with the following data:

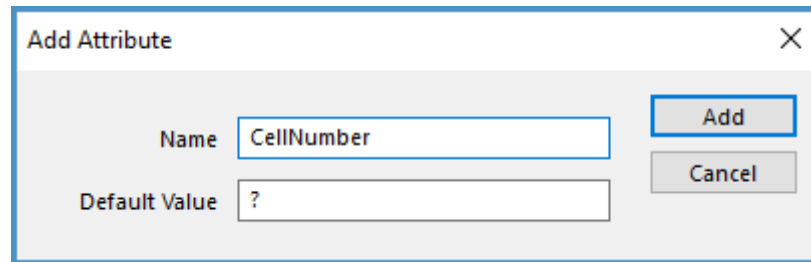
ID	Weight	Nested	Procedure	CellNumber	CellLabel
1	1			1	RedCar_exp
2	1			2	BlueCar_exp

Task 16: Add the required CellNumber attribute to the TrialList

Add a CellNumber attribute to the TrialList. Assign values to the CellNumber attribute to identify the specific cells represented in the TrialList.

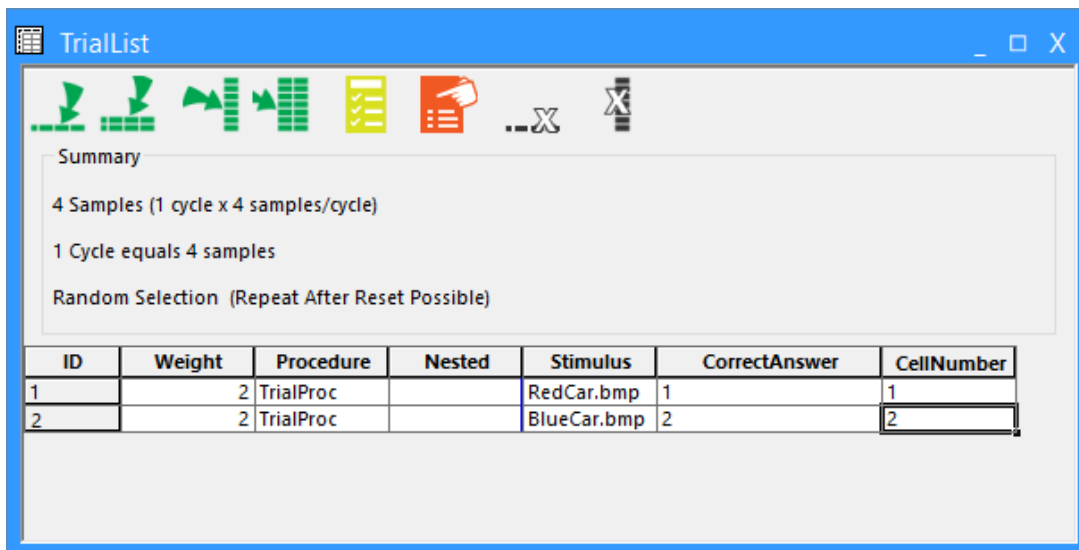
While you may define many different types of cells within an experiment, each individual trial must be associated with a single specific cell at runtime. You associate a trial with a cell by adding a CellNumber Attribute to your TrialList and then setting its values to identify the CellNumber of appropriate cell that is related to each trial represented in the list.

- 1) **Double click** the **TrialList** Object in the Workspace.
- 2) **Click** the **Add Attribute** button.
- 3) **Type “CellNumber”** in the **Name** field and **click Add** to create the Attribute.



The 'Add Attribute' dialog box is shown. It has a title bar with a close button (X). Inside, there are two input fields: 'Name' with the text 'CellNumber' and 'Default Value' with the text '?'. To the right of these fields are two buttons: 'Add' and 'Cancel'.

- 4) **Click** in each cell under **CellNumber** and edit the values to be “1” for Level 1 and “2” for Level 2.



The TrialList window is shown. It has a title bar with a close button (X). Below the title bar is a toolbar with several icons. The main area is divided into a 'Summary' section and a table. The 'Summary' section contains the following text: '4 Samples (1 cycle x 4 samples/cycle)', '1 Cycle equals 4 samples', and 'Random Selection (Repeat After Reset Possible)'. The table has the following columns: ID, Weight, Procedure, Nested, Stimulus, CorrectAnswer, and CellNumber. The table contains two rows of data:

ID	Weight	Procedure	Nested	Stimulus	CorrectAnswer	CellNumber
1	2	TrialProc		RedCar.bmp	1	1
2	2	TrialProc		BlueCar.bmp	2	2

Task 17: Run the Experiment

Run the experiment to verify that the appropriate prompts are presented and no runtime errors are generated.

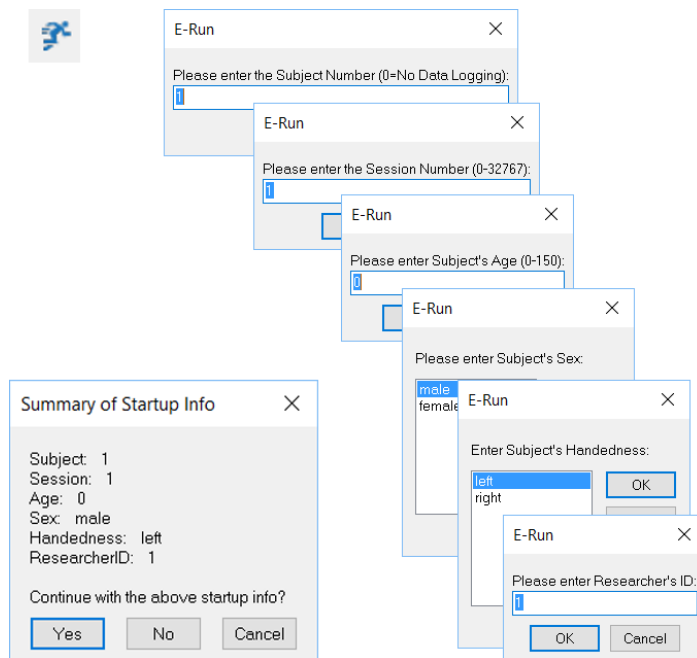
You have now completed the basic steps necessary to create Net Station-enabled paradigm. Net Station-enabled experiments can be run locally from E-Studio during development and testing. You should always test your experiment prior to scheduling actual participants or before using it to collect data.

Before you test the experimental task, it is important that you understand how the task works. First you see a brief fixation appear on the screen. After the fixation leaves the screen, either a red car or a blue car appears on the screen. Participants are told to press '1' if they see a red car and '2' if they see a blue car.

NOTE: When using E-Prime and the Net Station system you must have a two-computer setup to collect EEG data. E-Prime and E-Prime Extensions for Net Station must be installed on one computer and the Net Station system must be installed on another system. The two computers must be connected with a network hub or switch to communicate with each other. For information on connecting the machines, please refer to [Chapter 1, Section 1.3: Network Configuration](#).

NOTE: When testing without the Net Station hardware connected you receive a message during initialization that indicates that Net Station communications are disabled.

- 1) **Click** the **Run icon** on the toolbar to generate and run the experiment locally.
- 2) **Press Enter** to **accept** the **default values** for each of the initialization prompts presented.



- 3) **Observe** the **stimulus presentation sequence** to verify the experiment is functioning correctly and completes with no errors being generated.



3.4. Tutorial 2: Adding User Defined Keys

Summary:

Net Station allows you to send additional trial information to Net Station when sending the Trial Specifications or TRSP event. Any attribute that is defined during the trial can be sent. To enable this feature, you create and define a new “KeyList” within your experiment and add a reference to the KeyList in each call to the Net Station Package routine that is responsible for sending the TRSP event to Net Station. At runtime, the current value of all user-defined key’s attributes, defined in the KeyList, is retrieved from the Experiment Context and sent to Net Station as part of the TRSP event.

NOTE: *All the Tutorials, except for Tutorial 4: Timing Test, build upon each other and must be completed in order. Tutorial 2 begins where Tutorial 1 ends. This Tutorial assumes that you have completed [Tutorial 1: Adding Net Station Support to an E-Prime Experiment](#). You must load the experiment you edited in the last Tutorial into E-Studio before continuing with Tutorial 2.*

NOTE: *The Sample experiment found in the folder <My Experiments>\3.0\Net Station\Samples\NSPictureRT represents the complete set of changes documented in Tutorials 1 through 3.*

Goal:

This Tutorial illustrates how to add a user defined KeyList to the NSPictureRT experiment created in Tutorial 1. The KeyList includes one new key (named “stim”) that sends Net Station a reference to the name of the actual Stimulus that was presented on each trial.

Overview of Tasks:

- Add a KeyList to the Unreferenced E-Objects branch of the Structure window.
- Configure the KeyList Object to add the required attributes and user-defined keys.
- Add a reference to the KeyList Object in the NetStation_SendTRSPEvent call.
- Verify the overall experiment structure and run the experiment.

Recommended readings:

It is required that you have read and worked through and completed Tutorial 1 before beginning this Tutorial.

Because this Tutorial requires adding and editing an additional List Object, we recommend reviewing the following Article on List Objects:

[Article 22700: List Object](#)

Estimated Tutorial Time:

10-15 minutes

NOTE: *If you do not want to overwrite your previous work, save the experiment under a different name before you begin this Tutorial.*

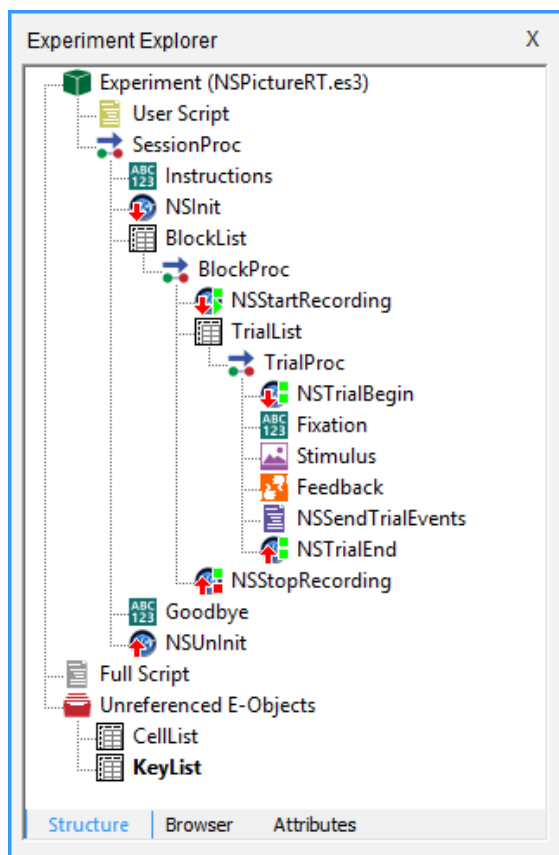
Task 1: Add a KeyList Object

Add a new List Object to the Unreferenced E-Objects branch of the Structure window.

A Net Station enabled experiment identifies user-defined keys using one or more “Key Lists” defined within the experiment. The Routines within the Net Station Package File use these lists at runtime to lookup information (e.g. name, ID, data type, value) related to each user defined key. A **KeyList** is a List Object that has been created in the experiment and configured with a specified set of named attributes. When designing your experiments, the Unreferenced E-Objects section of the Structure window is the typical location the KeyList is placed.

NOTE: If it is not already opened, open the NSPictureRT.es3 experiment you created in Tutorial 1.

- 1) **Drag** a new **List** Object from the **Toolbox** and **drop** it on the **Unreferenced E-Objects** branch of the **Structure** window.
- 2) **Click** on the new **List1** Object to **select** it then **press F2** to **rename** the object as “**KeyList**”.

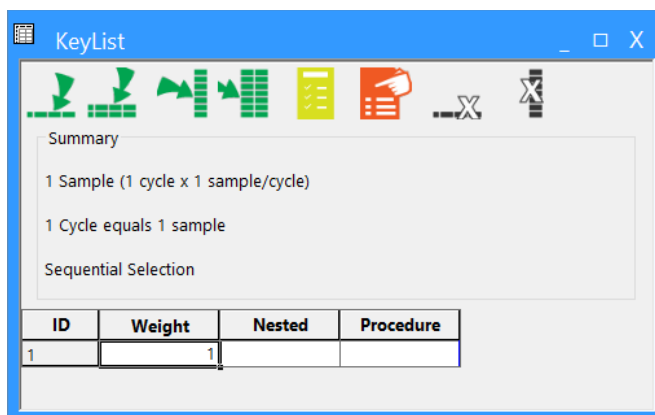


Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList

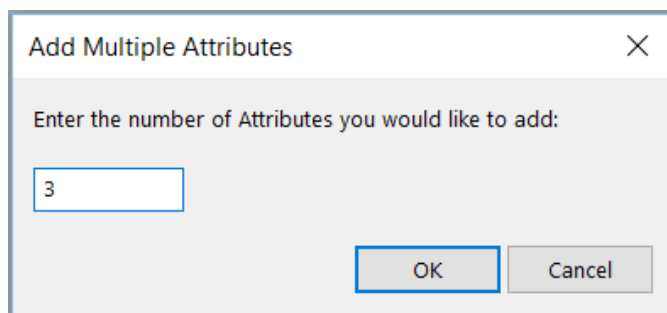
Open the KeyList Object in the workspace and add three new attributes.

Each Key List that is defined in an experiment must contain the three named attributes of “KeyName”, “KeyID”, and “KeyDataType”. The Routines within the Net Station PackageFile expect each List Object that is used as a Key List to contain these specifically named Attributes. If you misspell or forget to include one of these attributes the Net Station PackageFile generates an error at runtime. The value of **KeyName** must be a string. The value of **KeyID** must be a four-character code. The value of **KeyDataType** must be one of the following codes: “shor”, “long”, “bool”, or “TEXT”. If the value of KeyDataType is left blank, it will default to “shor”.

- 1) **Double click** the **KeyList** Object to open it in the workspace. **Click** the **Add Multiple Attributes** button.



- 2) **Type “3”** in the **Add Multiple Attributes** dialog.



- 3) **Click** the **OK** button to **add three** new attributes to the KeyList.

ID	Weight	Nested	Procedure	Attribute1	Attribute2	Attribute3
1	1			?	?	?

- 4) **Double click** the header of the **Attribute1** column to display the **Edit Attribute** dialog. **Type** “KeyName” in the **Name** field of the **Edit Attribute** dialog. **Click** the **OK** button to **accept** the change.

Edit Attribute

Name: KeyName

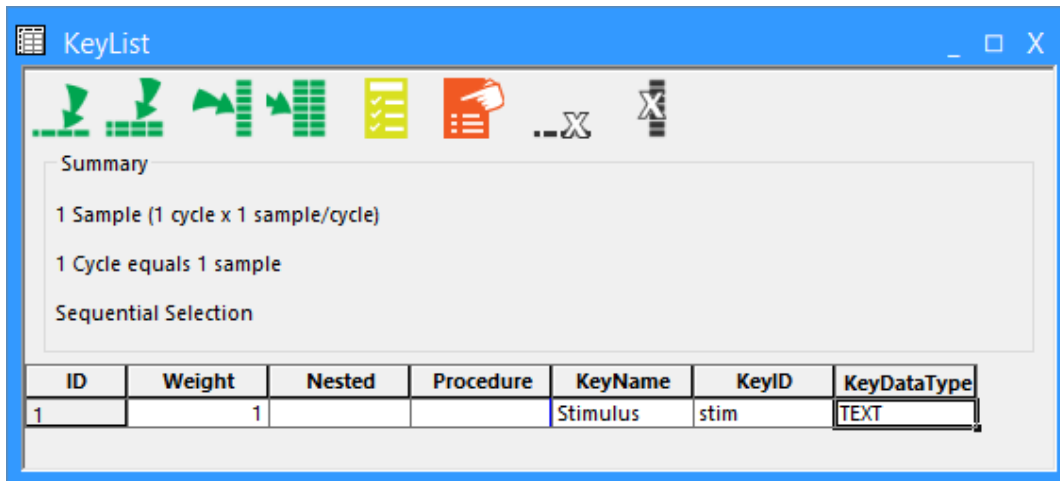
Default Value: ?

OK Cancel

- 5) **Repeat** Step 4 above to **rename Attribute2** and **Attribute3** to **KeyID** and **KeyDataType** respectively.

ID	Weight	Nested	Procedure	KeyName	KeyID	KeyDataType
1	1			?	?	?

- 6) **Click** in the cell under **KeyName** and edit the value to “**Stimulus**”. **Click** in the cell under **KeyID** and edit the value to “**stim**”. **Click** in the cell under **KeyDataType** and edit the value to “**TEXT**”.



Task 3: Add a reference to the KeyList

Add script to the *NSSendTrialEvents Inline* to append the *KeyList* to the *PackageFile* routine that sends the *TRial SPecific (TRSP)* event to Net Station.

The *NetStation_SendTRSPEvent* includes an optional parameter that is used to specify a reference to a list of user-defined keys. While it is acceptable to define multiple Key Lists within an experiment, typically only one list is used and one KeyList is sent for a given call to *NetStation_SendTRSPEvent*. Each Key List that is defined must be configured to fully describe the additional key data that is to be associated with the current trial.

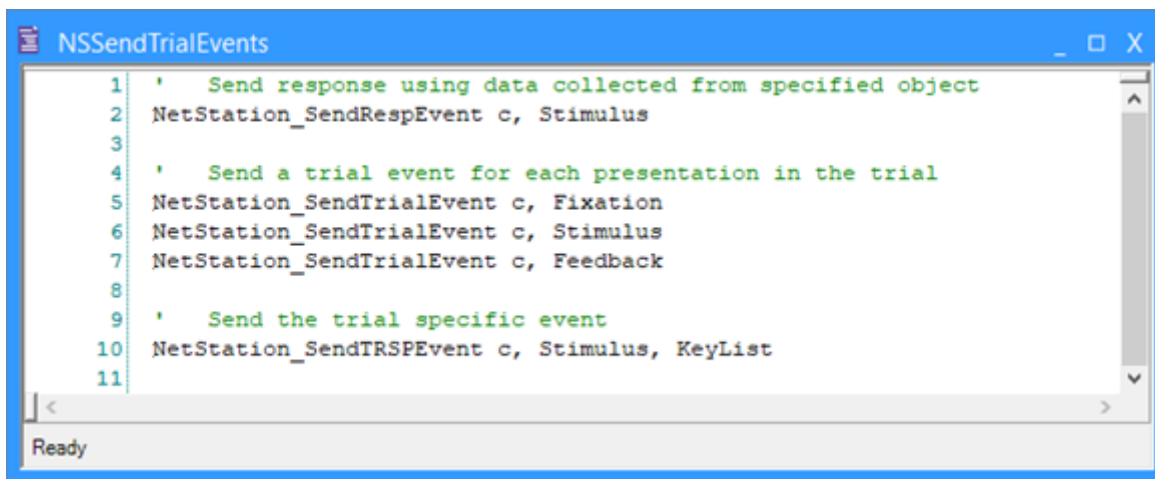
The optional parameter at the end of the *NetStation_SendTRSPEvent* parameter needs to be associated with the KeyList Object. At runtime, the *NetStation_SendTRSPEvent* routine processes the KeyList by retrieving each KeyName, looking up its current value within the Experiment Context, and then appending all the key information to the end of the TRSP event. If a specified KeyName cannot be found in the context, a runtime error is generated.

1) **Double click** on the **NSSendTrialEvents** Inline Object to open the dialog box.

2) **Edit** the **NetStation_SendTRSPEvent** script to include the **KeyList**:

```
'      Send the trial specific event
NetStation_SendTRSPEvent c, Stimulus, KeyList
```

NOTE: Your final *NSSendTrialEvents Inline* Object should look the same as the image below:



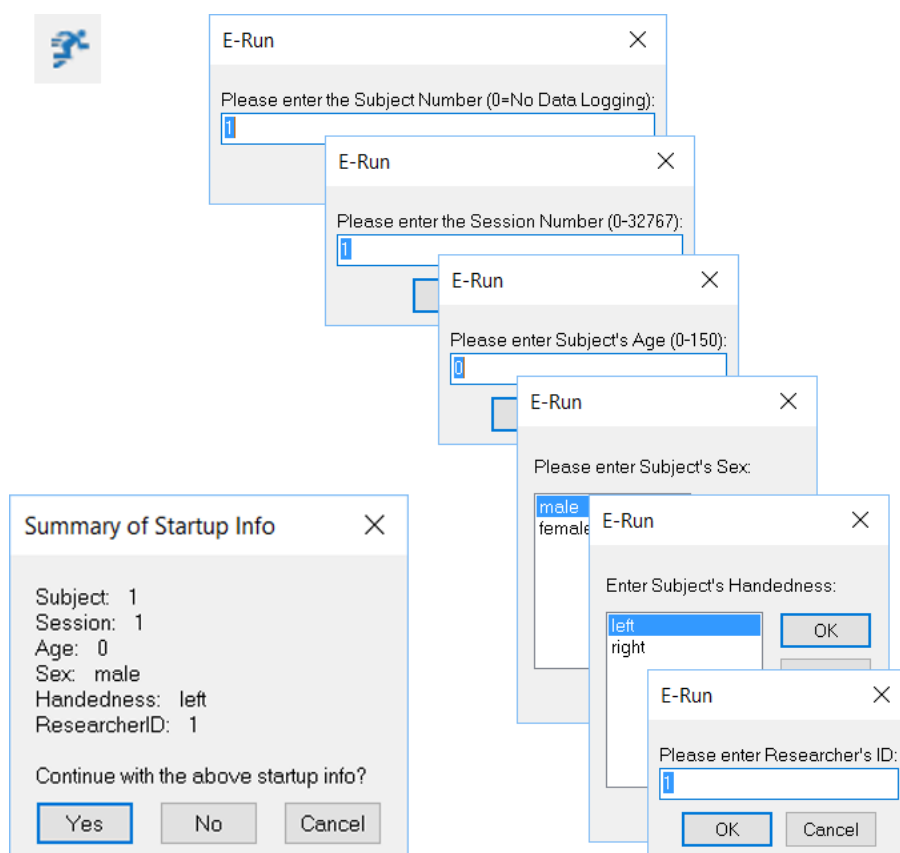
Task 4: Run the Experiment

Run the experiment to verify that the new key is being sent to Net Station and that no runtime errors are generated.

Your experiment should now be configured to send additional keys from the KeyList. Once you have launched the experiment you are presented with a series of initialization prompts. Press Enter or click OK to accept the default values during testing. Read and respond to any instructions presented then observe the stimulus presentation sequence to verify the experiment is functioning correctly and that no errors are generated. You can verify that the events are being sent by looking at the Event Track in Net Station and the ECI log window, which is launched from the ECI controls on the left panel of controls in Net Station Acquisition. Events should be visible in both locations.

NOTE: When using E-Prime and the Net Station system you must have a two-computer setup to collect EEG data. E-Prime and E-Prime Extensions for Net Station must be installed on one computer and the Net Station system must be installed on another system. The two computers must be connected with a network hub or switch to communicate with each other. For information on connecting the machines, please refer to [Chapter 1, Section 1.3: Network Configuration](#).

- 1) **Click** the **Run** icon on the toolbar to generate and run the experiment locally.
- 2) **Press Enter** to **accept** the **default values** for each of the initialization prompts presented.



- 3) **Observe** the **stimulus presentation sequence** to **verify** the **experiment** is **functioning** correctly and **completes** with **no errors** being generated.



3.5. Tutorial 3: Adding a Practice Block

Summary:

It is common for the design and implementation of an experiment to evolve over time (e.g., number of trials is increased/decreased as new stimuli are added/deleted, new experimental conditions/cells are introduced, participant friendly breaks are introduced, etc.). Many experiments introduce blocks of practice trials after a basic version of the experiment is functioning correctly. Within a Net Station-enabled experiment, introducing a block of practice trials often introduces a new set of cells that need to be reflected to Net Station (e.g. practice-related data can be ignored during critical data analysis). This process typically involves adding or restructuring entries in the CellList and assigning new CellNumbers to the appropriate trials.

NOTE: *All the Tutorials, except for Tutorial 4: Timing Test, build upon each other and must be completed in order. Tutorial 3 begins where Tutorial 2 ends. This Tutorial assumes that you have completed [Tutorial 1: Adding Net Station Support to an E-Prime Experiment](#) and [Tutorial 2: Adding User Defined Keys](#). You must load the experiment you edited in those Tutorials into E-Studio before continuing with Tutorial 3.*

NOTE: *The Sample experiment found in the folder <My Experiments>\3.0\Net Station\Samples\NSPictureRT represents the complete set of changes documented in Tutorials 1 through 3.*

Goal:

This Tutorial illustrates how to add a new block of practice trials into the NSPictureRT Sample experiment created during Tutorial 1 and Tutorial 2. You need to edit the experiment's CellList and assign new CellNumbers and CellLabels to the trials of the practice block.

Overview of Tasks:

- Add a new PracticeBlockProc procedure to the BlockList Object.
- Add the required Net Station PackageCalls to the PracticeBlockProc.
- Add and configure a new PracticeTrialList Object to represent the new practice trials.
- Add instruction displays to begin and end the block of practice trials.
- Verify the overall experiment structure and run the experiment.

Recommended readings:

It is required that you have read through and completed at least Tutorial 1 and 2 before beginning this Tutorial.

Estimated Tutorial Time:

15-30 minutes

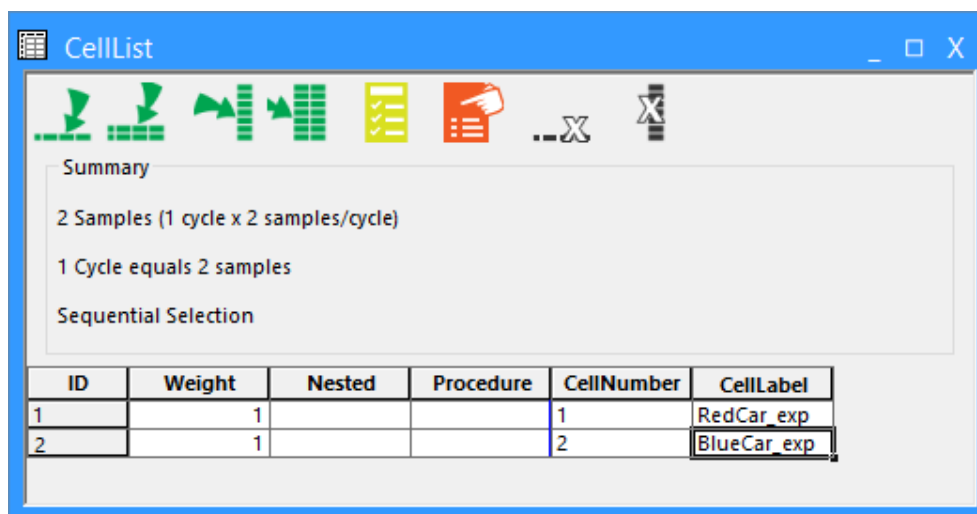
Task 1: Add new cells to the CellList

Add two new cells to the CellList, and edit the attribute values to define the new practice cells used within the experiment.

Adding a block of practice trials to the experiment creates two new Levels (i.e., rows) in the CellList. Once the Levels are added, the CellNumber and CellLabel need to be edited for the two new rows. The values for the **CellNumber** attribute must be numeric and must be unique within the experiment. You may currently define up to 255 unique cells within an experiment. It is recommended that when you add new cells to the CellList you assign new CellNumbers sequentially (e.g. to assure compatibility with any previously collected data). Values for the **CellLabel** can be any string you choose to identify and describe the cell (i.e., CellLabels do not have to be four-character codes).

NOTE: If it is not already opened, open the NSPictureRT.es3 experiment you created in Tutorial 1 and edited in Tutorial 2.

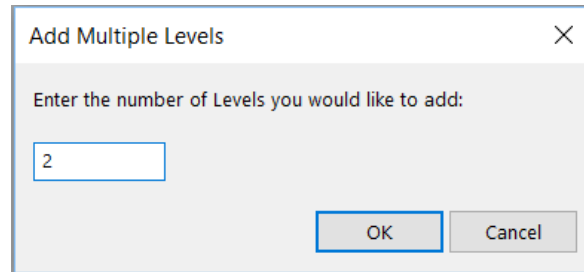
- 1) **Open NSPictureRT.es3** and **open** the **CellList** Object in the workspace.



- 2) **Click** the **Add Multiple Levels** button. **Type “2”** in the **Add Multiple Levels** dialog.

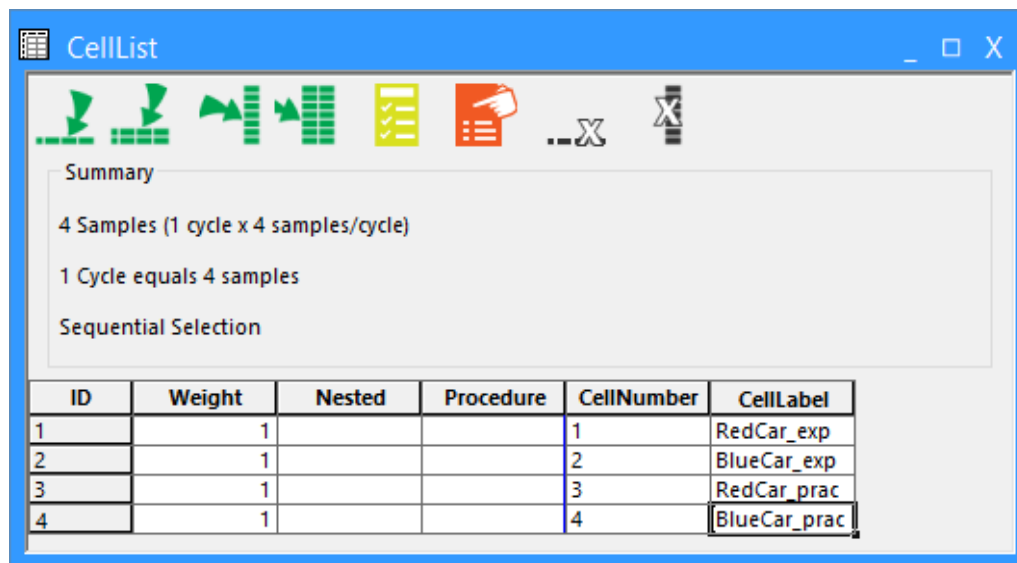


3) **Click** the **OK** button to **add two new levels/rows** to the **CellList**.



A dialog box titled "Add Multiple Levels" with a close button (X) in the top right corner. The text inside says "Enter the number of Levels you would like to add:". Below this text is a text input field containing the number "2". At the bottom right are two buttons: "OK" and "Cancel".

4) **Click** in each cell under **CellNumber** and edit the values to be **“3”** for Level 3 and **“4”** for Level 4. **Click** in each cell under **CellLabel** and edit the values to be **“RedCar_prac”** for Level 3 and **“BlueCar_prac”** for Level 4.



The "CellList" window shows a summary of the current setup and a table of cells. The summary indicates 4 samples (1 cycle x 4 samples/cycle), 1 cycle equals 4 samples, and sequential selection. The table below lists the cells with their IDs, weights, nested status, procedures, cell numbers, and cell labels.

ID	Weight	Nested	Procedure	CellNumber	CellLabel
1	1			1	RedCar_exp
2	1			2	BlueCar_exp
3	1			3	RedCar_prac
4	1			4	BlueCar_prac

Task 2: Edit the BlockList to add a PracticeBlockProc Procedure

Add a new row to the BlockList Object and create a new PracticeBlockProc Procedure to represent the practice block.

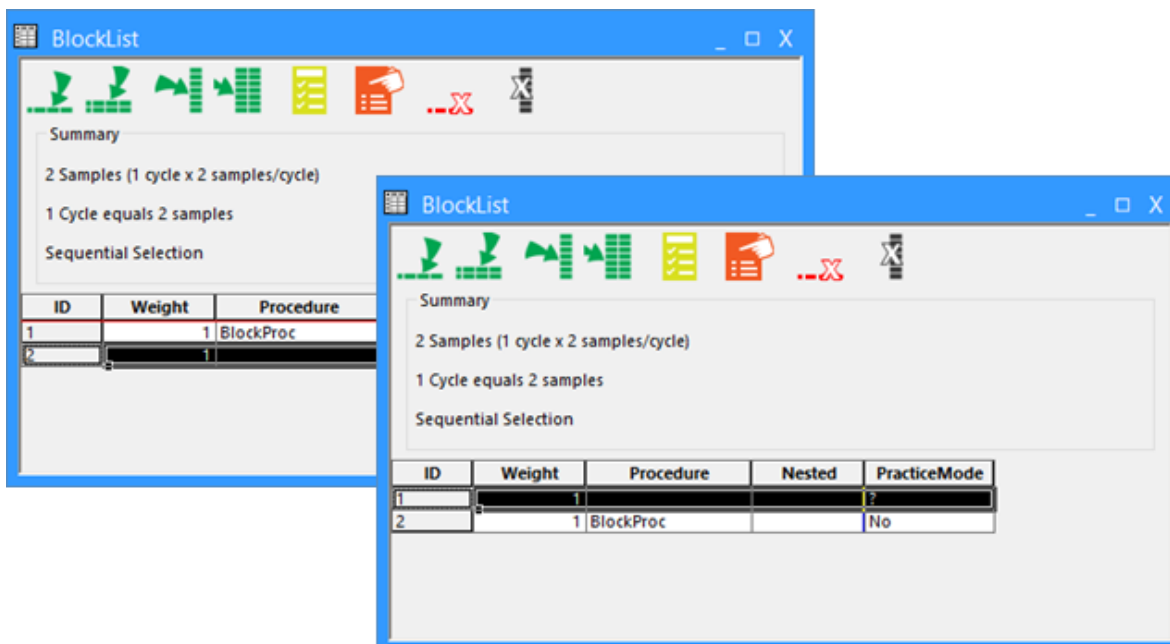
Now that the CellList is edited, you need to create a new level/row in the BlockList to represent the new block of practice trials. This new Procedure is named “PracticeBlockProc” and is used to execute the new block of practice trials. Once the Procedure is created, the existing PracticeMode attribute needs to be modified to “Yes” to log an indication that the block is practice trials.

When a List Object is configured to have selection be sequential, the order of the rows directly determines the sampling order. Therefore, PracticeBlockProc needs to be moved before the BlockProc so the practice occurs before the real trials. If you want to run multiple blocks of practice or real trials, you can edit the cells under Weight attribute as desired.

- 1) **Open** the **BlockList** Object in the workspace.
- 2) **Click** the **Add Level** button to add a new level/row to the **BlockList**.

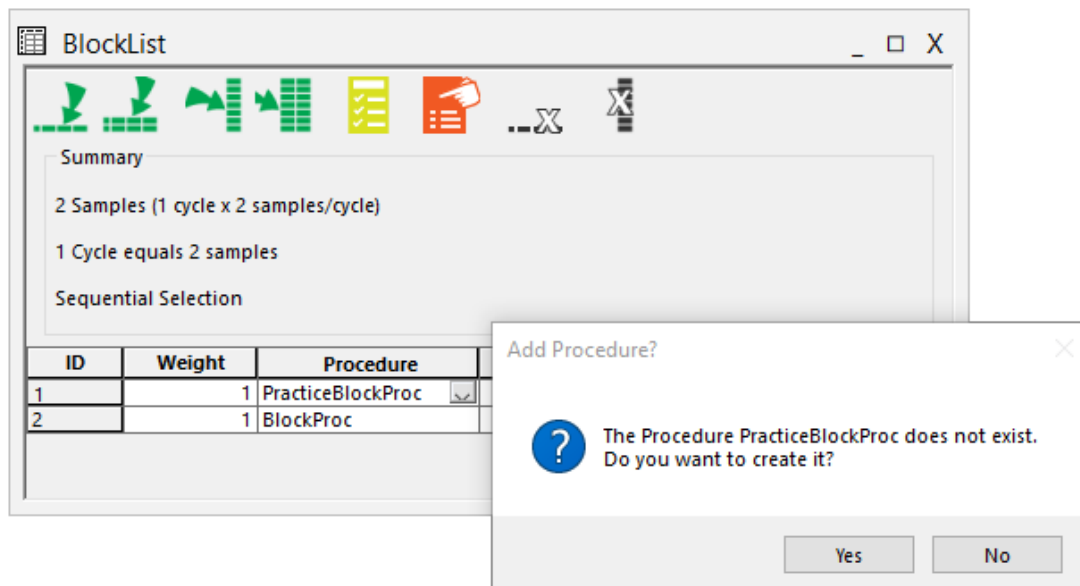


- 3) **Click** the **ID** row header to **select** the **second row**, **drag** the **second row** above the **first row** to **reorder** the blocks then **verify** that **newly added level** is the **first row**.

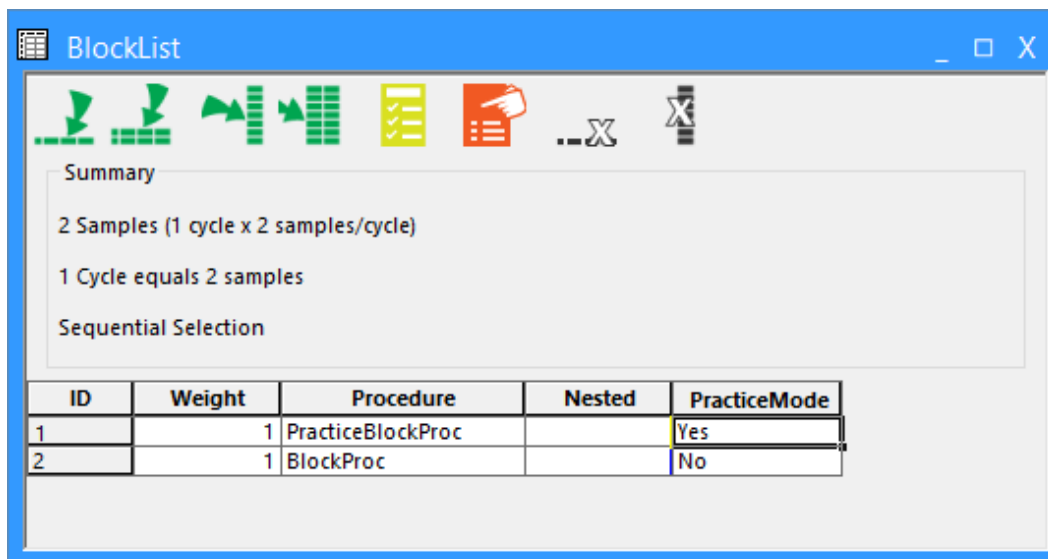


NOTE: During the drag operation, you will see a red line indicating where the selected row will be moved to when it is dropped.

- 4) **Click** in the cell under the **Procedure** attribute in the first row, **type PracticeBlockProc**, and then **press Enter**.
- 5) **Click** the **Yes** button to create a new **PracticeBlockProc** Procedure in the experiment.



- 6) **Click** in the cell under the **PracticeMode** attribute in the second row, **type Yes**, and **press Enter**.

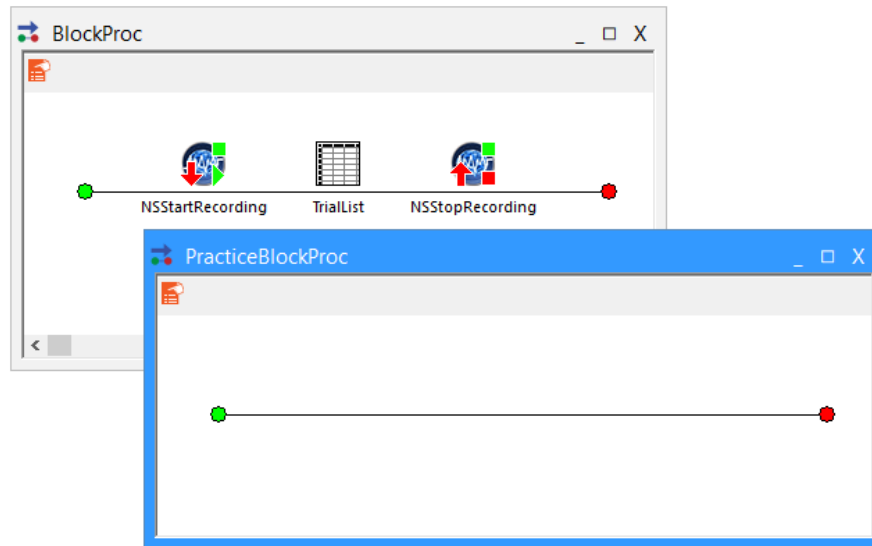


Task 3: Specify the PracticeBlockProc

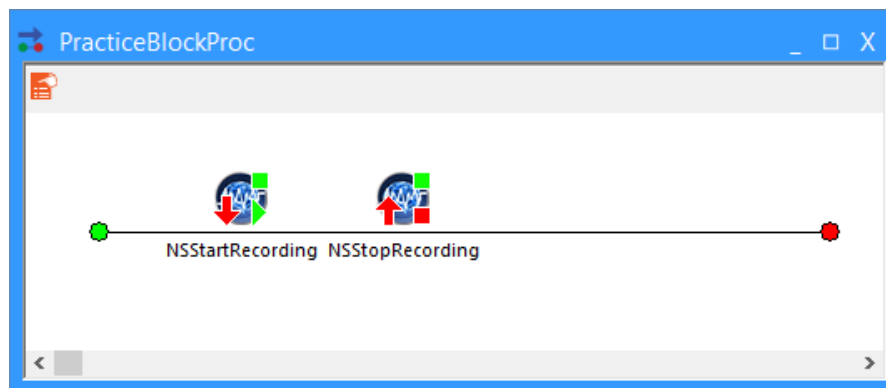
Add the required Net Station PackageCalls to the beginning and ending of the PracticeBlockProc.

Configure the PracticeBlockProc to start Net Station recording at the beginning of the practice block and stop Net Station recording at the end of the practice block. You can reuse the NSStartRecording and NSStopRecording Objects that are already being called from the original BlockProc by dragging and dropping them on PracticeBlockProc Object.

- 1) **Open BlockProc** and **PracticeBlockProc** in the workspace.



- 2) **Drag** the **NSStartRecording** and **NSStopRecording** Objects from the **BlockProc** and drop them as the **first** and **second** object in the Procedure respectively.



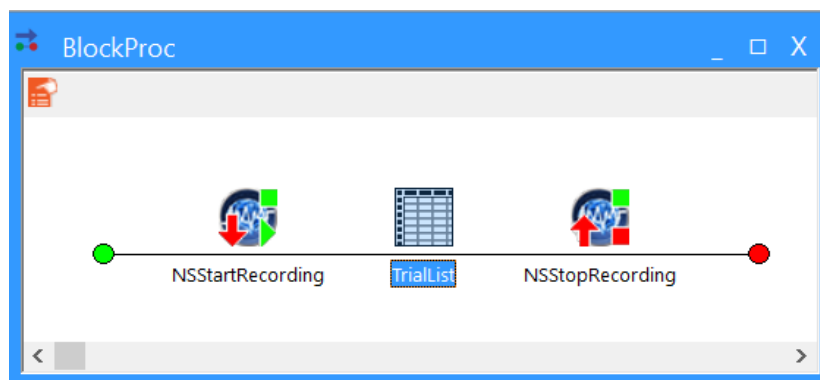
Task 4: Create and configure the PracticeTrialList

Create a new List to represent the practice trials by making a copy of the TrialList Object.

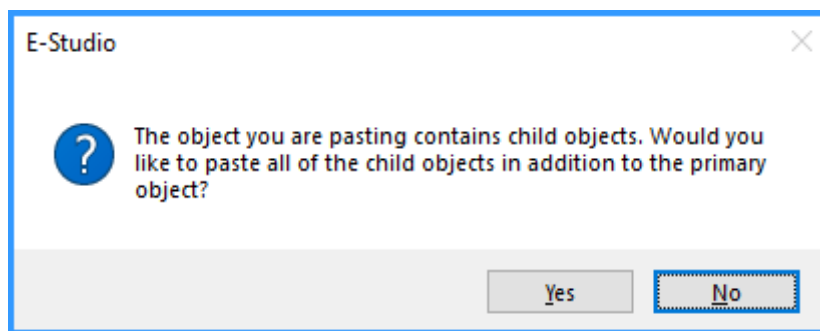
The new block of practice trials requires a unique List at the trial level since it references CellNumbers that are different from those used in the real trials. Because the existing TrialList Object is already configured appropriately for use with Net Station you need to add a separate List in the PracticeBlockProc.

The practice trials must reference the new cells that were created in the CellList (i.e., “redcar_prac” and “bluecar_prac”). Therefore, the cells under the CellNumber attribute needs to be edited to reference the new cell numbers of 3 and 4 respectively.

- 1) **Select** the **TrialList** Object within the **BlockProc** and **press Ctrl+C** to **Copy** the Object.

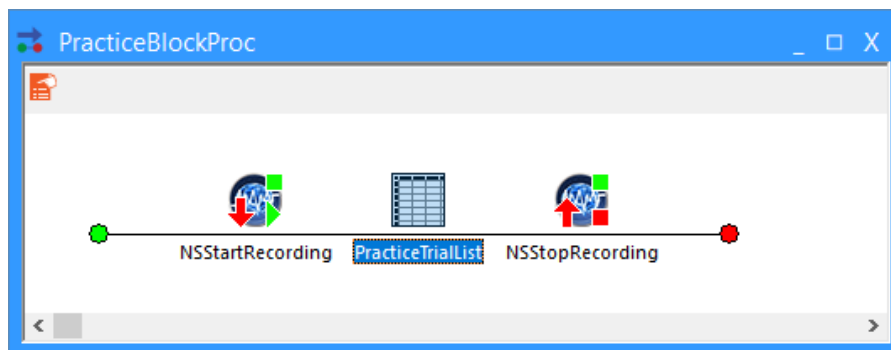


- 2) **Paste (Ctrl+V)** the **TrialList** in **PracticeBlockProc**.
- 3) **Select No** to the message about **pasting child objects**.



- 4) **Drag** the **List** you just pasted between the **NSStartRecording** and **NSStopRecording** Objects.

- 5) **Click** on the **List** you just moved (i.e., TrialList1) and **press F2** to **rename** the object as “**PracticeTrialList**”.



- 6) **Open** the **PracticeTrialList** Object in the workspace.
- 7) **Click** in each cell under **CellNumber** and edit the values to be “**3**” for Level 1 and “**4**” for Level 2.

PracticeTrialList

Summary

4 Samples (1 cycle x 4 samples/cycle)

1 Cycle equals 4 samples

Random Selection (Repeat After Reset Possible)

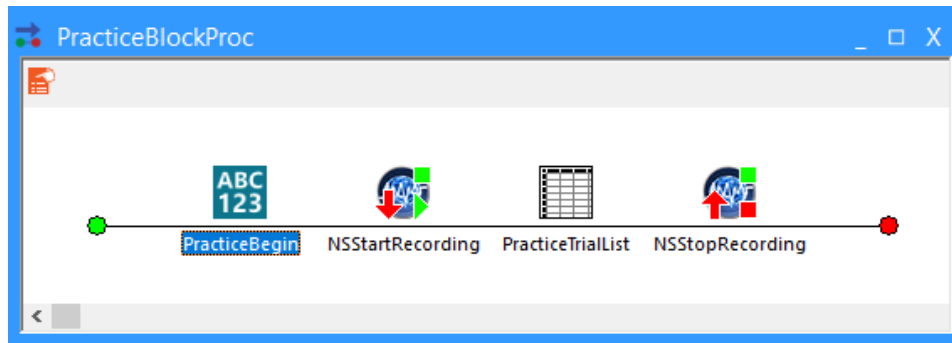
ID	Weight	Procedure	Nested	Stimulus	CorrectAnswer	CellNumber
1	2	TrialProc		RedCar.bmp	1	3
2	2	TrialProc		BlueCar.bmp	2	4

Task 5: Add an instruction display to begin practice

Add an Instruction screen to tell the participants that a practice trial is going to begin.

Whenever a block of practice trials is used in an experiment, it is good practice to add an instruction screen that states practice trials are about to commence. In this task, a TextDisplay is added that lets the participant know that practice is about to happen.

- 1) **Drag** a new **TextDisplay** Object from the **Toolbox** and **drop** it at the beginning of the **PracticeBlockProc**. **Rename (F2)** the object “**PracticeBegin**”.

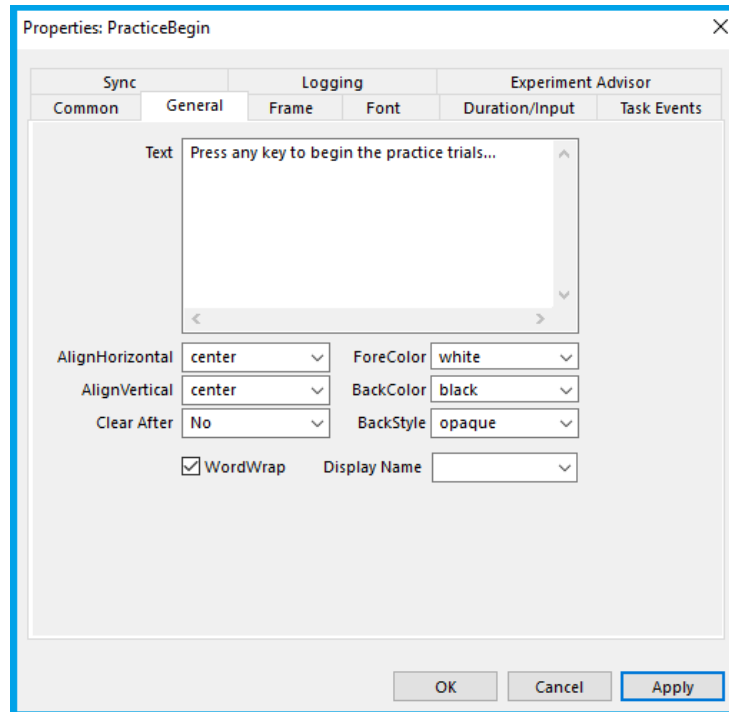


- 2) **Open** the **PracticeBegin** Object in the **workspace**. **Click** the **Properties** button to display the **Properties** dialog for the **PracticeBegin** Object.

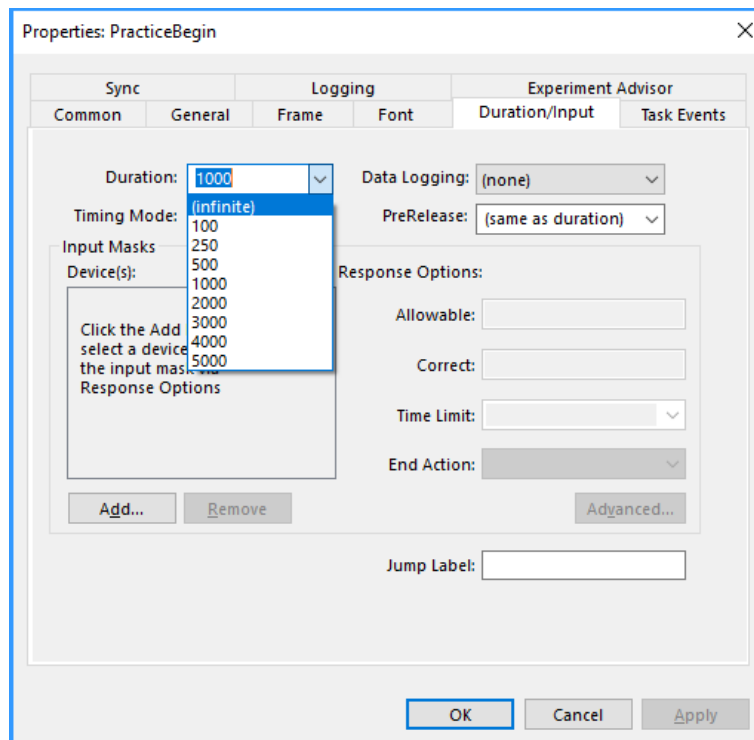


- 3) **Select** the **General** tab, then **type** the following in the **Text** field: “**Press any key to begin the practice trials...**”
- 4) **Select** “**white**” in the **ForeColor** field dropdown to set the text foreground color.
- 5) **Select** “**black**” in the **BackColor** field dropdown to set the text background color.

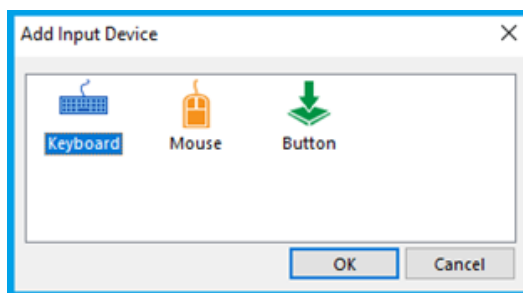
- 6) **Click** the **Apply** button to **save** the changes to this tab. **Verify** the changes with the **image** below.



- 7) **Select** the **Duration/Input** tab. **Select “(infinite)”** in the **Duration** field dropdown to set the duration of the object.



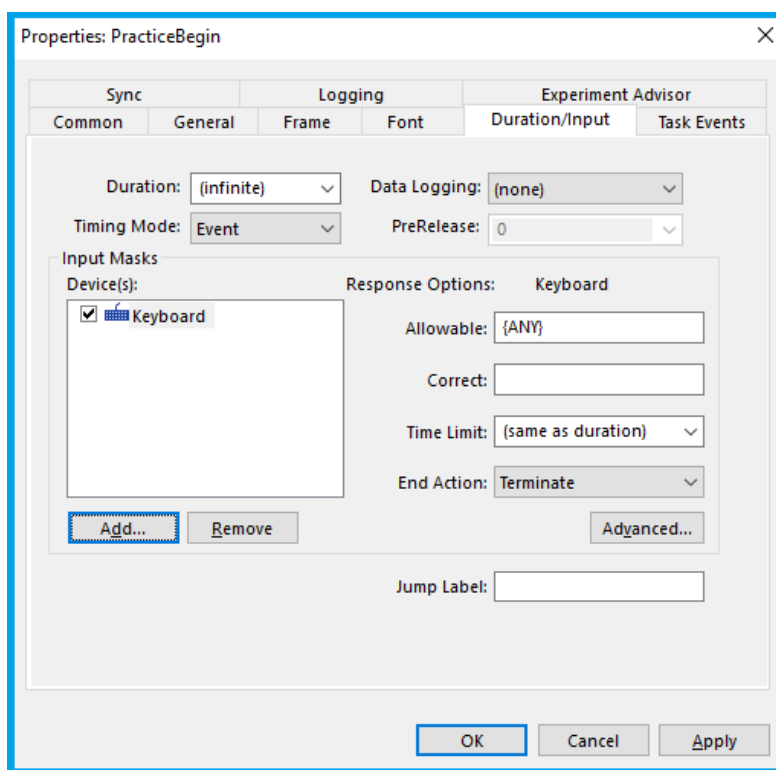
8) **Click** the **Add...** button to **add** an **Input Mask**. **Select** the **Keyboard** device.



9) **Click** the **OK** button to **add** a new Input Mask for the **Keyboard** device.

NOTE: By default, the Input Mask allows any key press to terminate the input.

10) **Click** the **Apply** button to **save** the changes to this tab. **Verify** the changes with the **image** below.

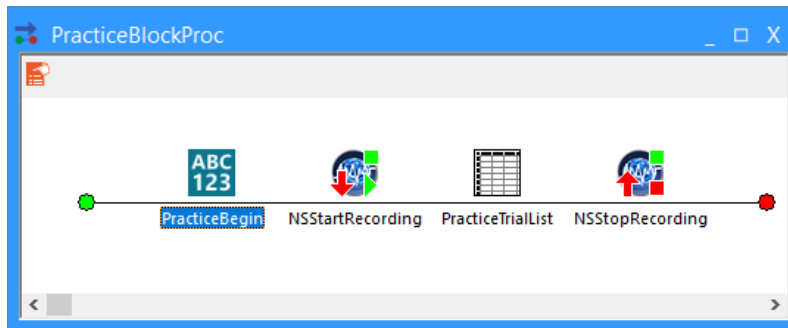


Task 6: Add an instruction display to end practice

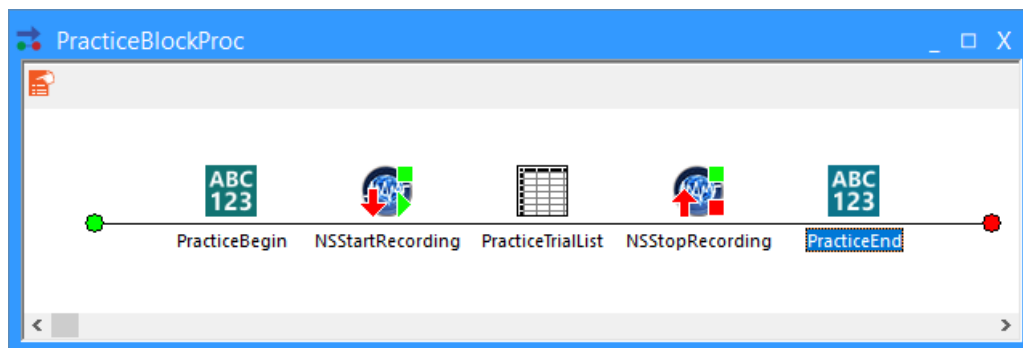
Add a new TextDisplay at the end of the practice block to notify the participant that the practice trials are complete.

At the end of the practice trials, a new TextDisplay Object needs to be added to notify the participant that the practice trials have been completed and the real trials are about to begin. Since the PracticeBegin Object is already configured appropriately for use in the experiment (e.g. foreground color, background color, and wait for a key press to continue), it can be copied, renamed, and edited for this purpose.

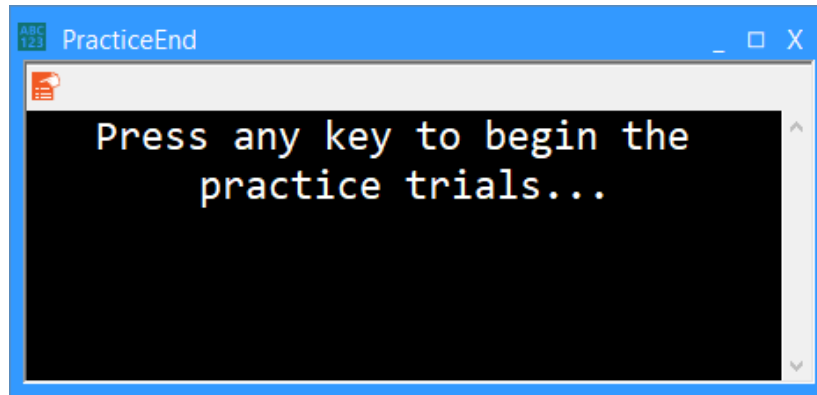
- 1) **Open PracticeBlockProc** in the workspace.
- 2) **Select** the **PracticeBegin** object within the **PracticeBlockProc** and **press Ctrl+C** to **Copy** the Object.



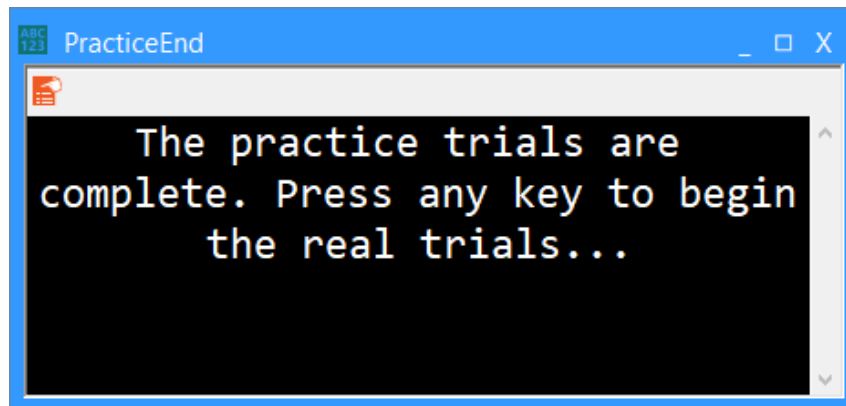
- 3) **Paste (Ctrl+V)** **PracticeBegin** in **PracticeBlockProc**.
- 4) **Select** the **PracticeBegin1** Object then **press F2** and **rename** the object as "**PracticeEnd**".
- 5) **Drag PracticeEnd** as the **last object** in the **PracticeBlockProc**.



6) **Open** the **PracticeEnd** Object in the workspace.



7) **Click** in **PracticeEnd** to **Edit** the text to say "The practice trials are complete. Press any key to begin the real trials."



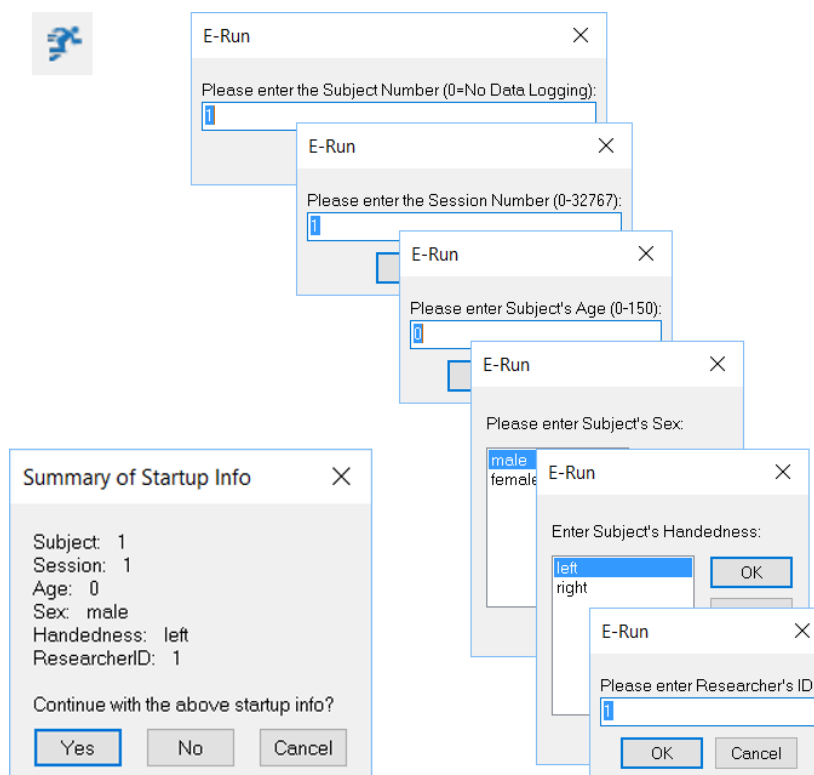
Task 7: Run the experiment

Run the experiment to verify that the new key is being sent to Net Station and that no runtime errors are generated.

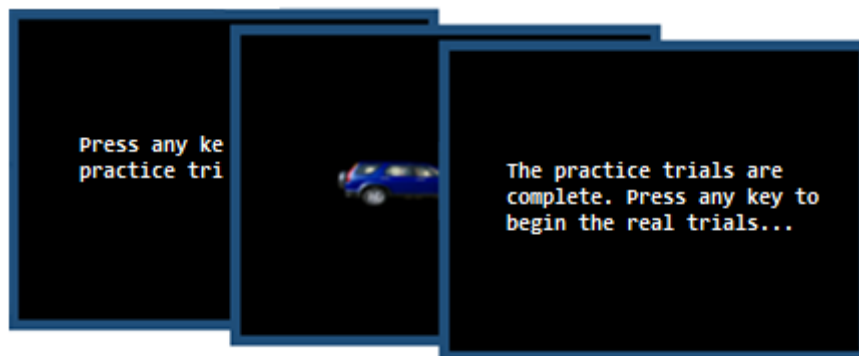
Your experiment should now be configured to contain a set of practice trials. Run the experiment to verify that the CellNumbers and CellLabels are correct and that the new TextDisplays appear at the appropriate times.

NOTE: When using E-Prime and the Net Station system you must have a two-computer setup to collect EEG data. E-Prime and E-Prime Extensions for Net Station must be installed on one computer and the Net Station system must be installed on another system. The two computers must be connected with a network hub or switch to communicate with each other. For information on connecting the machines, please refer to [Chapter 1, Section 1.3: Network Configuration](#).

- 1) **Click** the **Run icon** on the toolbar to generate and run the experiment locally.
- 2) **Press Enter** to **accept** the **default values** for each of the initialization prompts presented.



3) **Observe** the **practice trials** to **verify** the **experiment** is **functioning** correctly.



4) **Observe** the **stimulus presentation sequence** to **verify** the **experiment** is **functioning** correctly and **completes** with **no errors** being generated.



3.6. Tutorial 4: Timing Test

NTP timing method is very efficient and reliable, but it is always prudent to verify that your new system is functioning within specification before you begin collecting any real data. Running the sample NSTimingTest.es3 experiment provided with the E-Prime Extensions for Net Station 3.0 installation will help you check your system's visual timing accuracy. You should also repeat this process to verify timing in each new experiment you create, prior to running participants. For more information on timing tests for various media, contact EGI's [Knowledge Center](#).

Once you record test data using the NSTimingTest.es3 experiment and your EGI AV Tester, you can review the timing results by using the EGI Event Timing Tester application provided with Net Station.

The Net Amps amplifiers have a digital input plug in the back and an adapter that can connect with a DB9 cable. This DB9 cable, called the Digital Input or DIN cable, allows DIN events output from the AV Tester device to be recorded in the Net Station file.

Please refer to the Net Station, System, and AV Tester manuals that came with your EGI system for more information on how to use those EGI products, or contact EGI Support. If you have questions regarding how to set up and run experiments in E-Prime, review [Chapter 1, Section 1.3: Network Configuration](#).

Run the NSTimingTest.es3 Sample experiment:

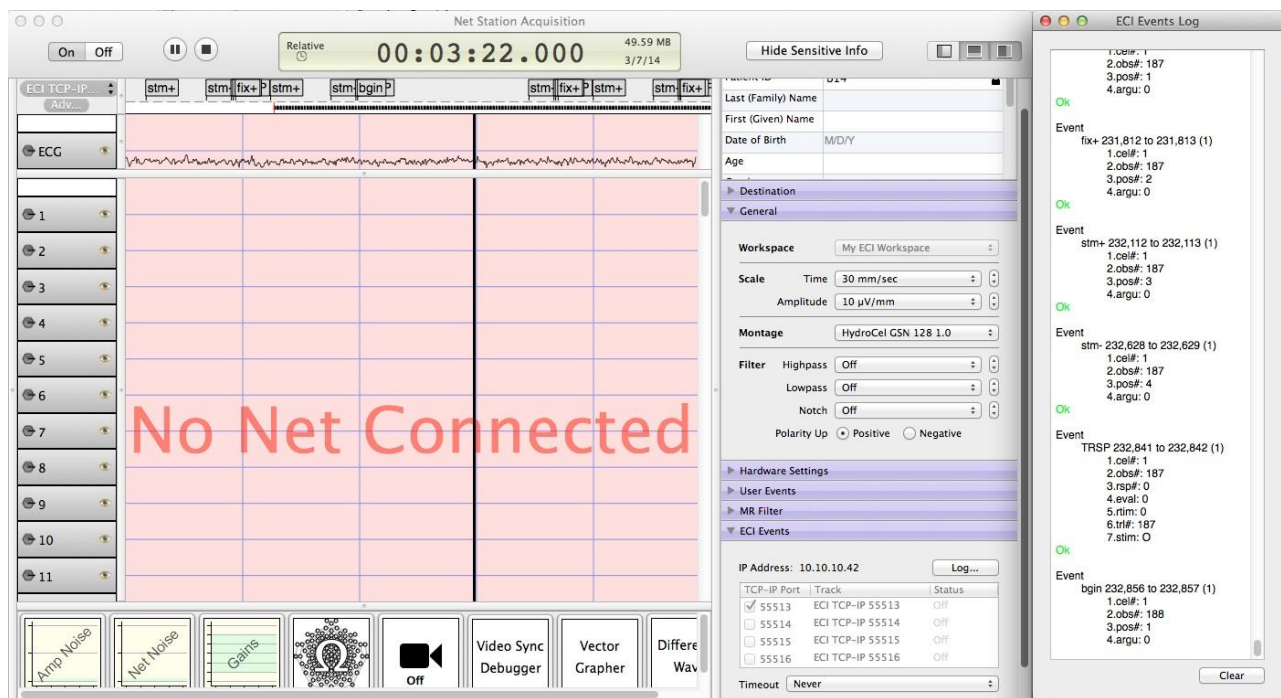
- 1) Once you have set up your AV Tester equipment and amplifier according to the EGI manuals, **launch** Net Station Acquisition on the Mac and configure the ECI controls to record the correct digital input (DIN) value. You can also open the ECI log window from this control to verify proper conveyance of events from E-Prime. You should have the ECI Log and the ECI panels open. Use default port 55513.
- 2) If you are running a visual timing test, make sure that you have enabled **only** DIN 3 in the ECI Controls. For an auditory experiment, select DIN2 only. This will eliminate any unneeded information and provide a clean DIN event in the recording file.

NOTE: *The three AV Tester DIN outputs differ in sensitivity. Under most circumstances, DIN 3 is appropriate for visual stimuli, while DIN 2 is better for auditory stimuli. Some monitors, such as CRT displays, may require using DIN 1. Some experimentation may be necessary.*

- 3) **Turn on Streaming** in **Net Station Acquisition**.
- 4) On the PC, launch the appropriate timing test for the media you are using in your experiment. This Tutorial uses the E-Prime Sample experiment "NSTimingTest.es3" located in the <My Experiments> folder. For additional timing test experiments for use with different media, contact EGI's [Knowledge Center](#).
- 5) **Specify "socket"** and your Mac computer's IP address ("e.g., "10.10.10.42") in the NSInit parameters.

- 6) **Select E-Run > Run** from the application menu (or click the Run button) to generate and begin running the experiment.
- 7) **After** the entering **Startup Info** in the message boxes, **line up** the **AV Tester Optical Sensor** with the white Stimulus square as prompted by the experiment. Make sure that the **sensor is even** and as **close as possible** to the **monitor screen**, but **avoid touching the display with the sensor**.
- 8) **Continue running** the **experiment**. You can monitor events in **Net Station Acquisition's ECI Event Track and ECI Event Log** as the experiment progresses.
- 9) The **NSTimingTest.es3** experiment **flashes** a white Stimulus rectangle on the screen every second for one hour. Each Stimulus flash corresponds with a "stm+" event.

With the Net Amps amplifiers, the AV Tester causes DIN events to be placed in the recording each time a flash of the Stimulus rectangle occurs, which can be compared to the events E-Prime sends through the ECI to verify accurate timing of events.



After you have successfully run NSTimingTest.es3, you can quit Net Station Acquisition and review the timing results with the Event Timing Tester included in your Net Station installation.

How to review a DIN file with Event Timing Tester

If you used the Net Amps for your testing and recorded DIN events in your file, you can use the EGI Event Timing Tester application to review your timing results.

- 1) On the Net Station Mac, Launch the EGI Event Timing Tester (ETT) application. This application can be found in the Net Station application folder.
- 2) **Choose “Open”** from the **ETT File menu**, and **select** your **timing test recording** using the file browser dialog. **ETT** will **load** event information from your file and present a File panel.
- 3) From the **“Stim Event Code”** pop-up menu, **select “stm+”**. This is the stimulus event from the NSTimingTest.es3 experiment.
- 4) From the **“DIN Event Code”** pop-up menu, **select “DIN1”**, or whichever **DIN code** you used in Net Station.
- 5) **Click** the **“Create Timing Table”** button to **generate** a table of event information. Each row represents a single trial (flash of the Stimulus rectangle) from the NSTimingTest.es3 experiment. The columns contain specific details about each trial.

The screenshot shows the EGI Event Timing Tester (ETT) application window. At the top, the file name 'fc1_20140307_040802.mff' is displayed. Below the file name, there are two dropdown menus: 'Stim Event Code' set to 'stm+' (39 events) and 'DIN Event Code' set to 'DIN3' (41 events). A 'Create Timing Table' button is located to the right of these menus. Below the menus is a table with 6 columns: Trial, Cell, Observation, Rel. Time, Offset, and Elapsed Time. The table contains 11 rows of data. Below the table, there are two sections: 'Compute Jitter From' with radio buttons for 'Average' (selected) and 'Median', and two tables for 'Offsets' and 'Elapsed' statistics.

Trial	Cell	Observation	Rel. Time	Offset	Elapsed Time
1	1	1	00:00:01.948	12	0
2	1	2	00:00:03.012	12	1,064
3	1	3	00:00:04.094	14	1,082
4	1	4	00:00:05.175	13	1,081
5	1	5	00:00:06.240	12	1,065
6	1	6	00:00:07.321	11	1,081
7	1	7	00:00:08.386	14	1,065
8	1	8	00:00:09.450	14	1,064
9	1	9	00:00:10.515	13	1,065
10	1	10	00:00:11.580	12	1,065
11	1	11	00:00:12.644	12	1,064

Compute Jitter From: ☒ Average ☐ Median

Offsets		Elapsed	
Average	12.48718	Average	1,071.658
Maximum	14	Maximum	1,098
Median	13	Median	1,065
Minimum	11	Minimum	1,064
+/-0	9 (23.08%)	+/-0	0 (0.00%)
+/-1	21 (53.85%)	+/-1	0 (0.00%)
+/-2	9 (23.08%)	+/-2	0 (0.00%)
+/-3	0 (0.00%)	+/-3	0 (0.00%)
+/-4	0 (0.00%)	+/-4	0 (0.00%)
+/-5	0 (0.00%)	+/-5	0 (0.00%)

- 6) **Review** the column **“Rel. Time” (Relative Time)**. This is the time, in milliseconds, from the beginning of the experiment to each stm+ event.

- 7) **Compare** the “**Rel. Time**” column to the “**Elapsed Time**” column. Elapsed Time is the time duration, in milliseconds, of each trial. The Elapsed Time should equal the difference between the Relative Time of the current and previous trials. In this experiment, that would be roughly **1000** ms.
- 8) Most importantly, **review** the “**Offset**” column. The Offset value listed for each trial is the number of milliseconds between the stim+ and DIN events in Net Station.

The Offset value represents the difference between the timestamp E-Prime assigned to the stim+ event it sent to Net Station, and the timestamp Net Station assigned the DIN event it created when the AV Tester detected the flash of the Stimulus rectangle.

- 9) Since E-Prime and Net Station both get time from the amplifier clock, the Offset for all trials should not vary by more than a millisecond or two for the entire recording. Look carefully through the ETT table for your test file and verify that this is true.

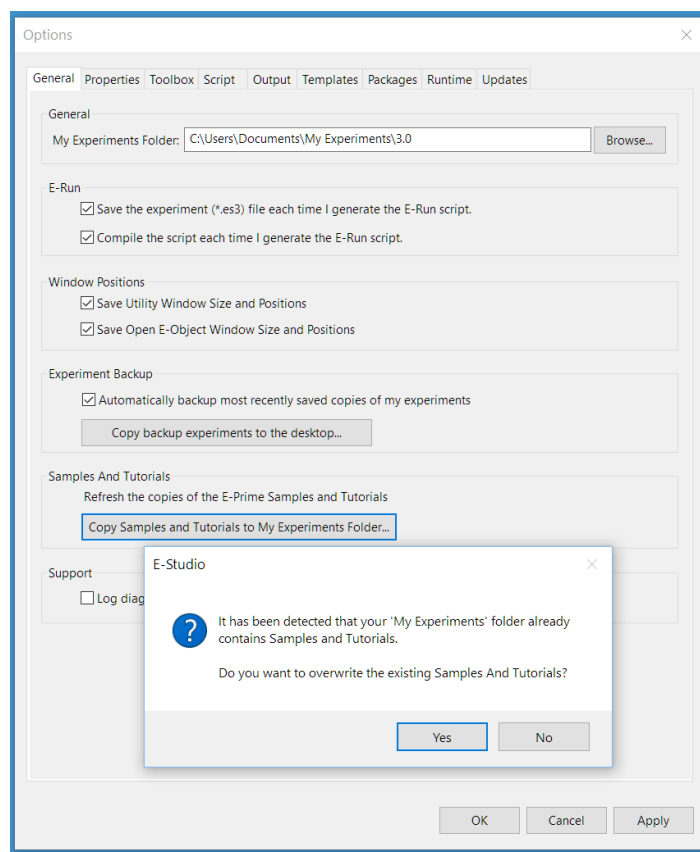
4. Troubleshooting

The following topics are common issues and solutions to help you when implementing and testing experiments with Net Station. In the case that one of the topics cannot fix your issue, Psychology Software Tools recommends saving and closing the experiment. If any issue persists please contact [Product Service and Support](#).

4.1. Cannot locate E-Prime Samples and Tutorials

When you launch E-Studio for the first time after installing E-Prime Extensions for Net Station, you are prompted to update the Samples and Tutorials experiments. If you choose to not update now, you can force another update of the Sample and Tutorials at any time from within E-Studio. Forcing a refresh of the Samples and Tutorials can be helpful if you cannot locate the Samples and Tutorials. This feature is also helpful if you make changes to a Sample or Tutorial but then wish to revert to the original Sample or Tutorial.

- 1) **Open E-Studio** and **select Tools > Options...** from the application menu bar.
- 2) **Click** on the “**Copy Samples and Tutorials to <My Experiments> folder...**” button at the bottom of the Options dialog.
- 3) **Click** on **Yes** if prompted to overwrite the Samples and Tutorials.

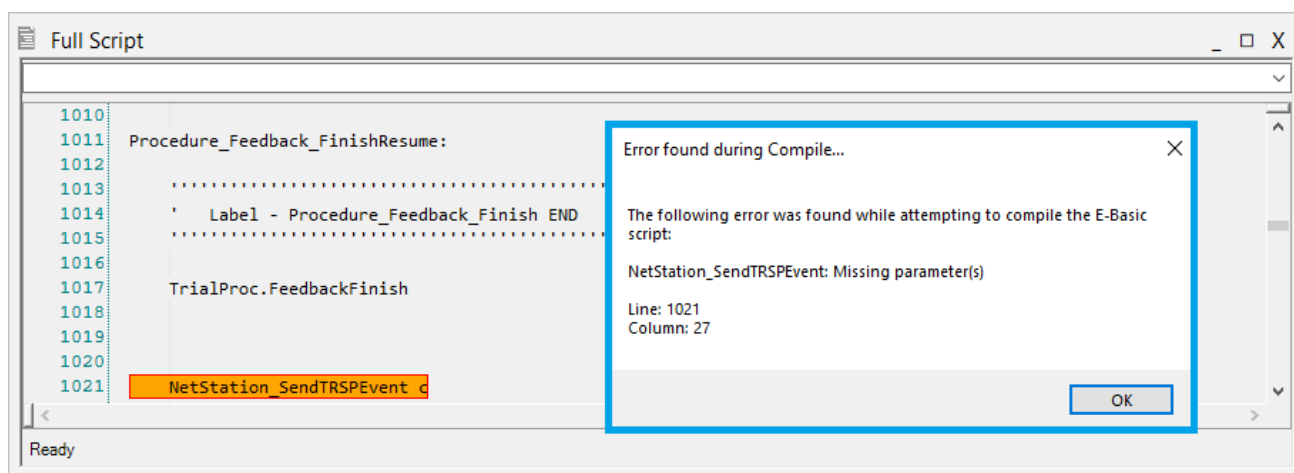


4.2. Missing Parameter Errors

Some PackageCalls require additional information specified in the parameters. For example, NSSendRespEvent requires an additional parameter: the name of the ResponseData/Stimulus Object.

When designing an experiment, be sure to check the PackageCall Properties for any additional parameters. When a PackageCall has “<>” for the parameters, then more information is needed. The angle brackets (“<>”) are placeholders for more information to your machine or experiment configuration.

Some PackageCalls require additional information for a parameter, but the parameter is not overtly specified in angle brackets. Whenever you generate the experiment and do not specify a parameter that was not in angle brackets an error occurs. In the example below, the error message indicates what Routine generated the error, however, this might not always be the case (e.g., a more generic error occurs). You can find out which Routine generated the error by looking in the Full Script window. The Routine with the missing parameter is highlighted. Additionally, you can scroll up in the Full Script window to determine which Procedure the error occurred on.

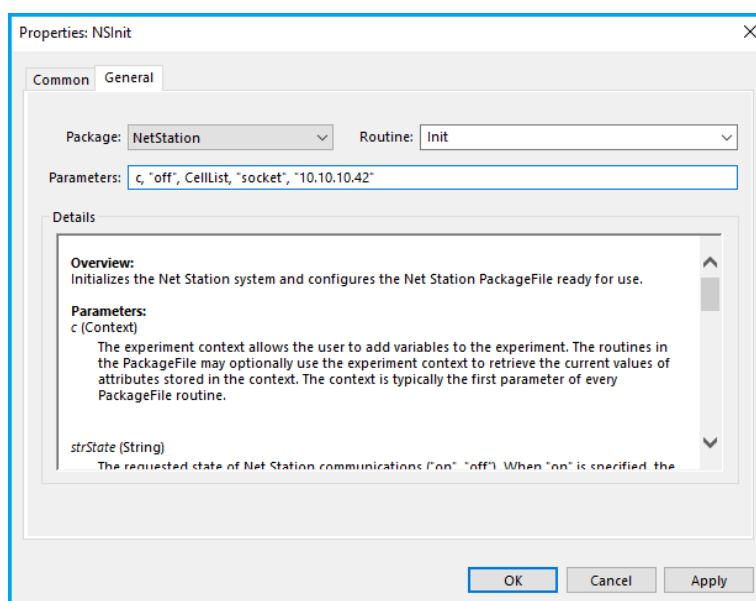


NOTE: When designing your experiment, we recommend reading through the Details of the PackageCall's properties. **Details** provides descriptions and remarks of all required and optional parameters.

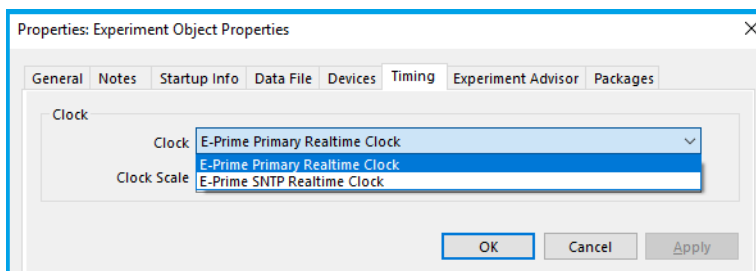
4.3. Running an Experiment Without Net Station Connected

Occasionally, you may want to run your Net Station-enabled experiment without the Net Station hardware attached and without calling the Net Station PackageCalls. It is often beneficial to run the experiment under those conditions whenever you are designing and testing your experiment. This can be helpful for determining if your critical stimuli are not presenting correctly or if there is a runtime error in your experiment. Whenever you want to run with Net Station communications enabled, you need to alter the PackageCall NSInit in your experiment. Also, you need to use the E-Prime Realtime clock rather than the SNTP clock.

- 1) **Open** the PackageCall **NSInit**.
- 2) **Change** the **strState** from “on” to “off”.



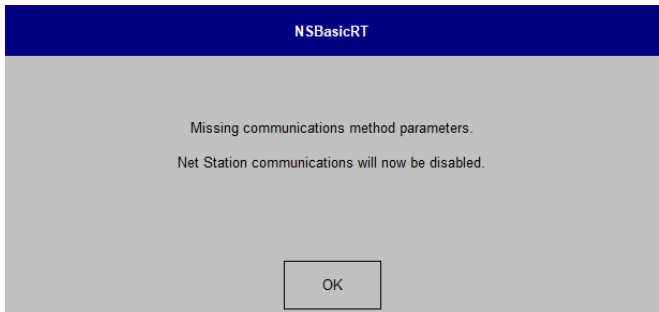
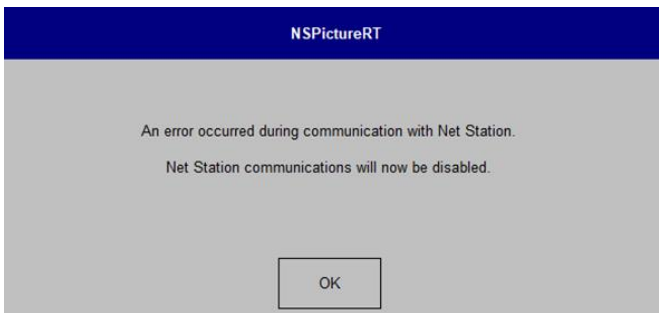
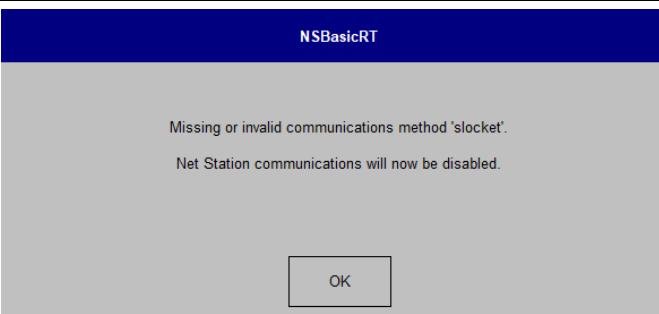
- 3) **Double click** the **Experiment Object** to **open** it and **Navigate** to the **Timing** tab.
- 4) **Switch** the **Clock** to the **E-Prime Primary Realtime Clock**. **Click OK** to accept the changes.

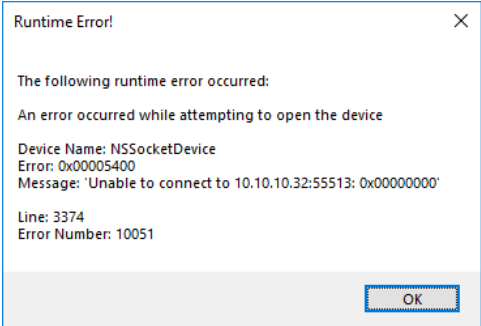


NOTE: The E-Prime Primary Realtime Clock is the default Clock in E-Prime and may already be selected in some cases.

4.4. Net Station Communication Errors

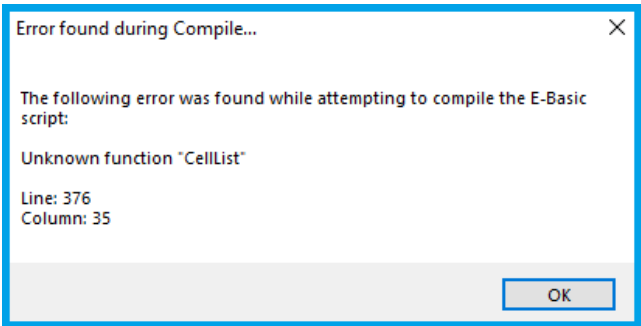
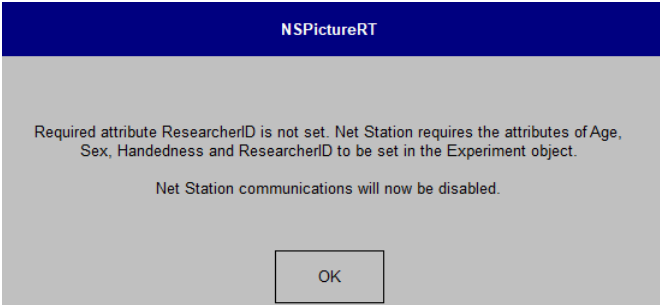
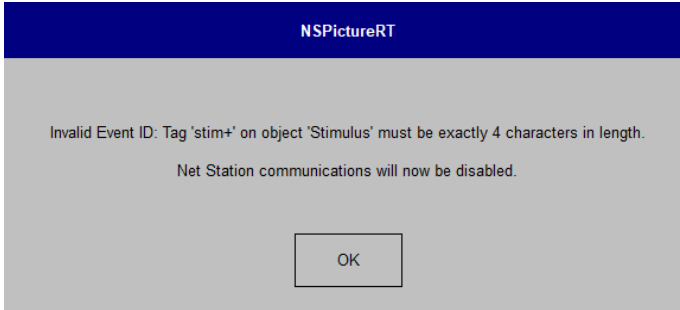
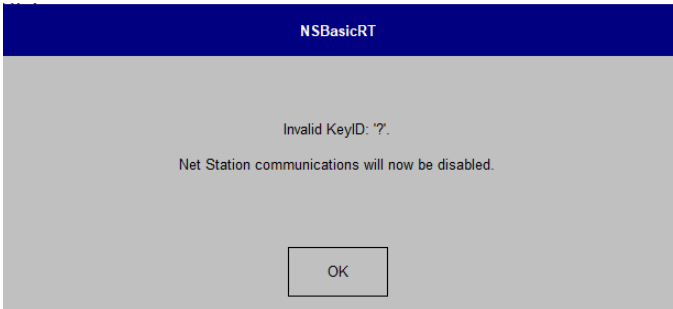
During the testing your Net Station-enabled experiments, you may encounter a communication error with Net Station and E-Prime. The images below show the various errors you may encounter and the reasons why they happen.

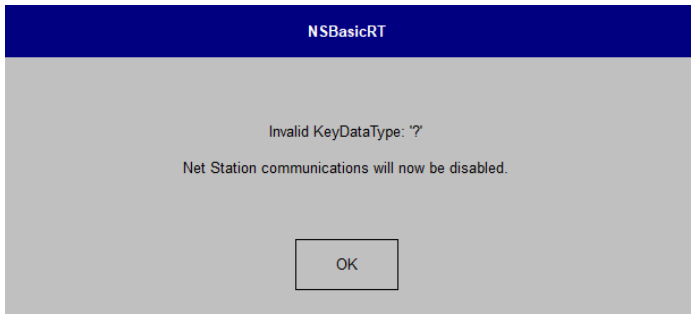
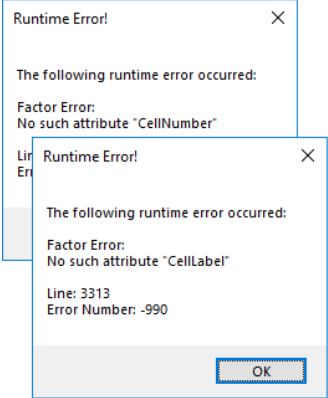
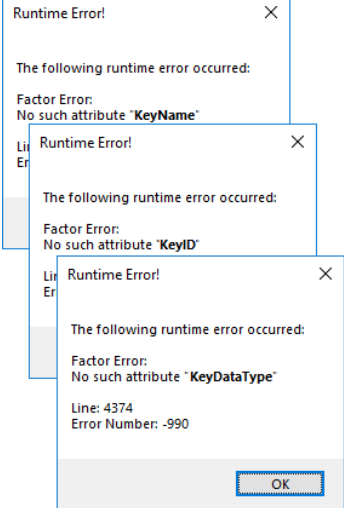
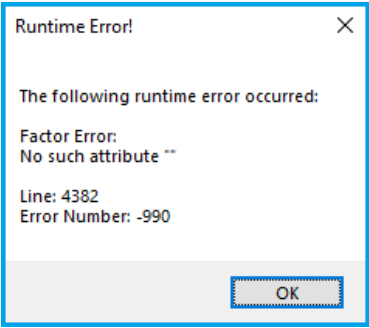
Error	Reason(s) why error occurred
	The network connection was interrupted during the test session (e.g., the network connection from the hub/switch was disconnected)
	varCommMethod is not specified in Startup Info, NSInit, or the NETSTATION.ini file. varCommMethodParams is not specified in Startup Info, NSInit, or the NETSTATION.ini file. Refer to Task 6: Add the NSInit PackageCall to Initialize the System for more information.
	Net Station Acquisition streaming is not set to "on".
	varCommMethod is specified incorrectly in Startup Info, NSInit, or the NETSTATION.ini file. Refer to Task 6: Add the NSInit PackageCall to Initialize the System for more information.

Error	Reason(s) why error occurred
	• An incorrect IP Address is specified in Startup Info, NSInit, or the NETSTATION.ini file. • The application Net Station Acquisition is not opened on the Net Station computer.

4.5. Net Station Non-Communication Errors

During the testing your Net Station-enabled experiments, you may encounter an error that is not a communication error with Net Station and E-Prime. These errors can often be solved by fixing an issue with the design of your experiment.

Error	Reason(s) why error occurred
	Either CellList does not exist or the name of the List is incorrect (e.g., CellList1) in the Unreferenced Objects. Refer to Task 14: Add a CellList Object for information.
	One of the required Startup Info parameters required by Net Station is not enabled in the experiment. Refer to Task 4: Configure the Experiment Startup Info for more information.
	The Tag Property on the Object or Event of interest was not set with a proper unique four-character code. Refer to Task 13: Assign a unique four-character code to each object of interest for more information.
	The KeyID value set in the KeyList is invalid. Refer to Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList for more information.

Error	Reason(s) why error occurred
	The KeyDataType value set in the KeyList is invalid. Refer to Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList for more information.
	The CellNumber or CellLabel Attribute does not exist in the CellList. Refer to Task 15: Add the required CellNumber and CellLabel attributes to the CellList for more information. NOTE: <i>CellNumber must be an integer value or you receive a Type mismatch error.</i>
	Either the KeyName, KeyID, or KeyDataType Attribute does not exist in the KeyList. Refer to Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList for more information.
	This is a generic E-Prime error but could indicate that the KeyName value set in the KeyList is invalid. Refer to Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList for more information.

5. Appendix A: Samples and Tutorials

The following table contains a list of all the Samples and Tutorials included with E-Prime Extensions for Net Station. The folder the experiment is in and a short description are included.

Sample / Tutorial	Folder	Description
NSBasicRT	Samples	NSBasicRT is the Net Station-enabled version of the E-Prime BasicRT sample experiment with an added block of practice trials.
NSPictureRT	Samples	NSPictureRT is the Net Station-enabled version of the E-Prime PictureRT sample experiment with an added block of practice trials. Additionally, it is the experiment created after completing EENS Tutorial's 1 – 3.
NSPulseTrainRT	Samples	NSPulseTrainRT is a version of the NSBasicRT experiment with added support for pulse trains. It demonstrates loading a train, starting a train, and finishing a pulse train during practice trials and real trials.
NSTimingTest	Samples	NSTimingTest is an experiment used to run a timing test and compute a clock scaling factor for Net Station.
VTD	Samples	The Visual Target Display (VTD) experiment is a visual oddball task enabled for use with Net Station. It contains a block of practice trials and a block of real trials.
PictureRT	Tutorials	PictureRT is the same experiment as the one installed with E-Prime. It serves as the starting point for EENS Tutorial's 1 – 3.

6. Contact Information



For additional information or support contact us at:

Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone: 412-449-0078
Fax: 412-449-0079
www.pstnet.com

For Product Service and Support:
<https://support.pstnet.com>