



DirectRT v2002 Research Software User's Guide and Reference

Copyright © 1995-2002 Empirisoft. All rights reserved.

COPYRIGHT

Copyright © 1995-2002 Empirisoft. All rights reserved.

Portions of the software described in this document copyright Microsoft Corporation. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written consent of Empirisoft.

Empirisoft Corporation
226 E 95th Street, Suite 506
New York, NY 10128

Web site: www.empirisoft.com.

Technical support is available from the web site above or by emailing Empirisoft at support@empirisoft.com.

Questions, suggestions, software bugs, and other feedback should be directed to support@empirisoft.com.

TRADEMARKS

DirectRT, MediaLab Research Software, the MediaLabRats, and the MediaLab Experiment Editor, are trademarks of Empirisoft. All other products mentioned are registered trademarks or trademarks of their respective companies.

Last modified: November 19, 2001.

Table of Contents

DirectRT

DirectRT Features.....	4
Setup	7
Introduction and Key Concepts	9
What's New in DirectRT v2002?	12
Main Menu Commands.....	14

Input Files

Creating Input Files	16
File Format.....	16
Block	17
Trial	18
Optional Fields	18
RBG - Randomization Between Groups.....	20
RWG - Randomization Within Groups	21
Style	22
Stim	23
Advanced Stimulus Features	27
Notes About Image Files	29
Loc - Stimulus Location	32
Time.....	35
Skipping	39
Jumping.....	41

Style Editor

About Styles and the Style Editor	42
Main Style Window	43
Style Menu Commands	43
Code - Assigning a Reference Number	44
Name	45
Back - Background Colour	45
Font Settings	47
Align - Stimulus and Text Alignment.....	47
MinMax - Setting RT Limits.....	48
Options.....	49
Messages.....	49
Screen Size.....	50
Trial Interval	50

Running an Input File

Running from the Main Program.....	52
Running from a Command Prompt.....	53
Running From MediaLab	53

Output Files

Overview of Data Output.....	55
Short Data Files	56
Log Files - More Data!.....	57

DirectRT Samples

About the Samples	59
Bare Bones.....	61
Text.....	61
Images.....	62
Text Files.....	62
StimLists	63
Setting Valid Keys	64
Changing Styles	64
Basics.....	65
More Stim Per Trial	65
The Basics of Style	65
Locating Stim	66
Compound Screens.....	67
Formatting Text Files	67
Using Optional Fields.....	68
Ordering Trials.....	69
RWG	69
RBG	69
RBG + RWG.....	70
Skipping	70
Jumping	71
Applications.....	72
Pretesting Images.....	72
Standard Priming.....	72
Priming with Images	73
Assessing Modern Racism	73
Other Features.....	74
Open Ended Responses	74
Open Ended Responses as Stimuli	74
Joystick and Mouse Input	76
Joystick Continuous	77
Require Correct Responses	79
Key Release RTs	80
Clear Screen During an RT.....	81
Key Press Feedback	82
Multiple RTs	83
Using Previous Stim	84
Play Sounds.....	86
Voice Responses.....	88
TTL Signals	91
IAT	93
Advanced Graphics	94

Welcome to DirectRT's help documentation! This manual is designed to be as painless as possible. If you are new to DirectRT, we suggest you start by running some of the samples to see what DirectRT can do. You can do this by going through the on-line tutorials (click 'DirectRT Samples' from the DirectRT Help menu) or you can access the samples directly just as if they were real experiments. All of the on-line tutorial instructions are printed at the end of this manual in case you prefer a hard copy reference. Note that the 2002 version comes with 15 new samples that demonstrate most of the new features. These new samples are not included in the basic tutorials but are included in the DirectRT samples folder titled 'Other Features'.

If you have a question that this manual does not answer, feel free to contact us any time at support@empirisoft.com or visit our website at <http://www.empirisoft.com>.

Thanks for trying DirectRT!

A handwritten signature in blue ink, appearing to read "BJ", enclosed within a light gray rectangular border.

Blair Jarvis
Author, DirectRT and MediaLab
jarvis@empirisoft.com

The Help documentation is divided into the following topics:

- DirectRT Features (p 4)
- Setup and System Requirements (p 7)
- Introduction and Key Concepts (p 9)
- Main Program Window and Menu Commands (p 14)
- Creating Input Files *** (p 16)
- About Styles and the Style Editor (p 42)
- Running an Input File (p 52)
- Overview of Data Output (p 55)
- Samples, Samples, Samples (p 59)

DirectRT Features

The following outlines most of the features and functions that DirectRT provides. Where useful, page numbers are provided to related 'how to..' documentation. Note that if you are already experienced with DirectRT v2000, you may find it useful to jump to the list of new features offered in version 2002.

Response Times

- Timing resolution of 1 millisecond.
- Response timing is synchronized with the screen display so timing always begins when the screen first begins to draw (eliminates 10-17 milliseconds of random error).
- Works with Windows to pause all other system events while timing is taking place. With everything else put on hold, the highest processing priority possible is given to DirectRT during the critical time intervals on each trial.
- Uses Microsoft's DirectX to gain the fastest access possible to input events from the keyboard, mouse, joystick and soundcard.
- Optionally define minimum and maximum response times and deliver custom message screens for each when violated. (p 48)
- Collect multiple response times on a single trial. (p 83)
- Collect key release response times in addition to key press times. (p 80)
- Define valid keys for any response. (p 35)
- Define 'correct' key for any response. (p 35)
- Optionally require a correct response. (p 79)

Input Options

- Keyboard, mouse, and joystick input for single keypress or button responses. (p 35)
- Short open ended (fill-in-the-blank) responses. (p 74)
- Voice responses via microphone connected to your soundcard. (p 88)
- Option of saving voice responses as wav files on your hard drive. (p 88)
- Continuous joystick motion. (p 77)
- TTL signals can be read from a defined port to gather data from external equipment. (p 91)

Stimulus Presentation

- Create simple or complex visual displays with single or multiple images and text.
- Create as many different displays as you need for any given trial.
- Play sound files. (p 86)
- Can handle multiple simultaneous and/or sequential sound files, with control over left/right panning and frequency (Hz). Easily synchronized with visual stimuli.

- Send precisely synchronized TTL signals to communicate with external equipment (e.g., another computer controlling physiological equipment). (p 91)
- Present subject's own responses to open ended questions as stimuli in the same session. (p 74)
- Easily create full screens of instructions with formatted text. (p 23)
- Create multiple style settings with choices of fonts and display colors. (p 42)
- Present full colour graphics (p 23) that can be as large as full screen and in any resolution from 640x480 to 1024x768. (p 42)
- Present multiple images and/or text in succession or simultaneously. (p 23)
- Superimpose custom graphics you make (e.g., special rating scales) over any other image or text. Great for pretesting and rating visual stimuli. (p 35)
- Capture screens from Microsoft PowerPoint and other presentation software and present them as stimulus or instruction screens in your experiment. (p 31)
- Precisely locate images and text with flexible and easy to use alignment options. (p 32)
- Display of stimulus screens is synchronized with the monitor's refresh rate resulting in highly controlled presentation rates. (p 35)
- Enter stimulus presentation times in milliseconds (p 35). DirectRT will determine the screen refresh rate and will present the stimulus for the number of screen refreshes that will come closest to your specified interval. All actual display times are written to a log file for each session.

Randomization & Trial Structure

- Randomize trials within a single block. (p 21)
- Randomize the order that blocks of trials are presented. (p 20)
- Randomly select which items from a set are presented.
- Extremely flexible trial design which can change from one trial to the next.
- Define stimuli directly in the input file or refer to stimulus lists. (p 23)
- Access stimuli from lists sequentially, randomly, or randomly without replacement. (p 23)
- Create scale responses and multiple choice questions in any format. (p 73)
- Skip pattern capability when a particular key is pressed on any trial. (p 39)
- Jump capability - Depending on which key is pressed you can have DirectRT jump to different events (e.g., such as a feedback sequence) and then resume where it left off. (p 41)
- Option of self-paced or time-limited responses. (35)
- Repeat trials for which stimuli were randomly selected at run-time. (p 84)

Data

- Writes data to an easy-to-analyze ASCII file.
- Add optional variables to the data file that identify within subject trial

conditions-makes for much easier analysis. (p 18)

- Writes all data into a single file that can be imported straight into Excel or SPSS.
- Includes a utility to easily merge data files collected on different computers. (p 14)
- Produces two data files per run-one that is simplified and easy to use for most purposes and a more detailed 'log' version that records everything including actual stimulus presentation times so you can ensure that your system is capable of achieving the timing you need. (p 55)

Other Features

- Easily modify and re-use your input files experiments.
- You don't have to learn a new editor! Do everything from your favourite spreadsheet application (e.g., Excel, Lotus, Quattro). If you know how to use one of these, then you already know how to edit DirectRT input files.
- Also works great in combination with MediaLab (versions 2000 and later). Just drop a DirectRT session into any MediaLab questionnaire. MediaLab will ask DirectRT to execute the session and will pass along the current subject and condition IDs. When DirectRT is finished, MediaLab will resume where it left off. This is a great way to assign multiple DirectRT input files as a function of between subject factors.
- Runs easily from a command line prompt so you can call the session and execute it seamlessly from your within own programs. You would never know you left your own program.
- Context sensitive help such that pressing F1 at any time will help you with what you are currently doing.
- Easy-to-use intuitive interface--no programming code necessary.
- Resolution independent-compatible with screen displays from 640x480 to 1024x768.

Setup

System Requirements

In order to execute input files, DirectRT requires that Microsoft's DirectX Version 7 or later be installed. If you don't already have it, there is a link to the download on our download page here: <http://www.empirisoft.com/download.htm>. Note that if you have Windows ME, 2000, or XP you will already have DirectX Version 7 or later installed. DirectX installation may be necessary only for Windows95 and Window98.

DirectRT should run very well on any high-end Pentium system (or equivalent), running Windows 95, 98, ME, 2000 or XP. DirectRT will not run on Windows NT because NT does not support DirectX beyond version 4 (7 or later is required). It is strongly recommended that systems running DirectRT have at least 64mb of memory, a graphics card with at least 4mb of video memory and a processor speed of at least 300mhz. The log files that DirectRT produces will show you whether or not the computer in question can handle your experimental design.

You can tell which version of DirectX is installed on your machine by typing 'dxdiag' from the run command on the Start menu. If DirectX is installed, doing this will tell you which version you are currently running. If you have version 7 or later installed, you can click on the 'Display' tab to ensure that there are no problems or warnings given with regard to the compatibility between DirectX and your display adaptor (video card). If there are no warnings, then all should be ok. If you'll be using sound with DirectRT, you can do the same with the 'Sound' tab.

Finally, it is also necessary to have a spreadsheet application installed (e.g., Excel) if you plan to edit or view the DirectRT input and output files on that machine-or if you plan to try the DirectRT on-line tutorials. All input and output files are created and saved in **.CSV** format which is editable from any standard spreadsheet application. See File Format for more details. You do NOT have to have a spreadsheet application installed simply to run the experiments and gather data. Note that users in a few countries may have to set their 'Regional Settings' in the Windows Control Panel to that of the United States in order to correctly view and edit .csv files in Excel.

Installation

To install DirectRT, first make sure you have DirectX version 7 or later installed (see above). Then double-click on the downloaded file 'drt2002.exe' or click 'Install DirectRT' from the CD-ROM. If possible, we always recommend installing from the downloadable file as it is the most likely version to be up-to-date (<http://www.empirisoft.com/download.htm>). The installation program will guide you through the set-up procedure. After the setup is complete, follow the security instructions below to gain authorization for each workstation to run DirectRT.

Security

To help control unauthorized distribution and use of DirectRT, it has been secured with ATX™ software protection. DirectRT will not run on any machine until this security has been disabled. To disable the security follow these steps:

- Install and run DirectRT.
- You will receive a message that the current workstation requires authorization to use the software, and a code number will be displayed.
- E-mail the code that appears to support@empirisoft.com. If you are an authorized user, you will receive a translation of this code which will be a registration number unique for the machine on which you have installed DirectRT (usually the same day). DirectRT should now be fully functional. If you are not yet an authorized user, you may still request a code which will allow you to use and evaluate DirectRT for up to 21 days. Important: ATX™ codes are unique to every machine. Be sure to get authorization for each machine that will be using the software. If you have any difficulty at all with this procedure, please contact us for assistance at support@empirisoft.com.

System Settings

The following are two suggestions for your general system set-up.

- *True colors.* To allow for the full range of colors usable by DirectRT, it is recommended that you use 16, 24, or 32 bit color. Many systems are set by default to 8 bit , which provides for only 256 colors (colors or color ranges may appear as dots, patterns, or solid regions). To set this option, right click anywhere on the desktop and then select properties. Choose settings, and then select the highest color range possible. 24 and 32 bit color ranges are both capable of displaying close to the full range of colors detectable by the human eye.
- *Show file extensions.* DirectRT frequently makes use of file extensions. These are the three-letters following the period on file names (e.g., .txt, .csv, .bmp). For this reason, it is advisable that you don't instruct Windows to hide these from you. To set this as a default, open any folder and select 'View' from the pull down menu. Select 'Options,' and then 'View' again. Deselect the option to have Windows 'Hide MS-DOS extensions...'

Introduction and Key Concepts

Timing is Everything

DirectRT is all about timing. Windows is generally a very chaotic environment to work in when it comes to controlling the timing of events. At least it has been until recently when Microsoft introduced DirectX, a software development platform that provides programmers with direct access to your computer's hardware. For our purposes, the most important element of DirectX is DirectDraw—a system allowing direct access to your computer's display hardware. What this means essentially is that the timing of what you see on the screen and when you see it can be controlled with great precision. Commonly used for the development of high-end 2 and 3D graphics applications and video games, DirectX provides a programming environment ideally suited for the rigorous demands of computerized psychological experimentation.

Screen Refresh Rate

Refresh Rate is one of the most fundamental concepts in high precision computerized experimentation. The screen refresh rate refers to how long it takes your computer to redraw what you see on the monitor. It is often expressed in Hz (e.g., 60Hz, 100Hz) which refers to the number of times your screen is refreshed (or redrawn) in a single second. This can easily be converted to milliseconds by dividing it into 1000, e.g., $60\text{Hz} = 1000/60 = 16.67\text{ms}$ per refresh; $100\text{Hz} = 1000/100 = 10\text{ms}$ per refresh. Why is this so critical?

Because everything DirectRT displays will be in *multiples* of the refresh rate and (hopefully) in perfect synch with the refresh rate. For example, if your refresh rate is 10ms, then DirectRT can display stimuli for 10ms, 20ms, 30ms, 40ms, 2000ms, or 2010ms, etc. There is simply no greater truth. So if you have a 15ms refresh rate and you want to show a stimulus for 12ms? Forget about it! 0, 15, 30, 45, 60... get the idea? Fortunately you don't have to worry much about this. You can ask DirectRT to display stimuli for any amount of time you like. Based on your refresh rate, DirectRT will display your stimulus for as many screen refreshes as it takes to come closest to your requested display time and will create a log data file for you so that you can compare requested against actual presentation times.

The Input File

The next great truth - to run a DirectRT session, you need an input file. This file will tell DirectRT what to present as well as where and when to present it and when to take reaction times. So how do you create an input file?

Use Your Spreadsheet Skills!

A key DirectRT concept is that you create an input file using whatever spreadsheet application you are most comfortable with. If you know how to work with a spreadsheet in Excel, Quattro, Lotus or other application then you already know how to edit DirectRT input files. When you're done creating or editing your input

file, you simply select Save As from the file menu of your spreadsheet application and save your input file in the .CSV format (comma-delimited).

The appeal of the .CSV format is that anybody can edit any DirectRT input file with any spreadsheet editor and then save it in this common format. This way no one has to learn how to use a new editing program for creating their input files. Everyone can use the spreadsheet application they already know best.

Structure of an Input File

When DirectRT reads your input file, it reads each row of your spreadsheet as a 'trial.' Because a spreadsheet can have as many rows as you like, you can put as many trials as you like in your input file. Just remember each row is a trial.

For each trial there are a few initial columns in which you specify general information about that trial, such as block and trial ids, what experimental condition(s) the trial belongs to, how it's supposed to be randomized and so forth. But then comes the really powerful and fun part about DirectRT... Screens!

Screens

For each trial, your input file tells DirectRT to create a series of 'screens'. A screen refers to what is supposed to be displayed on the monitor at a given time. This might be a stimulus word, some formatted instructions in a text file, a precisely located image, a full screen full colour image, one or more sound files playing, or any combination of these. Before a trial begins, DirectRT thinks briefly but intensely about the trial, creates the screens as you have specified, holds them all in memory and then flips through them - displaying each for the number of screen refreshes that best approximates the amount of time you've requested.

This sounds like it might be complicated, but it's actually quite easy once you've seen it in action. Creating a new screen for a trial is easy. You simply add 3 new columns to your input file: Stim, Loc, & Time. In the Stim (or Stimuli) column you enter what you want on the screen. In the Loc (or Location) column, you specify where you want it placed and whether you want it to be added to or replace the previous screen. DirectRT is very flexible in terms of creating screens-either from single or multiple stimuli sources. Finally, in the Time column, you specify how long the screen is to be displayed (in milliseconds) before DirectRT flips to the next screen in your trial. In the Time column you can also indicate that a response time is requested rather than automatically proceeding.

That's it. 3 columns-Stim, Loc, & Time. Remember this and you're set! You just keep adding a new set of three columns labeled with these same headers until your trial has been specified. Some trials might have only 1 or 2 sets, others may have 5, 10 or more. One appeal of this method is that it's very easy to create basic trials in just a few minutes. However, it will become apparent that with a few easy tricks you can also create very elaborate trials with a minimum of effort.

Styles

To save time, you can store font, colour, alignment and miscellaneous RT information in a Styles (.DRT) file. Combinations of such variables can be created and saved with a single style code. One of the initial columns of the input file is the Style column-this is where you tell DirectRT which of your styles you want to apply to the trial. You can create a custom style file and locate it in the same folder as your input file. Or, alternatively, DirectRT will simply access the styles located in your default styles file (located in the main program folder, 'styles.drt').

Running an Input File

By this point your work is done. You simply need to run the input file. There are three ways to do this. The easiest is to run DirectRT.exe-there should be a shortcut icon for it located on your desktop following the installation. From the file menu, select 'Select and run input file,' or hit Ctrl-S. Find and select your input file and then enter a subject and condition ID (these are for your own tracking purposes) and press OK. To stop a session at any point, press 'Esc.'

You can also run your input file from MediaLab (2000.4.1 or later). Simply select 'DirectRT Session' as a questionnaire item and enter the input file in the File field. MediaLab will pass the current subject and condition IDs to DirectRT, launch the session, and then resume where it left off when the session has completed. It is strongly recommended that you insert at least one MediaLab item (e.g., a basic instruction screen) between sequential DirectRT sessions. You can also run an input file from any program or batch file using a command prompt.

Viewing the Data

Just like input files, DirectRT uses the comma delimited or .CSV format for data files. This means you can view and edit the data again using your favourite spreadsheet application such as Excel. All data are written to a 'Data' folder stored in the same location as the input file. If this is not possible (e.g., the input file is on a CD-ROM), then it will be written to the 'Data' folder located in the main program folder. You can select 'Open Output File' from the main menu in DirectRT or select the file using Windows Explorer.

Note that there are 2 data files produced for each input file. One is a short version and the other is a longer more detailed log version. The shorter one will usually do for most purposes and is easier to deal with. However, when you want detailed information on each trial including the actual time each stimulus was presented for each trial etc., then the longer version can be found in the 'Log' subfolder.

Samples

A variety of sample have been included to illustrate the various capabilities of DirectRT. These input files are located in the 'Samples' folder of the main directory. Try running each of these samples and then look at the input file to see how it was done.

What's New in DirectRT v2002?

There are many enhancements that come with the 2002 version of DirectRT. For the most part all of these are optional. If you have become accustomed to using DirectRT v2000, you should be able to continue using 2002 without having to learn anything new. 2002 is fully backwards compatible so that DirectRT experiments from all previous versions should run the same as they always have without requiring modification.

With that said, here are the new features which you can learn and apply at your own pace. If you're new to DirectRT you can read this section or skip over it. All of this information is located in the appropriate places elsewhere in this manual. This section is intended for the more experienced user who just wants to know what's different from the last version. Have fun and let us know if you have any problems or if you have requests for additional features or fixes.

Additional Response Features

Request a short open ended response. You are now able to gather short fill-in-the-blank type responses. (p 74)

Record a voice response and reaction time via the microphone. Connect a microphone to your soundcard and you're able to collect voice activated response times. With a simple option, you can have the responses saved as sound files on your hard drive so you can listen to them later for verification. Record and save verbal responses from 100ms to 5 minutes in length. (p 88)

Record continuous joystick movement for a specified duration. Inspired by approach/avoidance research, you can track the continuous motion of a joystick while a visual or auditory stimulus is presented. DirectRT will track and record the x and y coordinates of the joystick at 10ms intervals. (p 77)

Wait for TTL signal from an external device (version 2002.3 and later). Connect external response hardware to your computer. If the external hardware can send a byte value to a port that DirectRT can read (e.g., the parallel port), then you can specify such these values as response options. (p 91)

Collect multiple response times per trial. No longer limited to recording a single response time per trial. Now request as many response times as you need. They will all be written to the extended 'log' data file. (p 83)

Record KeyRelease time in addition to subsequent KeyPress time. Instruct subjects to hold the spacebar (or other key) down when a stimulus is presented and then hit another when a stimulus is presented. Now you can record the response time to let go of the first key as well as the time to hit the second key. (p 80)

Require correct response (e.g., IAT). You can now require a correct response from subjects before they can continue. Just like they do in the IAT. You can specify the message (if any) that appears as incorrect keys are struck. (p 79)

Response feedback (display subject's response on screen). You can now display subject's response onscreen as visual feedback. This makes it clear to the subject that they have hit the intended key. (p 82)

Use mouse buttons for input. A simple alternative to the keyboard for dichotomous responding. (p 76)

Record discrete joystick responses. A simple alternative to continuous monitoring of the joystick, you use the basic directions of the joystick as discrete responses (e.g., up, down, up-right, etc.). 8, 45-degree regions have their own codes which can be used just like keycodes. (p 76)

Additional Stimulus Features

Play sounds files. DirectRT allows you to play sound files and to synchronize them with other events and stimuli in your session. You can control three aspects of the how the sound files are played: Frequency, Pan (left vs. right) and Volume. (p 86)

Use subjects own responses as subsequent stimuli. You can use subjects' responses as stimuli on later trials. This is especially useful when you want to personalize stimuli that are used in a task. (p 74)

Clear the screen during a response time interval. While waiting for a response from the subject, you can now clear a stimulus from the screen after a fixed interval-- even if the subject has not yet responded. (p 81)

Use randomly selected stimuli again on later trials. Use a stimulus from a previous trial without knowing exactly what that stimulus will be when you design your input file. This could happen for example, when a stimulus is randomly determined on one trial but then you want to repeat that trial. (p 84)

Send TTL signals to external equipment for physiological synchronization (e.g., SCAN). In order to synchronize events in DirectRT with external equipment (e.g., physiological equipment, other computers), you can send TTL (Transistor-Transistor Logic) signals through any of your computer's accessible ports (e.g., parallel or serial ports). (p 91)

15 new samples (p 74), including a complete image based Implicit Association Test (IAT, p 93) and an Advanced graphics sample (p 94) that uses just about every graphics trick that DirectRT has to offer.

Main Menu Commands

The main menu commands are found at the top of the DirectRT program window. These are the general commands used for executing input files, opening input files for editing, viewing data files, and other general purposes. The descriptions contain links you can click to learn more about the topic.

Files

Open Input File - Select and open a input file for editing. DirectRT will open the input file with your system's default application for editing .CSV files (e.g., Excel, Lotus, Quattro). To learn about input files, see [Creating Input Files](#).

Open Output File - Open a data file for viewing. As with input files, DirectRT will open the data file with your system's default application for editing .CSV files. To learn about DirectRT data files, see [Overview of Data Output](#).

Launch Explorer - Launches the Windows Explorer for file exploration and handling.

Select and Run Input File - If you haven't executed the input file before, this choice allows you to search through the folders on your machine to find the desired input file. Locate the input file and then either double click on it, or select it and click 'open.' To learn about executing an input file, see [Running an Input File](#).

Run Last Input File Again - This choice will execute the input file most recently run on the machine. The key sequence 'Ctrl-R' will perform the same function. This command is handy when multiple participants are being run back-to-back using the same input file, or when you're developing and testing an input file.

Run a Recent Input File - DirectRT keeps track of the ten most recently run input files and displays them in the Files menu. Clicking on any one of them will re-run the selected input file (assuming it's where you left it).

Clear Recent File List - Clears the list of recent files. Clears the list only-doesn't have any impact on the files themselves.

Exit DirectRT - Ends the program

Edit

Default Styles - Edit the default styles file located in the main program folder (styles.drt). To learn about styles, see [About Styles and the Style Editor](#).

Recent Style Files - Choose from a list of recently edited style files

Clear Recent File List - Clears the list of recent files. Clears the list only-doesn't have any impact on the files themselves.

Tools

Test Refresh Rate - Determine the refresh rate of your display. To learn about the significance of the refresh rate, see [How DirectRT Works](#).

Show System Info - Display hardware and software configurations of the current system. If you are experiencing display problems with DirectRT, this is a good place to start. Click on the Display tab and check the notes section. Are any problems noted here? You can capture the active screen as a graphic to send to tech support by pressing ALT->PRINT SCREEN. Using any image editor, create a new picture and select PASTE. You can then save the image and send it to us.

Merge Data Files - Utility program provided to help you merge data files collected from multiple subjects. After starting the data merge utility, you can find a quick tutorial for it from its Help menu. For direct access, the file merge utility is located here: ...\\DirectRT\\FileMerge\\filemerge.exe

Voice Response Setup - Utility program provided to setup and test your microphone for input with voice activated response times. For direct access, the SoundCheck utility is located here: ...\\DirectRT\\SoundCheck\\SoundCheck.exe

TTL I/O Test - Utility program provided to test the input/output routines used by DirectRT to communicate with the external world. This utility will let you send TTL signals to any port you like and will let you continuously monitor the value at any port. Since the code is identical to that used by DirectRT it's an easy way to make sure all is well with TTL operations. For direct access, the IOTest utility is located here: ...\\DirectRT\\TTL\\IOTest.exe

Show Status Bar - Sets the option to display or hide the status bar that appears at the bottom of the DirectRT window. In rare cases, this status bar may appear during sessions. You can prevent it by hiding the status bar with this option.

Help

Help Topics - Opens this help file at the beginning

DirectRT Samples - Takes you directly to the Samples section of the Help file where the various sample files are described.

About - Displays the current version of DirectRT installed on your system.

License - Provides a way to enter a new license code. Useful if you are using a trial code and want to replace it with a purchased code before the trial expires.

Key Codes - Takes you directly to the key codes page in the help file.

Creating Input Files

Covered in this topic:

Block	Trial	Optional Fields (!)
RBG	RWG	Style
Stim	Loc	Time
Skipping	Jumping	File Format

It is recommended to first read the DirectRT Introduction and Key Concepts before reading this section. It explains the fundamental concepts associated with Input Files. It is also highly recommended that you check out at least a few of the samples before beginning here. Note that you can go to the samples at any time by clicking on **Samples** in the left hand window, by clicking the **Samples** button on the top of the help window, or by clicking **Samples** on the Help menu of the main program window.

Note that when you click on any sample, a window for that sample will appear. In the window will be a short description of what the input file does as well as the option to **view the input file**, the actual session is produces (you'll see it in action) and the resulting data file.

This is a handy page to come to when you want to know details about a particular column or general features of an input file.

File Format

QuickInfo

All input and output files used in DirectRT are saved in the .CSV format. This standard comma-delimited format allows editing of input and output files using any favourite spreadsheet application.

** Excel and some other programs may tell you that saving in the .CSV format will cause a loss of formatting and/or other info. This is fine and in no way will harm your input file.

** Users in a few countries may have to set their 'Regional Settings' in the Windows Control Panel to that of the United States in order to correctly view and edit .csv files in Excel.

CSV File Format

A key DirectRT concept is that you create an input file using whatever spreadsheet application you are most comfortable with. If you know how to work with a

spreadsheet in Excel, Quattro, Lotus or other application then you already know how to edit DirectRT input files. When you're done creating or editing your input file, you simply select Save As from the file menu of your spreadsheet application and save your input file in the .CSV format (comma-delimited).

Hints:

When you ask DirectRT to open an Input or Output file it launches the application set by default to open .CSV files on your computer. If there is no default set, you may have to indicate which application you want to use when you first try this. Simply choose whichever spreadsheet program you like to use. If the default program is set but you want to change it then do this:

Go to Windows Explorer and find a file of the .CSV type. Click it once to highlight it. While holding down the SHIFT key (Win95/98 only) click the right mouse button on the file and select 'Open With.' Find and select your desired spreadsheet application. DirectRT will now send your files to that application for viewing and editing.

Block

QuickInfo

A label to help you keep track of which trials are which. DirectRT for the most part does not use this number but does write the assigned value to the data file for your reference.

Value

Any Integer. A '0' specifies the trial data is not to be written.

Details

If you are repeating sets of trials within a single input file, you may wish to label each trial with the set to which it belongs. You may use any integer value for any trial. Using the value of '0' will indicate that the data obtained for that trial is not to be written (e.g., in the case of instruction or debriefing screens).

Hints

You may be including a series of instruction screen or 'rest' trials in your session. If you do not want these trials to clutter up your data files then assign them a Block value of 0. Data will not be recorded for any trial with a block value of 0.

Trial

QuickInfo

A label to help you keep of which trials are which. As with block labels, DirectRT largely doesn't care what values you use. The values are primarily for your references and will be written as labels to the data file with each trial. DirectRT does use the trial labels though when performing skip and jump behaviours.

Value

Any Integer.

Details

DirectRT presents trials as they appear sequentially in the input file. Consequently, trial labels do not have an impact on the order in which trials are presented. Trial numbers become important only when you have skip or jump patterns in your input file.

Optional Fields

QuickInfo

Provides an easy way to write unique trial information to each record in the data file based on within-subject trial factors. Can make data analysis easier by identifying important trial characteristics a priori (as opposed to figuring it all out afterwards). DirectRT does not use these columns in any way except to write as labels to your data file.

Details

As the name implies, optional fields are, umm.. optional. Immediately following the Block and Trial columns you can insert any number of columns with header values starting with a !

Values from these columns will be copied with the other trial info to the data file.

** Note that you do not need to have ANY optional fields in your input file.

Example

Imagine you have a simple 2x2 design. On each trial you present either a positive or negative image (prime) followed by either a positive or negative word (target).

Without Optional Fields (default)

The first few columns of your input would normally look like this:

	A	B	C	D	E
1	block	trial	rbg	rwg	style
2	1	1	1	1	
3	1	2	1	1	
4	1	3	1	1	
5	1	4	1	1	

This would produce a data file that looked like this:

	A	B	C	D	E	F	G	H	I
1	Subj	Cond	Block	Trial	Order	Resp	Name	RT	Correct
2	23	2	1	1	3	44 Z		566	TRUE
3	23	2	1	2	4	53 /		444	TRUE
4	23	2	1	3	1	44 Z		466	TRUE
5	23	2	1	4	2	44 Z		333	FALSE

With Optional Fields

However, you might want to create two optional fields to represent the two conditions. The first might be !Prime, and the second !Target, like this:

	A	B	C	D	E	F	G
1	block	trial	lprime	ltarget	rbg	rwg	style
2	1	1	neg	pos	1	1	
3	1	2	neg	neg	1	1	
4	1	3	pos	pos	1	1	
5	1	4	pos	neg	1	1	

This would result in a data file like this:

	A	B	C	D	E	F	G	H	I	J	K
1	Subj	Cond	Block	Trial	Prime	Target	Order	Resp	Name	RT	Correct
2	23	2	1	1	neg	pos	3	44 Z		566	TRUE
3	23	2	1	2	neg	neg	4	53 /		444	TRUE
4	23	2	1	3	pos	pos	1	44 Z		466	TRUE
5	23	2	1	4	pos	neg	2	44 Z		333	FALSE

Won't that be easier to analyze?

Combining Optional Fields

Or alternately, you might combine them into a single Optional Field such as !Condition like this:

	A	B	C	D	E	F
1	block	trial	lcondition	rbg	rwg	style
2	1	1	negpos	1	1	
3	1	2	negneg	1	1	
4	1	3	pospos	1	1	
5	1	4	posneg	1	1	

This would result in a data file like this:

	A	B	C	D	E	F	G	H	I	J
1	Subj	Cond	Block	Trial	Condition	Order	Resp	Name	RT	Correct
2	23	2	1	1	negpos	3	44	Z	566	TRUE
3	23	2	1	2	neneg	4	53	/	444	TRUE
4	23	2	1	3	pospos	1	44	Z	466	TRUE
5	23	2	1	4	negneg	2	44	Z	333	FALSE

RBG - Randomization Between Groups

QuickInfo

All trials sharing the same RBG value define a group to be kept together. If multiple groups are defined, then DirectRT will present the groups in a random order. Identical to the MediaLab function of the same name.

Value

Any positive integer value

Details

You may wish to randomly order and present entire groups of trials in your questionnaire (e.g., randomizing which of four sets of 20 trials is presented 1st, 2nd, 3rd and 4th). To do this, assign the same RBG value to all of the items you want to keep together. For example, if you have five sets of ten trials and you want to randomize the order in which these 5 groups of trials are presented, then assign each group a different number and make sure all the items within each group share the same number for that group. The items within each group will all stay together, but the groups themselves will be randomly ordered. Items assigned a '0' or 'none' will not move.

****Note that all items within a group must be sequential in the questionnaire.**

Note for ease of analysis, trials are written to the data file in the order they occur in the input file, not in the randomized order in which they are presented. However an 'Order' variable is also written to indicate the sequential order in which the trials actually occurred.

Example

See Samples, Randomizing

See Also

RWG - Randomizing Within Groups

RWG - Randomization Within Groups

QuickInfo

DirectRT will randomly order and present all items sharing the same RWG value. Identical to the MediaLab function of the same name.

Value

Any positive integer value

Details

You may wish to randomly order and present trials within your questionnaire. To do this, assign the same RWG value to all the trials you want to have randomized with each other. For example, if you have five trials you want randomized, assign them each a '1'. The five trials will be randomly presented in the five positions they occupy in the input file. All other trials will be presented in their normal positions as long as their RWG value is 0. You can independently randomize another group of items by assigning them all a '2', another group with '3's and so forth.

****Note that unlike the RBG method, items do not need to be sequential to be assigned the same RWG value.**

Note for ease of analysis, trials are written to the data file in the order they occur in the input file, not in the randomized order in which they are presented. However an 'Order' variable is also written to indicate the sequential order in which the trials actually occurred.

Example

See Samples, Randomizing

See Also

RBG - Randomizing Between Groups

Style

QuickInfo

The style code is a number that identifies a particular style (e.g., font, colours, rt parameters) for the trial. Styles are stored in a .DRT style file. Edit style files by double clicking on them, or open the style editor from the DirectRT main program window (select Edit->Style File)

Details

To save time, you can store font, colour, alignment and miscellaneous RT information in a Styles (.DRT) file. Combinations of such variables can be created and saved with a single style code. In the style column of each trial, specify the style you want to use for the trial by using the numeric style code from the style file you're using.

Hints

You can (and probably should) have different style files for different experiments. You can copy, modify and rename any style file and locate it in the same folder as your input file-making it unique to the experiment you are currently working on. DirectRT will first look for a style file located in your experiment folder. If it doesn't find one, it will use the default 'styles.drt' file in the main program folder. Note thought that using the default style file can lead to unpredictable results if you change the default styles for another experiment and then try the current experiment again. For this reason it's usually best to copy a style file to your experiment folder and modify it as needed for that particular experiment. This way, the styles will be unaffected by changes in the default settings. For example, you'll see that every sample folder has its own unique style file.

Another useful hint is that you can change styles in the middle of a trial. To change a style part way through a trial, put the style in the Stim column surrounded by < > like this:

	E	F	G	H	I	J	K
	style	stim	loc	time	stim	loc	time
1	1	stim1	0,0,1	500	stim1b <5>	0,0,1	500
1	1	stim2	0,0,1	500	stim2b <5>	0,0,1	500
1	1	stim3	0,0,1	500	stim3b <5>	0,0,1	500
1	1	stim4	0,0,1	500	stim4b <5>	0,0,1	500

In the above example, each of these trials starts out using Style 1 but switches to Style 5 for the second stimulus.

Note you can also switch style settings in the same way within any text file you present (i.e., by placing style codes within the text). See Text Files for more details.

Stim

QuickInfo

The Stim column identifies a stimulus to be presented during the trial. For stimuli that are files (e.g., image, sound and text files), DirectRT expects the actual files to be in a 'Stim' folder located in the same folder as the input file. If the files are located elsewhere, then the full path needs to be specified. Every Stim column needs a corresponding Loc and Time column.

The following are the different stimulus types allowed in the Stim column. To tell DirectRT which type it is, you need to precede it with an identifying character as follows:

Identifier	Stim Type	File Type	Function
none	Image	*.bmp	Displays an image file
!	Sound	*.wav	Plays a sound file
~	Text		Displays specified text
#	Text File	*.txt	Displays the contents of a text file
&	StimList-Sequential	*.txt	Selects next or specific stimulus from a list
%	StimList-TrueRandom	*.txt	Randomly select stimulus from a list
\$	StimList-Random	*.txt	Randomly selects stimulus from a list without replacement
=	Prior Response		Presents a prior response of the subject as a stimulus
?	Prior Stimulus		Presents a prior stimulus from a specified trial

Image file

To present an image (.BMP) file, simply enter the name of the image file. Images are the default stimulus type so the filenames do not need to be preceded with a

special character. Also you do not need to type the '.bmp' following the filename as it is assumed.

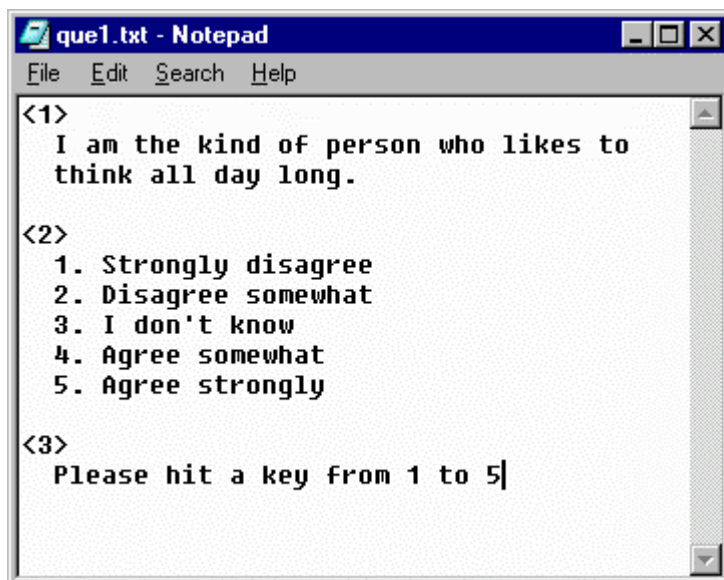
~ Text

To present a single string of text simply type the actual text you want to display preceded by a '~' Any single word or single line of text can easily be presented in this way. The text will be presented using the colours and font settings associated with the style code for that trial.

If you want to present more than a single line of text on the same screen, you can use compound screens (see samples), more simply, you can request a text file instead.

Text file

To present more than a single line of text on the same screen you can enter the name of a text file to be used as a stimulus. To display the contents of a text file, type the name of the text file preceded by a '#' You do not need to type the '.txt' (it is assumed) although it will not affect anything if you do. Also, you can embed style codes within a single text file for multiple font and colour formats. In the example below, different styles are used for the question wording (style 1), responses (style 2), and page footer (style 3). You can also place style codes in the middle of a line of text (e.g., to change the color or italics of a font). If you do use multiple style within a text file, DirectRT will use the first background colour it encounters (i.e., additional styles will not change the initial screen colour). Text files will always be located starting from the top left corner of the screen. Left and top margins are achieved with spaces and hard returns, respectively.



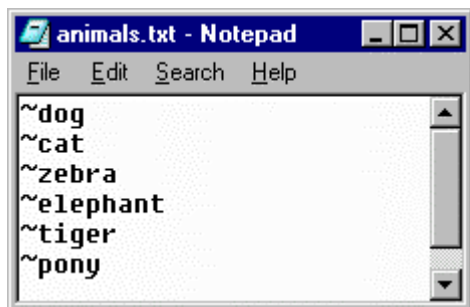
! Sound File

To present a sound (.WAV) file, simply enter the name of the sound file preceded by a '!' (e.g., '!mysound'). You do not need to type the '.wav' following the filename as it is assumed. Note that the sound file should be located in the experiment's Stim folder. When using a sound file for a stimulus, you should enter a '0' in the Time column if you want DirectRT to play the sound and proceed immediately to the next stimulus. If you want a delay before the next stimulus occurs, then you should enter the length of that delay in the Time column in milliseconds. To use the default settings for sound files, enter '0,0,0' in the Location column.

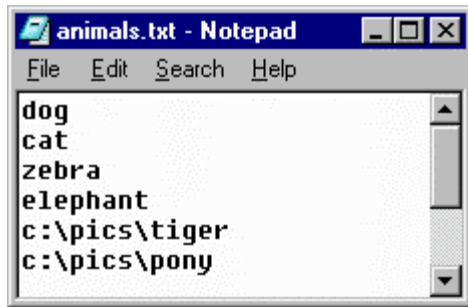
Advanced Options: When playing sound files, you can set options by using values other than '0,0,0' in the Location column. These three options are Frequency, Pan and Volume, respectively. To speed up or slow down the sound file, you can set the frequency in hertz (Hz) at which the sound is played by entering a value from 100 to 100000. To control whether the sound is dominant in the right or left channel, you can set a Pan value anywhere from -100 (left only) to 100 (right only). Finally, to control the volume at which the sound file is played, you can enter a value from 0 (normal) to -100 (completely muted). Sounds can only be reduced in volume and not amplified. Example: To play a wave file at 1500 Hz, in the left ear only, at half volume, you would enter '1500,-100,-50' in the Location column for the sound file stimulus.

StimLists

Rather than specifying a specific stimulus, you can opt to request that the stimulus be *chosen* from a stimulus list. StimLists are text (.txt) files that you create with any text editor that simply list stimuli. This can be especially handy if you want to randomize the stimuli that fill particular positions in your input file (e.g., as opposed to or in addition to randomizing the order of trials-see Randomizing Trials). The list can include any stimuli you could identify normally (e.g., text, images, text files). *Just specify all the stimuli the same way you would as if you were entering each directly in the Stim column.* For example, the following depicts a simple StimList with a series of Text stimuli:



In the next example, this StimList contains a series of images located in the Stim folder and then two images located in another folder.



Selecting Stim from a StimList

The symbols **\$**, **%**, and **&** each indicate a StimList. The difference between them is in how DirectRT selects the stimulus from those listed in the file. The three types of stimulus selection are: Sequential, TrueRandom, and Random.

& Sequential

The **&** symbol indicates that you want DirectRT to display the first stimulus in the list and then each subsequent stimulus whenever it comes across this StimList again. For example, if you were using the list of animals above, every time you entered **&animals** in the Stim column of your input file, the next animal in the list would be displayed. If all the stimuli are presented, then DirectRT will begin again from the start of the list.

With the sequential selection type, you also have the option of calling a specific stimulus from the list. To do this, follow the name of the StimList with a **^** and the ordinal position of the stimulus in the list. For example, in the list of animals above, to present 'zebra' you would enter **&animals^3**, or for tiger you would enter **&animals^5**. Note that doing so sets the sequential position of the file to that chosen item. If you do not specify a specific item the next time you call the file, it will continue from that point.

% TrueRandom

The **%** symbol indicates that you want DirectRT to display a randomly chosen stimulus from the list. *TrueRandom* refers to the fact that each time DirectRT chooses a random stimulus it is unconditional. This means that the next stimulus chosen from the list could also have been the last.

\$ Random

Finally, the **\$** symbol indicates that you want DirectRT to randomly choose a stimulus from the list and then take that item out of the running for future selection unless all of the items are first used up. This is how you get randomization without duplicating what is presented or, as it's often called, *randomization without replacement*. Only if all of the item on the list have been displayed, then DirectRT will re-randomize the list and start again.

Note if for some reason you want to re-set the randomization *before* all the stimuli have been displayed, the easiest way to do this is to call a different StimList with identical stimuli.

Hints

When the stimulus is a file and the file itself is located in a 'Stim' folder in the same location as the input file then you can simply enter the name of the file in the Stim column. If the file is located in a subfolder within the Stim folder, then you can enter the name of the subfolder followed by the name of the file (e.g., images\myimage). Otherwise, you need to specify the full path and name of the file you want to present (e.g., c:\mypictures\myimage).

The advantage of placing the stimulus files in the Stim folder (or a subfolder) is that the folder containing the input file(s) and Stim folder can then be moved to a different place and you won't have to worry about checking path names (e.g., c:\..., d:\..., etc.)

Advanced Stimulus Features

? Using Randomly Selected Stimuli Again

In special cases you might want to instruct DirectRT to use a stimulus from a previous trial without knowing exactly what that stimulus will be when you design your input file. This could happen for example, when a stimulus is randomly determined on one trial but then you want to repeat that trial. On the later trial you can simply tell DirectRT which stimulus you want to repeat. Essentially saying, 'use the third stimulus from block 2, trial 5--whatever it was!' To do so, you can use the '?' operator which means to use a previous stimulus, no matter whether it was randomly chosen or not. To refer to a previous stimulus, you just need to know the block (b), trial (t) and stimulus (s) ID. For example if you want to use block 1, trial 3, stimulus 4 in a later trial, you would request: '?b1t3s4' in the stim column.

See sample: 'using previous stim' in the 'other features' sample folder

= Using a Subject's Response as a Stimulus

Another special case can occur when you want to use the subject's own response as a stimulus. This could happen for example, if the subject were asked to type in a word, and then you wanted to use that word as a stimulus. To present a response as a stimulus, simply use '=' followed by the trial ID in the Stim column. When DirectRT gets to that trial it will substitute the prior response as the new stimulus. You can use the standard operators (e.g., ~, #, \$, %, etc.) to indicate what type of stimulus it is, e.g., '~ =4' means present the response from trial 4 as text.

For example, on trial 4 the subject enters their name, 'George'. Later on trial 10, you want to use that response as a stimulus. In the Stim column for trial 10, you

could enter '~=4'. This would tell DirectRT to present the response from trial 4 as a text stimulus on trial 10.

See sample: 'open ended responses as stimuli' in the 'other features' sample folder

If you want to get really fancy with this, you can name image, sound, and stimlist files with the possible responses to a question and then have them presented using this feature. For example, if the possible responses to trial 4, are 'a', 'b' and 'c', you could create stimlists called 'a.txt', 'b.txt' and 'c.txt'. Then later if you specified a stimulus as '\$=4', DirectRT would substitute the response here from trial 4. So if the subject had responded with 'b' to question 4, using a later stim value of '\$=4' would mean '\$b', in other words, randomly select a stimulus from stimlist 'b.txt'. The same logic applies to all file types that you can specify as a stimulus.

ttl Sending Synchronized TTL Signals to External Equipment

In order to communicate with external equipment (e.g., physiological equipment, response boxes, other computers), you can send and receive TTL (Transistor-Transistor Logic) signals through any of your computer's accessible ports (e.g., parallel or serial ports). In English, this means you can ask DirectRT to send a value from 0-255 to your external equipment any time you like, and/or you can have DirectRT wait for any value from 0-255 to come from the external equipment. To accomplish sending a value, you list a TTL signal just like any other stimulus in the Stim column of your input file. See the example below. We indicate that we want a TTL signal to be sent by entering 'ttl:n' in the Stim column where *n* is the byte value from 0-255 that we want to send. We enter the port ID in the Loc column (e.g., 888 is usually the LPT1 parallel port). Finally, we enter a 0 in the Time column if we want to send the value and forget about it. Alternatively, we could specify the number of milliseconds here that we want the signal to be sent before setting the port back to its original state.

So what happens in this example? On each of these trials, DirectRT will send the value (e.g., 2, 3, 4 or 5) to port 888 and then immediately proceed with the next stimulus. Any external equipment that can read the byte value located at port 888 will now be able to know precisely when DirectRT displayed each of the subsequent stimuli.

stim	loc	time	stim	loc	time
ttl:2	888	0	~dog	0,0,1	rt:53,44
ttl:3	888	0	~slow	0,0,1	rt:53,44
ttl:4	888	0	~slow	0,0,1	rt:53,44
ttl:5	888	0	~dog	0,0,1	rt:53,44

For details on receiving TTL signals for response times, see Time.

Notes About Image Files: BMP vs. JPG & GIF

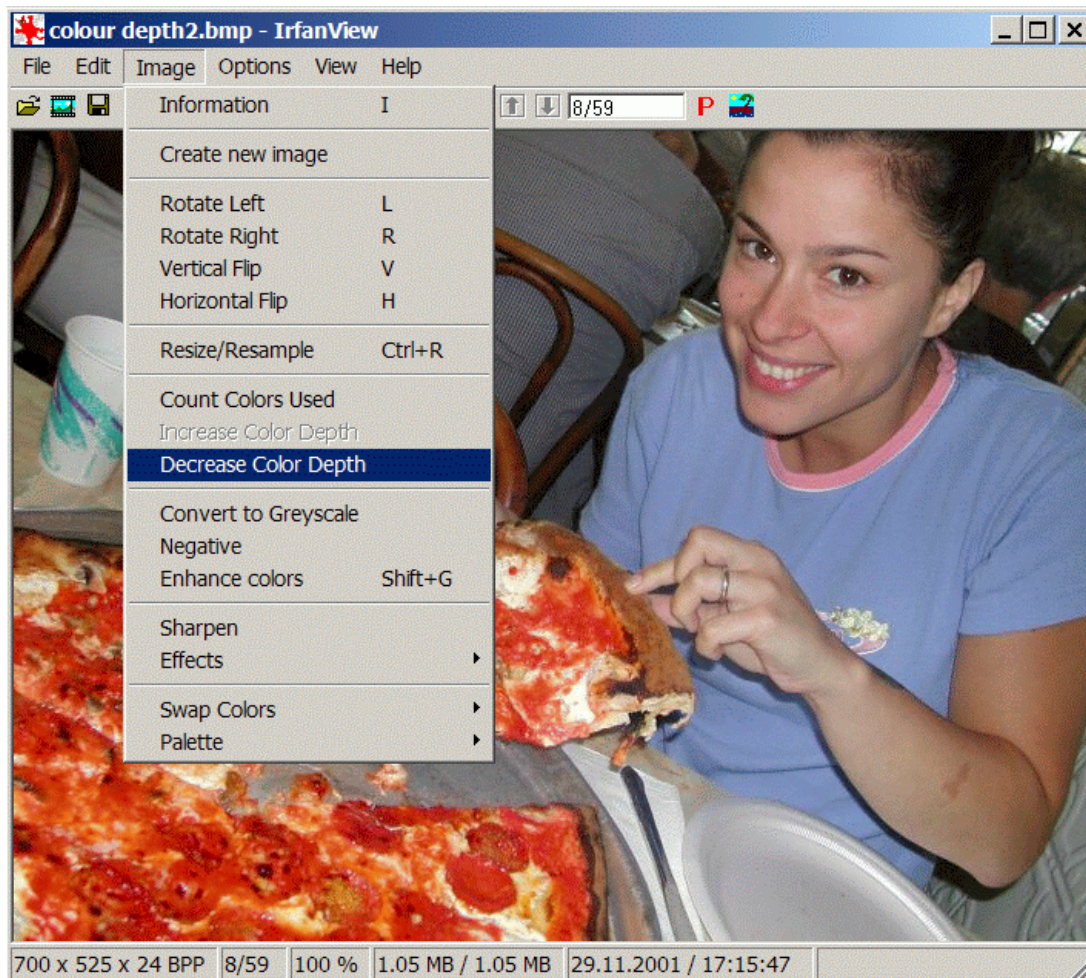
DirectRT uses bitmap (.BMP) images. The reason is that bitmap images are not compressed the way jpeg or gif files are. What's so good about that?! Yes it means bigger files taking up more hard drive space. Your pain is felt. The benefit is SPEED. Because bitmap files are not compressed, they do not need to be DE-compressed prior to their display. This hastens things up considerably.

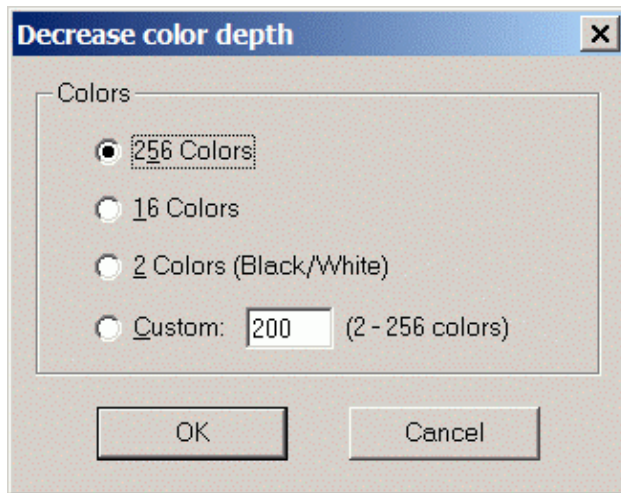
8-Bit, 256 Colours vs. 24-Bit, Millions of Colours

You can further ease the processing load on your computers by converting your images to 8-bit (256 colour) depth. *This will cut your file size down by two thirds!* This can be done with little loss of image quality. If you don't have a utility for doing this, there is a freeware program called IrfanView that does a great job. Find it at <http://www.irfanview.com/>.

With Irfanview, to convert from 24bit to 8bit colour (see Figure):

- Open the image, from the Image Menu, select Decrease Colour Depth
- Select 256 colours and save your image. Check the difference in file size!





Creating Graphics with PowerPoint

The 'Priming with Images' sample illustrates how you can use PowerPoint to create graphics to use in your DirectRT session. To do this follow these steps:

1. Create your PowerPoint show.
2. Select SaveAs and ask for PowerPoint to output the screens as .bmp files.

OR, if your computer smoothes its screen fonts, then you can create even nicer images from your PowerPoint show by doing the following

3. Instead of saving your show as .bmp files, go into your system's display settings and change your display to match the screen resolution in which you'll be running your DirectRT session (e.g., 640x480, 800x600, 1024x768).
4. Run your PowerPoint show at this resolution.
5. For each screen, when it appears, press **PrintScreen** on your keyboard. Press Esc to stop the PowerPoint show and minimize PowerPoint.
6. Open your graphics application and select **Paste**. The screen image will be pasted into your graphics application as a graphic.
7. Save the image as a .bmp file. Remember that DirectRT can process the image much faster if you save it in 8-bit or 256 colours (see more info on Colours).

From Other Windows

You can capture the contents of any open window as a graphic by pressing **Alt-PrintScreen**. By pressing Alt when you hit PrintScreen, you limit the graphic to the currently active window. Hitting PrintScreen alone capture the entire screen. Then , just open your graphics application and **Paste** the image and edit it as necessary.

Creating scales

Creating scale images with ticks and equal distances between the ticks can be tricky in standard graphics applications. The scale image used in the Assessing Modern Racism sample was done as follows:

1. Start Excel or another spreadsheet application
2. Select all the cells and choose a color fill (e.g., black)
3. Select all the cells and select a fairly thick color border (e.g., yellow)
4. Select a common column width according to how far apart your scale points should be graphically.
5. Use the Alt-PrintScreen method described above to capture the window as a graphic.
6. Paste the image into any graphics editor and trim the excess away leaving your scale and tick marks.
7. Add text labels as desired and save.

Loc - Stimulus Location

QuickInfo

Specifies where the stimulus is to be located on the screen and whether to add the stimulus to the previous screen or start a fresh screen. Consists of three numbers separated by commas: Horizontal, Vertical, Clear

Common Values:

0,0,1 = center the stimulus, clear the previous screen

0,0,0 = center the stimulus, add it to the previous screen

Remember values of 0 for Horizontal and Vertical mean center-screen.

Details

The purpose of the Horizontal and Vertical values is to define a single point on the screen. DirectRT uses this point combined with the Align property of the current style to figure out where to place the stimulus. If Align is set to Center, then the point defines the Center of the stimulus. If Align is set to Left, then the point defines the top, left corner of the stimulus. Finally, if Align is set to Right, then the point defines the top right corner of the stimulus.

Horizontal

The first of the three numbers is the number of pixels from the left of the screen. Optionally, a value of 0 indicates the horizontal center of the screen. Values can also be entered as a percentage, e.g., .3 would place the point 30% of the way across the screen.

Note: Where the stimulus is actually placed depends on the current Style setting for that trial. Every style has an alignment property of Left, Center, or Right. If the alignment setting of the current style is left, then the left point specified

Vertical

The second number is the number of pixels from the top of the screen. Optionally, a value of 0 indicates the vertical center of the screen. Values can also be entered as a percentage, e.g., .3 would place the point 30% of the way down the screen.

Clear

The third number indicates whether to add the stimulus to the previous screen (0) or to clear the previous screen and place the stimulus on a fresh screen (1).

Example

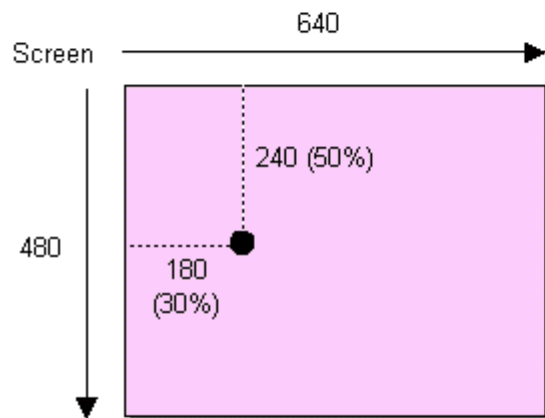
Using Pixels:

H	I	J
stim	loc	time
baby	180,240,1	500

or Percentages:

H	I	J
stim	loc	time
baby	.3,.5,1	500

A Point is Defined:



+



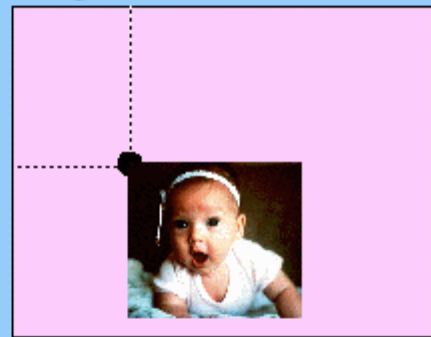
=

* Align is a setting of the current Style

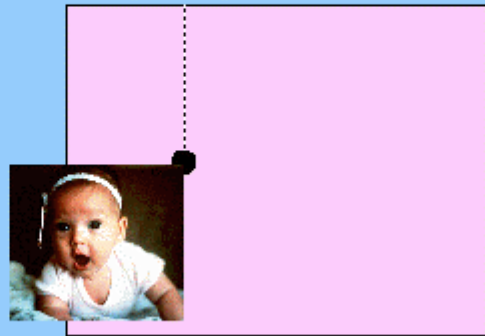
If Align = Center



If Align = Left



If Align = Right



About Pixels

Although DirectRT will work in a number of screen resolutions, how things look will depend on the resolution you choose. Certain items may need to be repositioned with the Left and Top parameters when running in high resolution modes. What do the Left and Top numbers mean? If you are running your session in 640x480 mode then the width of your screen is 640 pixels and the height is 480 pixels. This means that a 640x480 image will fill the screen in this mode. In 800x600 mode, the screen is 800 pixels wide and 600 high and will not be filled with a 640x480 image. You can change the screen resolution your session runs in by selecting Options in the Style Editor.

Special Values for Sound Files

For the special case where you are using a sound file as a stimulus, the location values take on different meanings. To use the default values, simply enter '0,0,0' in the Location column. However, you can set options by using values other than '0,0,0'--these three options are Frequency, Pan and Volume, respectively. To speed up or slow down the sound file, you can set the frequency in hertz (Hz) at which the sound is played by entering a value from 100 to 100000. To control whether the sound is dominant in the right or left channel, you can set a Pan value anywhere from -100 (left only) to 100 (right only). Finally, to control the volume at which the sound file is played, you can enter a value from 0 (normal) to -100 (completely muted). Sounds can only be reduced in volume and not amplified. Example: To play a wave file at 1500 Hz, in the left ear only, at half volume, you would enter '1500,-100,-50' in the Location column for the sound file stimulus.

Special Value for TTL Signals

For the special case where you are sending a TTL signal as a stimulus, the value of the Loc column should be the port address (in Decimal) to which you want the TTL signal sent (e.g., 888 is commonly used to identify the LPT1 parallel port). To help identify ports and port IDs, you can run the IOTest utility from the DirectRT Tools menu (...\\DirectRT\\TTL\\IOTest.exe). Note that users on Windows 2000/XP machines may have to be logged on with administrator rights to work with TTL functions.

Time

QuickInfo

Specifies how long the stimulus is to remain on the screen in milliseconds, OR tells DirectRT to wait for a specified input event (e.g., keypress, mouseclick, joystick event, microphone input, TTL signal).

The Time column can be used to specify 1) a stimulus duration or 2) that a response is required.

Specifying a Stimulus Duration

Enter any integer value to specify the number of milliseconds for which the stimulus will be displayed. E.g., in the following example, the stimulus 'stim1' would be displayed for 500 milliseconds before the next stimulus were displayed.

F	G	H
stim	loc	time
stim1	0,0,1	500

Advanced Hint: You can create a 'Compound-Screen' by using a Time value of 0 for the current stimulus and a Clear (see Loc) value of 0 for the *next* stimulus. Using a time value of 0 tells DirectRT that you don't want to display the stimulus alone. DirectRT will then take the next stimulus and put it on the SAME screen for display at the same time (i.e., with no delay in between). Just make sure you don't request a clear screen for the next stimulus! In fact, you can place as many stimuli on a single screen at one time as you like using this method.

Note about sound files: Sound files are always played to their completion. For sound files, the stimulus presentation time represents the number of milliseconds that DirectRT should wait after starting the sound before presenting the next stimulus. For example, if you wanted to play three sound files, each starting one second after the other, then you would use a time value of '1000' for each. To have them all play simultaneously, you would use a value of '0' for each. Note: When playing multiple sound files 'simultaneously', DirectRT will start each sound sequentially with as little delay as possible given your system's processing capabilities.

* * * *

Specifying that a Response is Required

In addition to specifying the duration of a stimulus, you can use the Time column to indicate that a response is expected before continuing. Responses include:

- KeyPresses, MouseClicks, and Discrete JoyStick Motions
- Request a short open ended response
- Record a voice response and reaction time via the microphone
- Record continuous joystick movement for a specified duration
- Wait for TTL signal from an external device (version 2002.3 and later)
- Collect multiple response times per trial
- Record KeyRelease time in addition to subsequent KeyPress time
- Require Correct Response (e.g., IAT)
- Response Feedback (display subject's response on screen)

KeyPresses, MouseClicks, and Discrete JoyStick Motions

Enter 'RT:' followed by any number of valid key, mouse or joystick codes (see below). In the following example, the stimulus 'Stim1' would be displayed and then DirectRT would wait for a response which in this case could be ANY key.

I	J	K
stim	loc	time
stim1	0,0,1	rt:any

In the next case, DirectRT would continue only if the subject hit the 'Z' key (code 44) or the '/' key (code 53). See the codes table below for allowing specific responses. You can list as many codes as you like.

L	M	N
stim	loc	time
stim1	0,0,1	rt:44,53

In addition to the specific key codes, other allowable keypress events are:

- **any** = any key
- **num** = any number key on the regular keyboard
- **numpad** = any number on the number pad
- **alpha** = any character of the 26-character alphabet
- **funct** = any function key

Common Key Codes

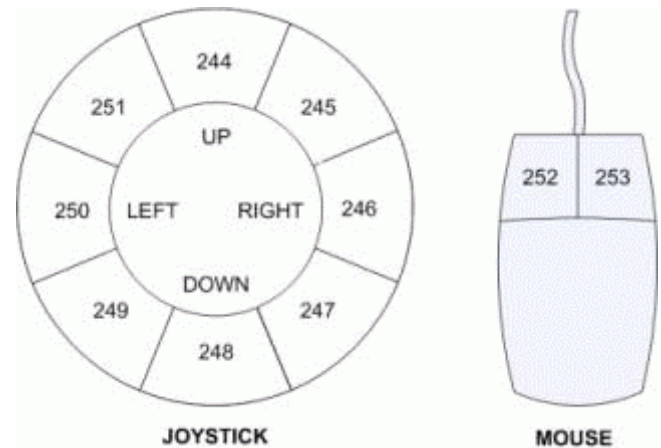
Key	Code	Key	Code	Key	Code	Key	Code	Key	Code	Key	Code
1	1	Pad1	79	F1	59	A	30	N	49	Enter	28
2	2	Pad2	80	F2	60	B	48	O	24	Space	57
3	3	Pad3	81	F3	61	C	46	P	25	LShift	42
4	4	Pad4	75	F4	62	D	32	Q	16	ICtrl	29
5	5	Pad5	76	F5	63	E	18	R	19	lAlt	56
6	6	Pad6	77	F6	64	F	33	S	31	rShift	54
7	7	Pad7	71	F7	65	G	34	T	20	rCtrl	157
8	8	Pad8	72	F8	66	H	35	U	22	rAlt	184
9	9	Pad9	73	F9	67	I	23	V	47	;	39
0	10	Pad0	82	F10	68	J	36	W	17	'	40
				F11	87	K	37	X	45	,	51
				F12	88	L	38	Y	21	.	52
						M	50	Z	44	/	53

Less Popular But Still Loved

Key	Code	Key	Code	Key	Code
-	12	Home	199	PgDn	209
[	26	Insert	210	PgUp	201
]	27	Left	203	Right	205
`	41	lWin	219	rWin	220
=	13	NumLock	69	ScrollLock	70
Apps	221	PadCom	91	SysRq	183
Backspace	14	PadDec	83	Tab	15
CapsLock	58	PadDivide	181	Up	200
Delete	211	PadEnter	156		
Down	208	PadMinus	74		
End	207	PadMult	55		
Esc	---	Pause	116		

Joystick & Mouse Codes

Joystick	Code	Mouse	Code
Button 1	241	Left-Click	252
Button 2	242	Right-Click	253
Button 3	243	Mid-Click	254
Up	244	Other-Click	255
Up-Right	245		
Right	246		
Down-Right	247		
Down	248		
Down-Left	249		
Left	250		
Up-Left	251		



Hint: When assessing response accuracy, the first key code defines the 'correct' response. If pressed, the 'Correct' column in the data file will read 'true'.

Note about joystick codes: You must first make sure your joystick is set up to work with Windows programs. You can set up your joystick by selecting 'Gaming Options' or 'Joystick' in the Windows Control Panel. You might need to 'add' your joystick if you do not see it currently configured there. When you check this, it's also a good idea to calibrate your joystick using the calibration utility provided by Windows. The movement codes above (i.e., 244-251) are triggered as follows: DirectRT divides the joystick range into 8, 45 degree angles (like cutting a pizza into 8 slices). As soon as the joystick has moved more than half of it's range in any direction, DirectRT determines which of the 8 'slices' the joystick is in and takes that as the response and records the RT at that point.

Note about mouse codes: You may choose to have subjects respond with mouse clicks rather than key presses. This can easily be done with the mouse codes above. Note that you will not see the mouse cursor.

Other Response Methods and Features

In addition to the traditional codes, DirectRT provides a number of other response methods and features. Since a number of these methods involve rather lengthy explanations, we will refer you to an illustrative sample for each one that fully explains the feature and simultaneously provides a concrete demonstration.

Request a short open ended response

Record a voice response and reaction time via the microphone

Record continuous joystick movement for a specified duration

Wait for TTL signal from an external device (version 2002.3 and later)

Collect multiple response times per trial

Record KeyRelease time in addition to subsequent KeyPress time

Require Correct Response (e.g., IAT)

Response Feedback (display subject's response on screen)

Skipping

QuickInfo

Skipping allows you to skip a subject to a later specified trial based on their response to an earlier trial.

Details

Whenever you allow for a key press, you can assign a 'skip' value for any of the key codes that are allowed as a response. To do this, simply follow the keycode with a '>' and the trial ID that DirectRT should skip to in the event that that key is pressed.

For example,

L	M	N
stim	loc	time
stim1	0,0,1	rt:44>200,53>210

In this case, you have allowed the subject to hit either the Z (code 44) or the / (code 53) key. You have also told DirectRT that if the subject hits the Z key then you want them skipped to trial 200, and that if they hit the / key, then they should be skipped to trial 210.

Note that the way DirectRT does this is that it literally skips through the trials looking for the requested trial ID. If it is found, then the trials resume, if not, the session ends. You can use this to intentionally end a session at any point during your input file (e.g., `rt:any>9999`).

Also note that skip patterns work on the basis of the trial ID column (the one you fill in) and not the sequential position of the trial (i.e., the row in the spreadsheet).

Unconditional Skipping

In some cases you may want to skip over items unconditionally. The command is the same as above but this is a useful concept that is sometimes overlooked. Essentially, it means that you skip the subject to the same later trial NO MATTER WHAT key they press.

E.g., `rt:any>210`

Or `rt:44>210,53>210`

Unconditional skipping is especially useful for rejoining subjects who have been conditionally skipped to various points in the input file so that they end up at a common place. When they are done their special sections, they can be joined up to the same trial by putting an unconditional skip item at the end of each of their custom sections. In each case they are all skipped to the same trial and continue on with the same set of items.

Advanced Hint

Combining Unconditional Skipping with RWG (Randomization Within Groups) creates a very useful randomization technique. With this technique you can not only randomly order your trials, you can randomly determine which trials your subject gets. Imagine you have created 100 trials and you want DirectRT to randomly choose 50 of these trials and then present them in a random order.

How hard is this? NOT! Just assign every trial an RWG value of 1. The trick? Stick a trial in the middle and give it an RWG value of 0. That means it's position is fixed. Assign an unconditional skip to this trial such that any response will skip the subject over the next 60 items. Done!

You can do this with any set of items located anywhere in your input file. Just assign them all the same RWG value and then place an unconditional skip anywhere in the middle that takes the subject beyond the remaining items in the same RWG set. The earlier you place it, the smaller the random sample of items from the group.

Note that this is just the start. If you want to get really fancy you can do the same with RBG (Randomize Between Groups) and in so doing, randomly select and administer entire blocks of items from within the same input file. And of course you can combine all of these concepts together for one extremely randomized experiment.

Jumping

QuickInfo

Jumping is just like skipping except that you come back. Maybe it should be called bouncing. Hmm..

Details

Jumping allows you to go off and run one trial of your choosing depending on the response a subject gives. This is a convenient way to give feedback and to execute other conditional events that are based on a subject's response.

The command works exactly the same way as the skip command except the arrow goes toward the keycode (i.e., <) instead of away from it (i.e., >)

Instead of continuing from the later item, DirectRT will return and resume from the next trial.

	A	B	C	D	E	F	G	H
1	block	trial	rbg	rwg	style	stim	loc	time
2	1	1	0	0	1	stim1	0,0,1	rt:44<98,53<99
3	1	2	0	0	1	stim2	0,0,1	rt:53<98,44<99
4	1	3	0	0	1	stim3	0,0,1	rt:44<98,53<99
5	1	4	0	0	1	stim4	0,0,1	rt:53<98,44<99
6	1	5	0	0	1	debrief	0,0,1	rt:any>100
7	1	98	0	0	1	correct	0,0,1	2000
8	1	99	0	0	1	wrong	0,0,1	2000

In the example above, each of the allowable responses is assigned a jump value. In this case, trials 98 and 99 are images that convey the subject was correct or wrong in their response, respectively. On each of the trials 1 through 4, it is specified which of these trials (98 or 99) to jump to on each response. As can be seen, the jump trial is displayed for 2000 ms after which the trials resume from where the jump initiated.

Note that you will likely want to insert an Unconditional Skip just before these feedback screens so that they are not displayed on their own. In the example above, a debriefing trial occurs at which time any key will skip subjects over the feedback trials.



About Styles and the Style Editor

Overview of Styles

The purpose of styles is to save you time and effort by storing font, colour and option information in a separate file in which you label every combination you desire with a single code number. You can make as many combinations of fonts and colours and options as you like and then store them each with their own code in a single file. You can use the same 'style file' for all your experiments, or you can copy and modify as many different style files as you like-one for every experiment if you want. If you place a custom style file in the same folder as an input file, then that is the one DirectRT will use. If not, then DirectRT will simply use the default style file located in the main program folder.

Every Trial Has a Style

In every input file, you'll notice that there is a column called 'Style'. Every trial has its own Style value that is set in this column. Before the session begins, DirectRT opens the Style file and reads every style in it. It organizes them by the code numbers you give to them. Then as DirectRT prepares each trial it reads what style you want applied to it. DirectRT says 'Style 12? No problem!' and proceeds to set the colours, the font, the alignment of stimuli and other settings accordingly.

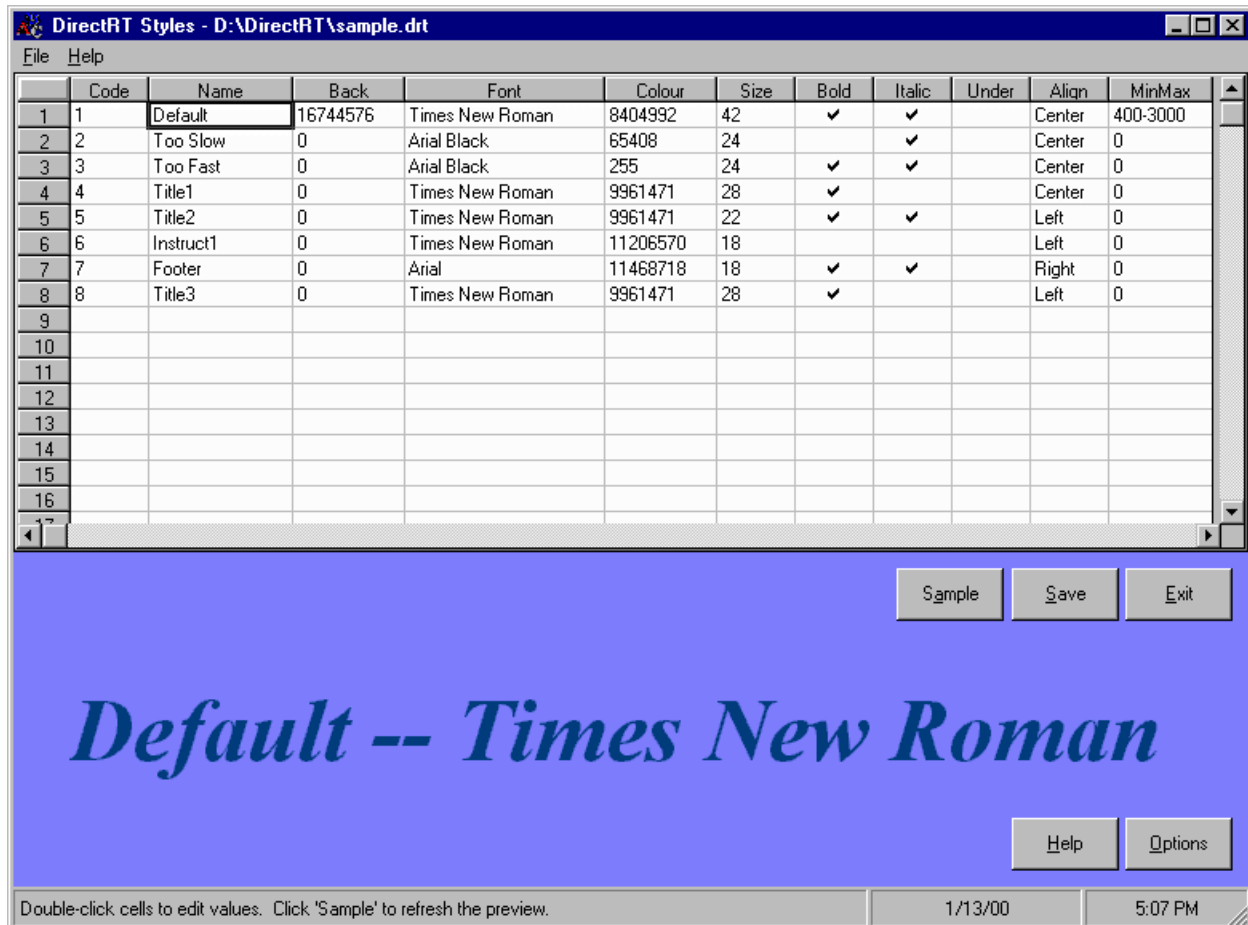
You'll also see that you're not limited to a single style on any given trial. You can switch styles at any time by placing a style codes in later columns of the input file. See XXX for details. You can also place multiple style codes within a single text file in order to present some nicely formatted text. See Formatting Text Files sample for how to do this.

Style Files

Every set of styles is located in a .DRT style file. Edit style files by double clicking on them, or open the style editor from the DirectRT main program window (select Edit->Style File)

This section will explain the purpose of the different style options and explain how to use the DirectRT Style Editor

Main Style Window



Style Menu Commands



You can have different style files for different experiments. By default, DirectRT uses the style set located in the default styles file 'styles.drt.' This file is located in the main program folder (e.g., c:\directRT\styles.drt). You can copy, modify and rename a style file and locate it in the same folder as your input file. DirectRT will first look for a local style file (i.e., one in the same folder as the input file). If it doesn't find one, it will use the default style.drt file in the main program folder.

Open - Select and open a style file

Save - Save the current settings of the open style file

Copy to Folder - Copies the current style file to another location. Note that DirectRT needs to save the current file before it does this. The original will be closed and the active file will become the copy in the new location.

Open Default Style File - Open the default style file (styles.drt) located in the main program folder. This is the style file that is assumed if no file is located in the same folder as the input file.

Open Last Style File Again - Open and edit the most recently opened style file.

Launch Explorer - Launches the Windows Explorer for file exploration and handling.

Open a Recent Style File - Choose from a list of most recently edited style files.

Clear Recent Files List - Clears the list of recent files. Clears the list only-doesn't have any impact on the files themselves.

Code - Assigning a Reference Number

QuickInfo

The style code is a number that identifies a particular style (e.g., font, colours, rt parameters) for trials in the input file. The numeric code is used as a reference in the input file such that when DirectRT encounters a new style code it changes the program setting accordingly.

Value

Any integer

Name

QuickInfo

A text label you can use to give a meaningful descriptor to each code.

Value

Any.

Note: Be sure to refer to the style by its code number and not its name in your input file.

Back - Background Colour

QuickInfo

The Back column in the Style Editor refers to the screen colour upon which text and images are drawn. Double-click to select color from the Windows colour dialog.

Values

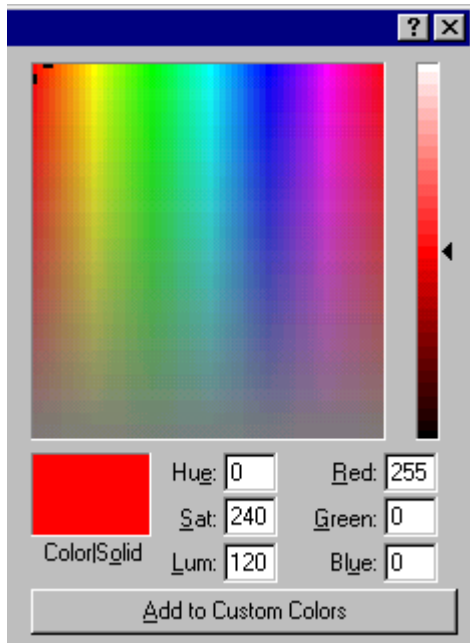
Any integer value in the range of 0 (Black) to 16777215 (White)

Hints

In the Style Editor, when you double-click on the 'Back' field, you will pull up the windows colour dialog.



You can select one of the predefined colours and then click Ok. DirectRT will enter the appropriate colour code in the 'Back' colour column for you. If you need or desire more selection you can choose from the full range of colours available by selecting 'Define Custom Color.' This will pull up the custom colour dialog window:



In this window you can click anywhere on the colour spectrum to choose a color. If you first click on one of the standard colours (above) this will give you a starting point from which you can adjust. Either way, click 'Ok' when you're done to enter the color code. If you think you'll use the colour again you can click, 'Add to Custom Colors' so that it will be available for you as a standard 'custom' color in the future.

Note that if you like a colour but it's too dark or too light it's easy to adjust this using the slider on the right hand side of the custom colour dialog.

Font Settings

QuickInfo

Series of font variables for a given style

Font	Colour	Size	Bold	Italic	Under
Times New Roman	16777088	28	✓		
Arial Black	65408	24		✓	
Arial Black	255	24	✓	✓	
Times New Roman	9961471	28	✓		
Times New Roman	9961471	22	✓	✓	
Times New Roman	11206570	18			
Arial	11468718	18	✓	✓	
Times New Roman	9961471	28	✓		

Font - Double-click to select from fonts available on the current system. Be sure to choose fonts that will also be available on any other machines you'll be using.

Colour - Double-click to select colour with which text will be drawn. Any integer value in the range of 0 (Black) to 16777215 (White). See Back - Background Colour for more details on selecting colours.

Size - Size of the font in points. Values may range between 1 (extremely tiny) and 72 (extremely huge). The font with which this paragraph is displayed is 11 points.

Bold - Check to indicate the text should be **bold**

Italic - Check to indicate the text should be *italicized*

Under - Check to indicate the text should be underlined

Note you can check any combination or all of the **latter three**.

Align - Stimulus and Text Alignment

QuickInfo

Defines how stimuli appear relative to the location point defined in the Loc column of the input file.

Values

Left, Center, Right

Details

In the Loc column of the input file, you define a single point on the screen using horizontal and vertical coordinates.

The Align setting of the current style determines how stimuli (e.g., text and images) appear relative to this point on the screen.

Left - If Align is set to left, then the top left corner of the stimulus is placed at the point. Left alignment is especially useful when instructions are being presented via a text file stimulus.

Right - If Align is set to right, then the top right corner of the stimulus is placed at the point. This is useful for special cases such as Hebrew text and footer text.

Center - If Align is set to center, then the center of the stimulus will be placed at the point. This is especially useful when displaying images and text strings of different sizes and you want them all centered on the same point.

See Also

Loc - Stimulus Location for more details and an example

MinMax - Setting RT Limits

QuickInfo

Defines the range of acceptable response times. RTs under the minimum or over the maximum cause a message screen to appear

Values

0 for none OR two integers separated by a dash (e.g.) 200-1500.

An exclamation mark (!) following the value will cause DirectRT to automatically proceed when the maximum time allowed for the response is reached (e.g., **200-1500!**). Otherwise, DirectRT will wait for the response before delivering the message assigned for exceeding the maximum response time.

Details

In some case you may want to notify participants that they are responding too quickly or too slowly. If so, you can place a MinMax value in the style you are using for that trial. If the RT is under the minimum or over the maximum the participant will be presented with screens that you define in Options.

If you do not want the feedback to occur as a function of response time, then set a MinMax value of 0 (e.g., in the case of instructions).

See Also

Options Window for details on defining the MinMax feedback events.

Options

The screenshot shows a Windows-style dialog box titled "Options". It is divided into three main sections. The first section, "Message: Responding Before Minimum RT", contains a text input field with the text "Please wait for the stimulus!", a "Delay" spinner set to 2000, and a "Style" dropdown menu set to 2. The second section, "Message: Exceeding Maximum RT", contains a text input field with the text "Please try to respond faster!", a "Delay" spinner set to 2000, and a "Style" dropdown menu set to 3. The third section contains two more settings: "Screen Size" with a dropdown menu set to 800x600, and "Trial Interval" with a spinner set to 750. An "Ok" button is located at the bottom right of the dialog box.

Messages

QuickInfo

Defines the feedback events that will occur if the participant responds before the minimum RT or after the maximum RT for a given trial.

Message: Responding Before Minimum RT - This is the message that will be displayed if the participant's response is less than the minimum allowable RT set for that trial.

Message: Exceeding Maximum RT - This is the message that will be displayed if the participant's response is greater than the maximum RT set for that trial.

Delay - This is the number of milliseconds for which the message screen will be displayed before the trials resume. Separate delays can be defined for breaking the minimum and maximum RTs.

Style - This is the style setting for the feedback screen. Separate styles can be used for breaking the minimum and maximum RTs.

Note that these feedback events will not occur if you set the MinMax value of the trial's style to 0.

See Also

MinMax-Setting RT Limits

Screen Size

QuickInfo

Determines the screen resolution in which the DirectRT session will be run. The essential trade off is that text and graphics look better under higher resolutions (e.g., 1024x768), but run faster allowing very short display times under lower resolution (e.g., 640x480).

Values

640x480, 800x600, 1024x768

Details

DirectRT will automatically switch the display to the resolution set here and then set it back when the session is complete. Note that text and images will appear smaller under higher resolutions than they do in lower. This is because it takes less screen space to display the same number of pixels. Make sure you consider what resolution you will run your session in when sizing your images and text. For example, a 640x480 image like ones in the samples will fill the whole screen in 640x480 mode, but only about half the screen in 1024x768 mode.

Trial Interval

QuickInfo

Sets the inter-trial interval in milliseconds.

Value

Any integer, typically in the range of 350-2000ms

Details

As soon as DirectRT finishes a trial it determines the background colour of the next trial, and immediately flips to a screen of that colour and begins the *Trial Interval*. This is the amount of time that must pass before your first stimulus is displayed. During this time, DirectRT goes off to read all of the information for the trial and begins building all of the screens that will be required. This requires TIME! Hopefully, less time than you have requested as the Trial Interval! If so, DirectRT will return with all the screens prepared, and simply wait for the Trial Interval to expire. 1000 milliseconds is a very safe setting.

More Detail if You're Interested

Depending on the number of screens and the intensity of the image requirements and the power of the computer, the time DirectRT requires to load the screens will range.

Because DirectRT does all of this construction in the background WHILE the trial interval is occurring, the construction process will have NO effect on timing as long as you set a Trial Interval that is greater than the time it takes to load everything.

Usually a Trial Interval of 1000ms is sufficient for most computers to prepare most trials, although some computers may need as little as 20-100ms even for complex trials, whereas older computers-well, maybe you should just get new computers :).

You can experiment with this. In the Log file that DirectRT creates, it records the number of milliseconds it takes to load each of your trials. You'll see in every case where the load times were less than your Trial Interval, the Trial Interval was maintained. If there is ever a case where it takes LONGER than the interval to load the trials, then the interval will be extended for as long as it takes to finish preparing the trials.

If you requested a sequence of 10 full screen full color images to flash during a single trial, then a trial interval of 200ms would not likely give the system enough time to prepare. In such a case, the trial would begin as soon as possible and the extra time required would be recorded in the log file so you could adjust your Trial Interval accordingly.

Running from the Main Program

QuickInfo

The easiest way to run an input file is through the main program window of DirectRT (what you see when you run DirectRT.exe). From the file menu, select 'Select and run input file,' or hit Ctrl-S.

Details

From the Files menu of the main program window, you have three ways to select an input file to execute:

Select and Run Input File - If you haven't executed the input file before, this choice allows you to search through the folders on your machine to find the desired input file. Locate the input file and then either double click on it, or select it and click 'open.'

Run Last Input File Again - This choice will execute the input file most recently run on the machine. The key sequence 'Ctrl-R' will perform the same function. This command is handy when multiple participants are being run back-to-back using the same input file, or when you're developing and testing an input file.

Run a Recent Input File - DirectRT keeps track of the ten most recently run input files and displays them in the Files menu. Clicking on any one of them will re-run the selected input file (assuming it's where you left it).

Subject and Condition

After selecting your input file, you will be prompted to enter a subject and condition ID (these are for your own tracking purposes). Enter an integer value for Subject and any string value for Condition and then press OK to start the session.

Running a Range of Trials

To run a select range of trials from the input file, type the range in the Range field. Do this by indicating the first and last rows of the input file you want to run, separated by a dash (e.g., entering 5-15 would cause DirectRT to run the trials located in the 5th through 15th rows of the input file). To run all the trials you can simply enter the word *all*.

To stop a session at any point, press 'Esc.'

Running from a Command Prompt

QuickInfo

You can bypass the DirectRT window and subject/condition prompt by using the command line option.

Details

If you are a programmer and want to launch DirectRT from another program or if you want to run DirectRT from a batch file then this is the syntax:

`"ExecuteDRT.exe /subject,condition,input file,range"`

Where:

- Subject is an integer
- Condition is any string
- Input file is the full path and file name of the input file to run
- Range is the range of trials to run ('all' for all trials)

Examples

- `ExecuteDRT.exe /1,1,c:\directrt\samples\sample1.csv,all`
- `ExecuteDRT.exe /23,2,c:\input files\input1.csv,all`
- `ExecuteDRT.exe /345,c4,c:\input files\input1.csv,5-37`

Running From MediaLab

QuickInfo

With MediaLab2000, you can easily embed a DirectRT session within your MediaLab experiment. Subject and Condition info will be consistent.

Details

You can run your input file from MediaLab if you have version 2000.4 or later installed. Simply select 'DirectRT Session' as a questionnaire item and enter the input file in the File field. MediaLab will pass the current subject and condition Ids to DirectRT, launch the session, and then resume where it left off when the session has completed.

Note that DirectRT will still output its own data files that are separate from the MediaLab data files. You will be able to easily match them up though since DirectRT will use the subject and condition info from the current MediaLab Session.

Important: It is strongly advised that you don't run two DirectRT sessions from MediaLab back-to-back. DirectRT can fail to load if a previous instance of it has not yet had a chance to fully shut down. An easy fix for this is to insert a simple MediaLab instruction item between the DirectRT sessions (e.g., reading 'Press the spacebar to continue').

Advanced Hint

If you only want to run a range of trials then use the c (command line argument) parameter followed by the range (e.g., c12-44), otherwise you can leave the parameters field as 'none'.

Overview of Data Output

For the meaning of specific columns in the data files see also:

Short Data Files

Log Data Files

When DirectRT finishes running an input file, it writes the data file to a 'data' subdirectory located in the same folder as the input file. If the data subdirectory does not exist, DirectRT creates it. Since DirectRT will anticipate you running multiple subjects with the same input file it will create a subfolder with the name of your input file. In this folder will be the data for each subject you run using that input file.

For example, if you have an input file called:

```
c:\input files\input1.csv
```

Then DirectRT will create a folder:

```
c:\input files\data\input1
```

And then it will create a data file for each subject. For example, data from Subject 34 and Subject 35 would be in the following files, respectively:

```
c:\input files\data\input1\34.csv
```

```
c:\input files\data\input1\35.csv
```

These data files are short form data files that are useful for most purposes. DirectRT also creates log data files that contain much more information if you need it (e.g., exactly how long each stimulus was displayed). These files are placed in a log subfolder. For example,

```
c:\input files\data\input1\log\log_34.csv
```

```
c:\input files\data\input 1\log\log_35.csv
```

An Alternative Method

If you have many different versions of an input file and don't want the data from each one put in a separate folder then you can do the following: Name your input files with common name (e.g., **input**) and then use an *underscore and an ID*, e.g., **input_1.csv**, **input_2.csv**, etc.). DirectRT will create folder in the data directory using the common name and will place any data from input file sharing that common name in the same folder. E.g., **c:\input files\data\input**. You will be able to identify exactly which input file the subject received because it's written to the log data file.

Merging Data Files

If you're looking for a convenient method for merging the data files, DirectRT offers a utility called FileMerge (..\DirectRT\FileMerge\FileMerge.exe) to do this rather painlessly. Essentially all you have to do is drag the files you want to merge into the FileMerge window and select 'Merge Files' from the File menu. If you like, you can also save the list of files to merge them again later. A quick tutorial on FileMerge is available by clicking on the FileMerge Help menu. You can also start FileMerge by selecting a 'Merge Data Files' from the DirectRT Tools menu.

Short Data Files

Subj

The subject ID entered manually at the start of the session or passed through another program via a command line prompt.

Cond

The condition ID entered manually at the start of the session or passed through another program via a command line prompt.

Block

The Block ID label assigned to the trial in the input file

Trial

The Trial ID label assigned to the trial in the input file

Order

The ordinal position of the trial in the actual session. This vary as a function of how the trials were randomized.

Optional Fields

If you have used Optional Fields in your input file to identify characteristics of trials they will be written after the Order column.

Resp

This is the keycode associated with the key pressed by the subject. If multiple key presses are required on a single trial (e.g., multiple instruction screens each requiring the subject hit the spacebar), this will be the code associated with the last key required.

Name

The name associated with the key pressed by the subject (e.g., 'A', '3', 'SPACE').

Stim

The final stimulus presented before the reaction time was captured.

RT

The reaction time in milliseconds.

Correct

A true or false value according to whether or not the subject's key press corresponded to the first valid key listed for that trial in the input file.

Log Files - More Data!

Log Data Files contain all of the columns contained in the Short Data File as well as these additional columns:

Date

Date the session was run

Time

Time the session was begun

File

Name of the input file

Started@

Number of milliseconds expired between the start of the first trial and the beginning of the current trial. Keep in mind that the trials are written to the data file in their original order-not the order in which they were presented. If you want to follow the time sequence, you can sort the data file by the **Order** column. This value is the actual difference in the two times and is independent of all other time measures.

LoadTime

The number of milliseconds it took DirectRT to prepare all of the screens for the current trial. This is done during the interval for the trial and does not add to the time it takes for the trial to execute. This is important to monitor though. You want to make sure that this value never exceeds the actual trial interval value that you've specified. This is usually not a problem, but you can verify this by examining the TotalInt value (next column).

TotalInt

This is the amount of time from the end of last stimulus screen of the previous trial to the beginning of the first stimulus screen of the current trial. This value should approximate the value in the **Style file, Options->Trial Interval**. Very slight variations in this value between trials is normal (about 10ms). Timing then becomes precise at the start of the first stimulus. If you find a TotalInt value that significantly exceeds the trial interval you specified, this can mean you have not set a sufficiently long interval such that DirectRT can prepare all of the stimulus screens before your interval is up. In such cases, try extending your trial interval.

Stim n

The name of each stimulus screen subsequent to the trial interval

Time n

The actual display time for each stimulus screen. You might notice that the display time for the final stimulus is usually a little bit longer than the RT value (e.g., 1-15ms longer). This is because the time at which the subject responds is not the exact time at which the stimulus leaves the screen. Depending on how early into the screen refresh cycle the response occurs, the stimulus may be displayed for an additional number of milliseconds.

Doing the math if you're interested

If you want to account for where all the time has gone, then sort your data by the Order column (or the Started@ column) so that the trials are in their temporal order. If you take the trial's start time (Started@) and add the trial interval (TotalInt) and each of the Time *n* values, then this *should* add up to the Started@ value of the subsequent trial.

Do not include the LoadTime value because this is going DURING the trial interval.

Make sure you don't have any missing trial data if you're going to do this (e.g., this won't work if you have omitted trial data from the data file by specifying a Block ID of 0-at least for trials surrounding this missing trial(s)).

DirectRT Samples

A good way to get the feeling of how DirectRT works is to run through the following series of samples sessions and then look at how they were put together. These samples are available from various points in the on-line help file where they are relevant. In the on-line tutorial, wherever you see the **Show Me the ...!** button, you can click to see the relevant sample. You will be able to click to pull up the actual input file to view or even modify it. You will also be able to execute it by clicking on a button to see exactly how it will look in action. Finally, you will be able to pull of the resulting data file, again just by clicking on a button. This is a very fast and effective way to learn the basics of input files.

You can access these samples by selecting 'Samples' from the Help menu of the main DirectRT program window. You can also access the samples *directly* if you like-they're located in the 'samples' folder of the DirectRT directory. When creating your first input files you may wish to copy an appropriate sample folder and simply modify it to meet your needs.

Notes: You will need a spreadsheet application installed such as Excel in order to view or edit the input and data files-even with the tutorials.

Bare Bones (45 min)

Text (p 61)

Images (p 62)

Text Files (p 62)

StimLists (p 63)

Setting Valid Keys (p 64)

Changing Styles (p 64)

Basics (60 min)

More Stim Per Trial (p 65)

The Basics of Style (p 65)

Locating Stim (p 66)

Compound Screens (p 67)

Formatting Text Files (p 67)

Optional Fields (p 68)

Ordering Trials (30 min)

RWG (p 69)

RBG (p 69)

RWG + RBG (p 70)

Skipping (p 70)

Jumping (p 71)

Applications (30 min)

Pretesting Images (p 72)

Standard Priming (p 72)

Priming with Images (p 73)

Assessing Modern Racism (p 73)

More Samples

The samples above are those intended to give you a good basic understanding of how DirectRT works. There are other useful features of DirectRT though that you can explore later in the following samples:

Additional Response Features

Request a short open ended response (p 74)

Record a voice response and reaction time via the microphone (p 88)

Record continuous joystick movement for a specified duration (p 77)

Wait for TTL signal from an external device (version 2002.3 and later, p 91)

Collect multiple response times per trial (p 83)

Record KeyRelease time in addition to subsequent KeyPress time (p 80)

Require correct response (e.g., IAT, p 79)

Response feedback (display subject's response on screen, p 82)

Use mouse buttons for input (p 76)

Record discrete joystick responses (p 76)

Additional Stimulus Features

Play sounds files (p 86)

Use subjects own responses as subsequent stimuli (p 74)

Clear the screen during a response time interval (p 81)

Use randomly selected stimuli again on later trials (p 84)

Send TTL signals to external equipment for physiological synchronization (e.g., SCAN, p 91)

Implicit Association Test (IAT) sample (p 93)

Advanced graphics sample (p 94)

Tutorial 1 Samples: Bare Bones

Sample 1a: Text

In this first sample, you'll see the bare minimum it takes to run a DirectRT session. There are no external stimuli and no stimulus lists. **Everything** is located in the input file. Click above on **Show me the input!** To see what the input looks like.

The block and trial columns are for your own reference. They'll be written to your data file. The trial column also serves a useful purpose when you want to use skips and jumps. The block column can also indicate you don't want data written to the data file if you use a **0**.

The **bgr** and **wgr** columns are for randomization. Currently they're all set to **0** which means that no randomization of trials will occur.

The style column indicates that all trials are to be presented in style **1**. More detail on styles will follow shortly in another sample.

Then come the three critical columns: **Stim**, **Loc** and **Time**. In this sample there is only one set for each trial.

In the Stim column, the ~ indicates that the stimulus is the actual text in the cell. The Loc column tells DirectRT where to place the stimulus and whether to clear the screen or append the stimulus to the previous screen. 0,0,1 is a shorthand way of telling DirectRT to center the stimulus and to present it on a fresh screen.

Finally, the Time column tells DirectRT what to do after the screen is displayed. In this case, 'RT:any' indicates that it should wait for any key response.

Try running the input file by clicking on **Show me the session!** When you see each word appear, press any key and your response time will be recorded.

Now, let's look at the data you just produced. Click on **Show me the data!** Note that there are 2 data files that DirectRT produces for each session.. a simple one useful for most needs, and a more detailed one called a **log** file that records everything including actual display times for your various stimuli. In these tutorials, it is the **log** file that you will see when you click on **Show me the data!** Most of the columns in the data file should self-explanatory. The 2 most important columns are name and RT. This tells you which key you pressed and your reaction time. See Output Files for the details regarding any specific column.

That's it! That's a bare bones DirectRT session. As an exercise, click on **Show me the input file!** and change some of the stimulus values (e.g., change dog to puppy) save the file and re-run it. Be sure to save the file in the **.csv** format even if your spreadsheet application warns you that you'll lose formatting. The **.csv** format is critical because that's what makes the input files independent of any particular spreadsheet application.

Now we'll start adding some more DirectRT features...

Samples: Images

Click on **Show me the input file!** to pull up our next sample. Notice that this sample is **identical** to the previous **text** sample, with the exception of the **Stim** column. Notice that there is no ~ before each stimulus. *If there is no special character preceding the stimulus, this tells DirectRT that you want to display an image.* You do not need to add the **.bmp** to the end, because DirectRT assumes it (although you can put it there if you like).

By default, DirectRT assumes that all stimuli are located in a **Stim folder** located in the same folder as the input file. *So if the image is in the **Stim folder**, then all you have to do is enter the name of the image.* That's what's happening here.

If it's located somewhere else, then you could specify the exact location (e.g., **c:\images\pic01**). If the image is located in a folder *within* the Stim folder, then you just need to add the name of that subfolder (e.g., **faces\pic01**).

Try running this session and see what happens. You see that it works the same as the text sample, except that images appear instead of text.

As an exercise try dropping a **.bmp** image of your own in the **directRT\samples\1 bare bones\stim** folder, and change one of the images in the **Stim** column of the input file to the name of your image file. Save the file and re-run the session and see what happens. Note that this session runs in 640x480 mode so an image of those dimensions or less will work best. We'll look at changing display modes a little later.

As an additional exercise, try adding a ~ in front of one of the image names and see what happens. Can you guess? ☺

Hint: for exploring folders and copying files, you can just click on the **Explorer** button in the Main Help window, or click on **Launch Explorer** from the **File** menu of any DirectRT application.

Samples: Text Files

Take a look at the input file for this sample. In this example, notice that the input file is identical to the previous sample except that we've added two trials at the beginning and one trial at the end. In these new trials, look at the Stim column. There is a **#** character in front of each stimulus. This indicates that stimulus is an entire text file. As with images, you don't need to put the **.txt** at the end, because DirectRT assumes it.

Try running the session and see what it looks like. This is a really handy way to present instruction screens or other large quantities of information with a minimal amount of effort.

As an exercise, try opening the file `directRT\samples\1 bare bones\stim\instruct1.txt` and modify it. Save it and then re-run the session.

Later we'll see how to easily change the text alignment and text formatting so you have more control over how the text file appears.

Samples: StimLists

Rather than specifying a specific stimulus on every trial (as in the previous three samples), you can opt to request that the stimulus be chosen from a **stimulus list or StimList**. StimLists are text (.txt) files that you create with any text editor that simply list stimuli. This can be especially handy if you want to randomize the stimuli that fill particular positions in your input file. There are three types of StimLists:

In the first example, notice that the input file is identical to the last one except that instead of identifying specific stimuli in the Stim column, we have replaced the image names with **&pics**. In the stim folder is a file called `pics.txt`. This file simply lists all of our images just as we would have entered them in the stim column. The **&** symbol tells DirectRT that you want to *draw stimuli from this list sequentially*-i.e., in the order they appear in the text file. Try running the session and see how it looks. We've labeled each image so you can see that they are appearing sequentially as specified in the list.

Now, open the next input file. Notice that it's the same as the last except that instead of the **&** symbol, the StimLists are preceded by the **%** character. This tells DirectRT to *randomly select a stimulus from the list*. Try running the session. Notice that since this is **True Random** selection, some of the images may appear twice or more. Then click, **Show me the data!** and take a look. You'll see that the **Stim1** column tells you which image was selected and presented.

Finally, take a look at this input file. This time, we use the **\$** symbol. This again tells DirectRT we're using **StimList**, but that we want the stimuli *selected randomly without replacement*. DirectRT will make sure all of the stimuli are used before it re-randomizes the file and starts over.

As an exercise, try adding a stimulus or two to the `directRT\samples\1 bare bones\stim\pics.txt` StimList in any text editor. Try entering an image that's located somewhere else on your hard drive, such as **c:\images\mypic**. Or add a text item such as **~I love pizza!**, or a text file such as **#instruct1**. Place the additional stimuli anywhere in the list, save it, and re-run any of these session and see what happens.

Samples: Setting Valid Keys

In the previous samples, you have been able to hit any key and the session would continue. Often you will want to limit subjects to particular key presses.

Open the input file for this sample and see what's different from the last one. Notice that instead of **RT:any**, we now see **RT:num,numpad** indicated for the image trials. Try running this session and see what happens. When you see an image try to press any key other a number. **Num** allows the regular keyboard numbers and **numpad** allows those on the keyboard number pad (if it has one).

You set valid keys by separating them with **commas** after the RT:

All of the valid key codes can be accessed by clicking on **KeyCodes** at the top of the main Help window, or by clicking on **KeyCodes** in the help menu of the main DirectRT program. Try this. For example, see that the code for the **Z** key is **44** and the code for the **/** key is **53**. If you want to limit responses to either of those two keys, you would enter **RT:44,53**. Try changing the one of the values to **RT:44,53** (or any other codes) and see what happens. You can list as many codes as you want.

Notice that for the instruction screens we have left the value at **RT:any**. This is because for instructions we don't really require any specific key be pressed. You could optionally limit it to the **spacebar (57)** or the **enter key (28)** or other key(s).

Hint: Whichever code you list first defines the '**correct**' key for the trial. If the subject hits the '**correct**' key, then a value of **TRUE** will be written to the **Correct** column of the data file. Other valid keys will produce a value of **FALSE**. This is useful of course only when you have one correct value and can be ignored otherwise.

Samples: Changing Styles

This example shows how you can apply different styles to different trials. First, click on **Show me the styles!** Click anywhere on Style 1 and see that the preview window shows what you've been seeing so far. Click on Style2 and Style3 and see how they differ.

Now click on Show me the input! And see that we've changed the Style values for the first two instruction screens and for the debriefing trial. Run the session and see what happens. Notice that the styles of the instruction and debriefing trials have changed to reflect the styles assigned to them.

As an exercise, try modifying any of the three styles. You can change the font, the foreground and background colours and the alignment of the text. Save the style file and re-run the session. Or, create a new style and enter the style code for any one of the trials. Congratulations! You have finished the Bare Bones!

Tutorial 2 Samples: Basics

Samples: More Stim Per Trial

Now that you've been through the fundamentals, we're ready to start adding some zest to our study. In this sample, you can see how you can use as many stimuli on a single trial as you need. For each stimulus, you simply add another set of Stim, Loc and Time cells to the trial row of your input file.

Check out the input file for this sample. Notice that it's very similar to the ones we were just using. The difference is that on trials 3-10, we've **inserted** two extra stimuli. Can you tell what these stimuli will look like?

First notice, that we inserted **~XXXXXXXXXX** as the first stimulus for each of these trials. Since the ~ indicates that the actual text in the cell should be presented, the string of X's will appear in order to focus the subject's attention on where the next stimulus is about to appear. The time column tells DirectRT to display this stimulus for **500ms**.

Next, we see an adjective as the next stim in the sequence for each trial. The adjectives also are preceded with ~ indicating that the text in the cell is the stimulus. The time column for the adjectives on each trial indicates the adjectives will appear for **100ms**.

Finally, we see the final stim on each of these trials is **\$pic**, which you saw in previous samples. Again, the \$ symbol indicated that pic is a text file (pic.txt) listing stimuli from which DirectRT will draw randomly without replacement. In this case pic.txt again contains the names of image files. The time column for the pictures indicates the images will be displayed until a response key is hit, and that only the Z (code **44**) or / key (code **53**) will be considered valid.

Important: Note that you can have as many stimuli as you like on any given trial. The important thing to know is that you need to enter a ***** in the final cell of the row. This is a terminator, telling DirectRT you're done with that trial at that point.

Samples: The Basics of Style

This next sample is actually the same as the last. However you'll see how to apply a few other aspects of the style file. Click on Show me the styles! You've seen most of this by now.

Take a look at the MinMax value of the different styles. See how only Style 1 has a value for this? This tells DirectRT that you want to set a minimum and maximum RT value for any trial to which this style has been applied. Typically you wouldn't set a MinMax value for instruction screens.

Now click on the Options button in the lower right. There are a couple of important variables you can change here.

First you can see the messages that appear for trials on which subjects respond too slow or too fast. Try changing these messages to something you'd like. You can also change the delay-that is, how long the message stays on the screen. You can also change the style that's applied to the message. Try setting each to one of the other style codes in your style file.

You can also set the trial interval here. This is the amount of time that passes between the last stimulus of the last trial and the first stimulus of the next trial. It's important that this be sufficiently long for DirectRT to load and prepare the next trial if you want your trial interval to be constant.

Finally, you'll see here that you can change the display size that your session runs in. These samples were prepared to be run in 640x480 mode, but try selecting 800x600 or 1024x768 just to see what happens when you rerun the input file. Just make sure to switch it back for the remaining samples.

Samples: Locating Stim

This sample shows how any stimulus can be precisely located anywhere on the screen. This input file is the same as the last with the exception of the **Loc** columns. Remember the Loc column has three variables in it: **x**, **y** and **clear** each separated by a comma. X and y define a single point on the screen that determines where the stimulus is placed. You can enter in actual pixel values (e.g., 200, 150 etc.) but an easier way is to use percentages (e.g., .3, .7). For example if you want to define a point that is 30% across the screen and 70% down, you would just enter **.3,.7** for the x,y values. **Note you can always enter 0,0 and this will automatically center the stimulus. Entering 1,1 will place it at the left, top corner.**

Run the session and watch how the stimuli are located as a function of the Loc values.

In this sample the **x,y** point determines where the **center** of the stimulus is placed. If you're interested, you can read more about locating stimuli for more detailed instructions and other options.

Finally, **Clear** defines whether to *clear the previous screen* before drawing the stimulus (**Clear = 1**), or whether to *add to it* (**Clear = 0**) We'll look at the clear values and what they can do in the next sample.

Samples: Compound Screens

This is a tricky but pretty powerful concept. Open the input file and see that it's the same as the last except that we have added a new stimulus to each trial. Just before each image appears, we've added an image called **scale1**. This is a graphic of a scale that we'd like to have appear on the same screen as the reaction time stimulus (i.e., the faces).

To do this, we insert **scale1** as a stimulus and locate it where we want it. We've told DirectRT here to place it at $x = .5$, $y = .95$, which is telling it to place the center of the image half way across and 95% of the way down the screen.

The trick? Notice the Time column for the scale image. It's 0! This tells DirectRT to create the screen but not to display it! Why would want to do this?? Because if on the NEXT screen we tell DirectRT to NOT clear the last screen, then this previous screen will serve as a BACKGROUND for the next stimulus.

Sounds complicated but it really isn't. All this means is that by using 0's in the Clear value of the Loc column you can have layers upon layers of stimuli appending to the same screen. If you set the Time value of any of these stimuli to 0, then they will appear at the same time the subsequent stimulus appears.

I hope this is making sense.

Finally, notice that the on the last two trials (before the debriefing) the clear value in the loc column of \$pic is set to 0 instead of 1. Notice how when you run the session, the adjective stays on the screen when the picture appears. As an exercise, try setting the Time value of one of these adjectives to 1000 instead of 0. See that this makes the adjective appear for 1000ms before the image appears, but still remains on screen because it is never cleared in any subsequent Loc column.

Samples: Formatting Text Files

Congratulations for making it this far! The hard stuff is over and things will start moving a lot faster now. This input file is the same as the last except that it calls some different text files for the instructions and debriefing.

Check them out: instruct3 instruct4 debrief2

This is to show how you can insert style codes from your style file in the middle of text files. Try running the session. As an exercise, try changing the text or the styles and see how it looks. Or create a new style and then refer to in your text file.

Samples: Using Optional Fields

The final lesson in the Basics is to show you about Optional Fields. Open the input file and check out the new column. There's a new column inserted called '!Prime' This is totally for you so that you can identify which trials are which when analyzing your data. **You can add as many of these optional fields as you want. Just make sure they all go after the trial column and that column header always starts with !**

For example, in this input file, the '!Prime' optional field indicates which trials have a positive adjective as a prime and which have a negative adjective. Since optional field columns are written to the data file, this will make analyzing the data much easier.

As an exercise, try adding another Optional Field column beside the !Prime column and entering some values for each trial. Run the session and then click on **Show me the data!** to see that your new optional field was written.

Congratulations! You have finished the Basics!

Tutorial 3 Samples: Ordering Trials

Samples: RWG

The next three samples show some of the basic methods for the quick and easy randomization of trials.

The first method, **RWG** stands for *Randomization Within Groups*. Open the input file and look at the RWG column. Every item that shares the same RWG value with another belongs to the same RWG group. Items sharing the same RWG value will be randomized with each other no matter where they are in the list of trials. The only exception is if an item is assigned an RWG value of 0, then it will stay exactly where it is. Often, instruction trials will be given an RWG value of 0.

Try running the session and see what happens. Notice that the stimulus for each trial is the trial number itself so that you can observe the effects of the randomization.

As an exercise, try randomizing all the trials by assigning them each an RWG value of 1.

Samples: RBG

A second basic method of randomization, **RBG**, stands for *Randomization Between Groups*. As with RWG, every trial that is assigned the same RBG value belongs to the same RBG group. The difference is that instead of randomizing the trials within the group, all the items are treated as a set to stay together. DirectRT will then randomize the order in which the different RBG groups is presented.

Open the input file and take a look at the RBG values. Notice that there are three RBG groups (1, 2 and 3). Run the session and notice what happens to the ordering of the trials. You'll notice that the order of the trials within each group doesn't change-just the ordering of the groups themselves. Also notice that any trials with an RBG value of 0 never moves. DirectRT knows not to include any trial with a value of 0.

Important: In order for trials to share the same RBG value, they all need to be together in one continuous set.

Samples: RBG + RWG

Can you guess what's next? That's right, we can *combine* **RBG and RWG** values to randomize the order in which sets of items occur (RBG), and also the order of the trials *within* those sets (RWG). Open up the input file and you can see that we have values for both RBG and RWG.

Run the session and see what happens. Can you predict?

Notice trials 11-14. They all belong to the same RBG block which means that set of trials may come 1st, 2nd, or 3rd out of the three RBG groups. However, unlike set 1 and 2, they all have an *RWG* value of 0. That means they will appear in their *original* order within the group. In contrast the trials within RBG groups 1 and 2 will be randomized.

Note also how trials 1, 6 and 10 each have 0's for both RWG and RBG. This means they will stay exactly where they are-in the 1st, 6th, and 10th position respectively.

As an exercise, play with the RBG and RWG values and see if you can predict the effects on the session. Just remember that the only constraint in doing this is that trials sharing the same **RBG** value need to be located next to each other in the input file.

Samples: Skipping

In some situations, you might want to skip participants over a series of trials depending on their response to a given trial. To do this, simply use the symbol **>** and the **trial ID** you want to skip to, after the appropriate keycode. For example, imagine you have defined the **Z** key (code **44**) and the **I** key (code **53**) as valid keys. Normally you would do this by entering **RT:44,53**. However, if you want DirectRT to *skip to Trial 34 if the participant hits the Z key (code 44)*, then you could enter **RT:44>34,53**.

Open up the input file for this sample. Notice that the first trial presents a text file **#preference** (i.e., preference.txt). This text file contains a question asking participants which they find most attractive: 1) men, 2) women or 3) food.

Following this question are three groups of trials each presenting stimuli from one of these three categories (i.e., men, women or food). The trials involving men start at trial 2, those involving women at trial 7, and those involving food at trial 12. We want to skip the participant to one of these sections depending on how they respond to the preference question in trial 1.

Notice on the preference question (Trial 1) the valid keys are 1 (code 1), 2 (code 2) and 3 (code 3). If they respond with a 2 (i.e., code 2), this means they have indicated they find women most attractive of the three. Therefore, to skip them to the section on women (which starts at Trial 7) we indicate this with 2>7.

If they hit 3 (code 3), then we will skip them to the section on food by entering 3>12.

Since the section on men immediately follows the preference question, we do not have to use a skip value for anyone who says they find men most attractive.

Finally, notice that there is a simple single screen stimulus at the end of each section. This is an easy way to have participants all meet up at the same place after their section is complete. The screen presents a text file #continue (continue.txt) which simply says to press the spacebar to continue. We set the spacebar to be the only valid key (code 57) and we tell DirectRT to skip to trial 17 (the debriefing screen) when this key is pressed.

Samples: Jumping

In this next sample we see how to jump to a later trial depending on a subject's response, execute it and then return where we left off. This is an easy way to administer feedback for particular responses.

Open the input file and notice that it is basically the same as the one we used for skipping. The main difference is that for certain key press events, we have defined a jump event (you may have to widen the last **Time** column to see everything).

In order to skip to later trials, we use the > symbol. In order to **jump** to a later trial and then *come right back*, we the < symbol.

Notice that we have added two new trials at the end. One presents the text 'Incorrect Response!' and the other presents 'Correct Response!' We have also indicated a series of jump instructions. Specifically, notice that on some trials, if the subject enters a 5 (code 5) then DirectRT will execute Trial 18, and on other trials Trial 19. This way we can give feedback to subjects telling them that their response was correct or incorrect.

Try running the session. Notice that you'll only jump to one of the feedback trials if you hit a key for which a jump was defined. In most of the trials here, that will only happen if you hit 5 (Very attractive). However, in the food trials, we've also included jumps if the subject enters a 1 (Very unattractive).

Important: Notice that on the debriefing trial, we have added a skip value (**RT:57>999**). This is done so that when DirectRT gets to this trial, hitting the space key (code 57) will skip the subject out of the session entirely (since there is no trial 999). This way, the feedback trials are never presented on their own.

Legal Note: Judgements of correct and incorrect for the purpose of this sample were assigned randomly and in no way reflect upon the actual attractiveness of the individuals or objects presented.

Congratulations! You have finished the Ordering Trials samples!

Tutorial 4 Samples: Applications

Samples: Pretesting Images

This sample is a simplified version of the samples used to illustrate skipping and jumping (it excludes the skips and jumps). It is included here because it illustrates a quick and easy way to pretest or gather ratings of images. By combining text files that contain the rating scales and the images to be rated on a single screen, this type of input file can be prepared very quickly.

Click to view one of the rating text files that are used for these trials:

ratew.txt ratem.txt ratef.txt

As an exercise, try changing the response options, save the file and re-run the session.

Notice too that this session is run in 800x600 mode (see **Options** in the style file)

Samples: Standard Priming

This sample illustrates a standard priming paradigm with very short priming intervals. Open the input file and notice that the entire session is performed by randomly drawing text stimuli from text files located in the experiment's stim folder.

Remember that **\$** instructs DirectRT to randomly select from a StimList without replacement until all the stimuli have been used. Click [here](#) to view the StimLists that are accessed.

pos.txt neg.txt neu.txt

Note that the prime is both forward and backward masked by using two different text stimuli: ~XXXXXXXXXX, and ~&&&&&&&&&. To change the masks, just change the characters.

As an exercise, play with the display times. Currently, the first mask is displayed for 500ms and the second mask for 200. Try changing the mask characters and see what effect that has on your ability to view the prime.

See also that the order of the valid keys is different for the trials that have positive versus negative targets. This is done because the first key defines what is the 'correct' response. This is not required but it is sometimes useful for data analysis since whether or not the 'correct' key was pressed is written to the data file.

Note also, that this input file utilizes Optional Fields. In this case, 2 additional columns have been inserted, **!prime** and **!target**, that contain stimulus factor information. These are handy for data analysis and are included in the short version of the data file (run the session before viewing):

Short version of the data file

Another important thing to note is the stimulus duration of the prime. It is currently set to 10ms. Remember that DirectRT displays stimuli for a number of screen display cycles. If your monitor takes 16ms to refresh its screen, then this is the minimum a stimulus can be displayed. However, you can enter whatever value you want. DirectRT will round the display time to the nearest screen refresh. Actual display times can be viewed in the log data file (which you can view by clicking on **Show me the data!** above). This way you can see if your system is capable of the display rates you are requesting. See How DirectRT Works for more detail.

Samples: Priming with Images

This sample is similar to the standard priming example except that images are used as primes and targets instead of text. The display times are different but they don't have to be.

This sample also illustrates the use of using PowerPoint graphics within your DirectRT sessions. This is very easy and quick to do. See Tips and Tricks for more details. Notice how multiple PowerPoint graphics are placed sequentially in a single trial, with each requiring the press of the spacebar to continue. Also note how the Block value of this trial is set to 0 which means it won't be written to the data file.

Samples: Assessing Modern Racism

This sample illustrates a very easy way to create and administer questionnaires. Check out the input file. Notice that the graphic called scale (scale.bmp). This is a standard scale image we will use for all of the questionnaire items. We place this on each screen and then follow it with questionnaire items that are drawn randomly without replacement (\$) from a file called race (race.txt).

This illustrates the concept of compound screens (see Compound Screens Sample). By using a Time value of 0 for the scale graphic, and a clear value of 0 for the subsequent questionnaire item, they are combined into a single screen that is presented at the same time.

Congratulations! You have finished the applications samples!

Other Features

Open Ended Responses

Location:

...\DirectRT\samples\5 other features\01 open ended responses

Sample Description

This sample demonstrates how to request an open-ended response using an anagram task. In this particular task the subject receives a very brief prime before each anagram to see how it will influence their solution. But how do we allow subjects to enter their answer when it requires more than a single key response?

Take a look at the input file, **anagram.csv**.

The anagrams are presented in columns O through Q. For example, the anagram **odg** is presented half-way across the screen and about two-thirds of the way down on a fresh screen for 1000ms.

At that point, we request an open-ended response in columns R through T. First, we specify what the question should be in the **Stim** column (e.g., What is your answer?). The x and y coordinates will always be entered as 0's for an open-ended question, but you *can* specify whether you want the screen cleared (0,0,1) or not (0,0,0). In this case, we specify (0,0,0) because we want the previous anagram stimulus to stay on the screen. Finally, in the **Time** column, we enter **rt:text** to indicate an open ended response.

O	P	Q	R	S	T
stim	loc	time	stim	loc	time
~odg	.5,.65,1	1000	~\What is your answer?	0,0,0	rt:text
~tca	.5,.65,1	1000	~\What is your answer?	0,0,0	rt:text
~ihio	.5,.65,1	1000	~\What is your answer?	0,0,0	rt:text
~dbri	.5,.65,1	1000	~\What is your answer?	0,0,0	rt:text

Open Ended Responses as Stimuli

Location:

...\DirectRT\samples\5 other features\02 open ended responses as stimuli

Sample Description

This sample illustrates two features: (1) Taking open-ended responses, and (2) Using subject responses as subsequent stimuli.

On trials 2 & 3, the subject is prompted with 'What is your name?' and 'What is your favourite food?' respectively. In the Time column, we enter 'rt:text' to indicate we want an open ended answer. Whatever the subject types here will be stored as the response. See the open-ended responses sample for more detail on this procedure.

What follows are a series of priming trials. On most of the trials we are randomly selecting primes and targets from stimulus lists (e.g., **\$pos** and **\$neg**). Notice though that on trials 6, 9, 13 & 17, we ask DirectRT to use the subject's prior responses as stimuli.

\$neg	0,0,1	rt:53,44
~=2	0,0,1	rt:44,53
\$neg	0,0,1	rt:53,44
\$pos	0,0,1	rt:44,53
~=3	0,0,1	rt:53,44

On trials 6 and 13, we have entered a **Stim** value of **~=2**. This means 'present a text stimulus that equals the response from trial 2'. Consequently, the subject's name will appear as the target on these priming trials. On trials 9 and 17, we have entered a **Stim** value of **~=3**. This means 'present a text stimulus that equals the response from trial 3'. Consequently, the subject's favourite food will appear as the target on these priming trials.

You can also use this feature to present images, sounds or text files with filenames that match closed-ended responses. To illustrate, imagine that on trial 3, instead of an open-ended response, you offered 5 different foods for the subject to choose from and they had to indicate their favourite food by pressing a letter from A to E. You could present their response as a stimulus later on by creating 5 different text files-a.txt, b.txt, c.txt and so on, with each containing the response associated with that letter.

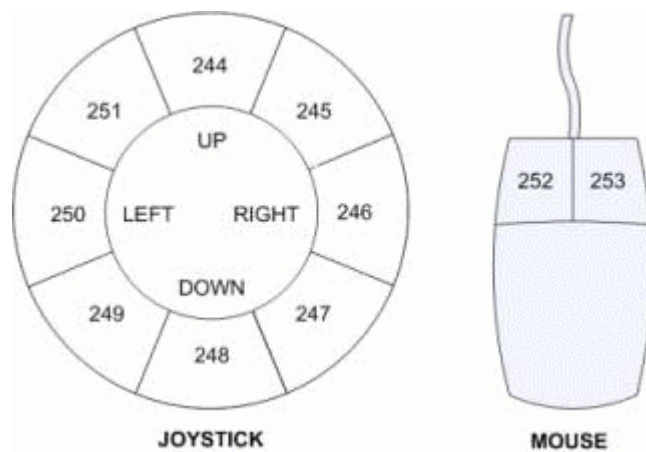
Joystick and Mouse Input

Location:

...\DirectRT\samples\5 other features\03 joystick and mouse input

Sample Description

The joystick and mouse input functions are fairly intuitive if you've played with the regular keyboard codes. Each 45 degree joystick region has its own code that acts just like a keyboard code. Similarly, each mouse button has its own code. The codes are as follows:



Also:

Joystick buttons (if applicable):

Button 1 = 241

Button 2 = 242

Button 3 = 243

Other mouse buttons (if applicable):

Middle button = 254

Fourth button = 255

The 'priming with joystick.csv' sample is the standard priming example but instead of hitting the z or / key, you move the joystick up for negative and down for positive (simulating an approach/avoidance reaction). You might have to calibrate your joystick if you haven't already done so. In the Windows control panel, select 'gaming options' or 'gaming controllers'. If you don't see your joystick there you'll have to add it (e.g., by 'adding' a 2-axis, 2-button joystick or whatever you have). Then click on 'properties' and 'settings' and then click on 'calibrate'. This will let you know that your joystick is working. Then you should then be ready to try the joystick sample. You'll notice in the resulting data file that the joystick response is coded as correct or incorrect depending on what response the subject made. As with regular keycodes, the first code listed tells DirectRT which code is the correct response.

The 'priming with mouse.csv' sample is identical except that it uses mouse input. Instead of hitting the z or / key, you click the left or right mouse button.

See also:

'joystick continuous' sample for details on recording continuous (rather than discrete) joystick motion.

Joystick Continuous

Location:

...\DirectRT\samples\5 other features\04 joystick continuous

Sample Description

DirectRT will allow you to continuously monitor (in 10ms intervals) the motion of a joystick in response to any stimulus. Note that if all you want to do is capture a discrete joystick response (e.g., up, down), then check out the sample titled 'joystick and mouse input').

First, you'll have to have a joystick connected to the back of your computer. Second, you'll need to make sure the joystick is setup in Windows. To do this, go to the windows Control Panel and select 'Gaming Options'. In the 'Controllers' tab, click 'Add' and select the type of joystick you're using (e.g., 2-axis, 2 button). Then click 'Properties, Settings, Calibrate' from the same tab. Calibrate the joystick according to the instructions and now you're ready to try the sample.

This sample displays images and records negative and positive responses via the joystick using push and pull motions respectively. It will assess joystick action

instead of waiting for a key response. Try the sample and when each picture appears, move the joystick around. After 8 images, the session will end.

So how do you create the experiment? If you open the 'joystick.csv' input file you'll see it looks just like any other input file. The difference is that in the Time column, you request 'RT:joy2000' for a joystick response of 2000 ms (or RT:joy1500 for 1500ms etc.). That's the only difference. That tells DirectRT to assess continuous joystick responding instead of waiting for a key or other type of input.

I	J	K
stim	loc	time
pos01	0,0,1	rt:joy2000
neg01	0,0,1	rt:joy2000
pos02	0,0,1	rt:joy2000

Since monitoring continuous joystick motion creates **many** data points, a separate data file is dedicated to storing these data. You will find the resulting data in the sample's data folder in a file called 'joystick.txt'. To open this file in Excel, you will need to select 'All Files' in the 'Files of Type' box when trying to find it. Then, tell Excel you want to open it as a tab-delimited data file. Note that you can also open tab delimited data files in SPSS in case you have more than 255 variables which is Excel's limit. This will be necessary if you monitor the joystick for much more than two seconds since a data point is written every 10ms (i.e., 2 seconds = 200 data points).

This is an illustration of what the data might look like:

A	B	C	D	E	F	G	H	I	J	K	L	M
subj	cond	block	trial	stim	value	low	low_rt	high	high_rt	@0	@10	@20
1	1	0	1	pos01	x	51	500	75	1190	51	51	51
1	1	0	1	pos01	y	50	490	53	1930	50	50	50
1	1	0	2	neg01	x	23	1410	51	500	51	51	51
1	1	0	2	neg01	y	50	490	50	490	50	50	50
1	1	0	3	pos02	x	51	500	96	1380	51	51	51
1	1	0	3	pos02	y	50	490	57	1470	50	50	50

Each trial has two rows--a series of X-values and a series of Y-values. Each row tells you the Subject and Condition IDs from the session. It labels the Block and Trial IDs as well. It then tells you the stimulus (image names in this case) and whether the row contains the X values or Y values. It then tells you the lowest

value from 1-100 that occurred on that trial. 1 is far left for X and far down for Y. 100 is far right for X and far up for Y. It then tells you the time at which that lowest value was reached in milliseconds and does then the same data are shown for the high values.

Then, you will see the raw data. These data show you the X (or Y) position of the joystick in 10ms intervals across the stimulus display interval (in this case, 2 seconds)

Require Correct Responses

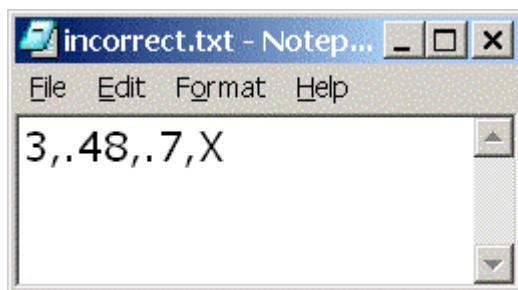
Location:

...\DirectRT\samples\5 other features\05 require correct response

Sample Description

In some cases you may want to require that the subject respond with the correct key before continuing. To accomplish this you can create a short text file called 'incorrect.txt' in your session folder that tells DirectRT to take a special action when valid but incorrect keys are hit. Using this method, if the subject hits a valid but incorrect key then a prompt that you define will appear to tell subjects their response was incorrect and that they need to hit the correct key to continue.

Try running the sample. It is the standard priming sample with the exception that the following text file called 'incorrect.txt' is located in the session folder:



If DirectRT sees the 'incorrect.txt' file in your session folder, then it will know to require a correct response on trials. This text file contains 4 values. The first is the style ID that you want to apply to the prompt that occurs when an incorrect response happens. The second value is the x-coordinate for the prompt and the third is the y-coordinate. The fourth value is the prompt itself. So in this case, when an incorrect response occurs, an 'X' will appear 48% of the way across the screen and 70% of the way down, and it will appear in Style 3.

Some notes about requiring correct responses:

1. The data file will reflect the response time to hit the correct key
2. The key code recorded will be that of the correct key
3. You can identify that an incorrect key was pressed first by looking in the 'Correct' column-those trials on which an incorrect key was pressed first will be labeled as 'False.'
4. The style you apply to the incorrect response prompt should usually have a min-max value of 0. This will inhibit DirectRT from prompting subjects to go faster after hitting the correct key. This might be desirable since they probably feel bad enough that they got the answer wrong.
5. The incorrect prompt will never be applied to any response that only has one valid response option (including **rt:any**) . This makes it easy to include instructions screens without the risk of an inappropriate prompt.

Key Release RTs

Location:

...\DirectRT\samples\5 other features\06 key release rts

Sample Description

In some paradigms, the subject is required to hold a key down prior to their response. When the time comes to react, the subject releases this key and responds as quickly as possible by hitting another key. This sample demonstrates such a task and shows how you can record the amount of time it took for the subject to release the first key **and** how long it then took to hit the response key.

Try running the sample. You will be asked to press down on the space bar whenever you see the words 'get ready' appear. You will then see a 3x2 grid appear that corresponds to the 6 keys: insert, home, pageup, delete, end, and pagedown. An X will appear in one of these cells and your task is to release the spacebar and hit the key that matches the location of the X as quickly as possible. You will then return your finger to the space bar for the next trial.

When you open up the data file, you'll see a column called 'Release'--this is the number of milliseconds it took you to release the spacebar from the time the image first appeared. The RT column is the reaction time to hit the correct (or incorrect) key which also starts from the time the image appears.

So how do we accomplish this? Take a look at the keyrelease.csv input file. You'll see that the only thing unusual in here is that in the Time column, you have 'RTR:' instead of 'RT:' for your trials. This essentially stands for 'ReactionTime with ReleaseTime'.

F	G	H	I	J	K
stim	loc	time	stim	loc	time
~get ready	0,0,1	2000	210.bmp	0,0,1	rtr:210,199,201,211,207,209
~get ready	0,0,1	2000	199.bmp	0,0,1	rtr:199,210,201,211,207,209
~get ready	0,0,1	2000	201.bmp	0,0,1	rtr:201,210,199,211,207,209

The previous stim is simply a text prompt so that subjects have time to press down on the spacebar (or any other key you instruct subjects to hold down). When you specify an RTR response, DirectRT will check to see if a key is being held down when the RT starts. If so, it will record the time it took for the subject to release that key and write it to the data file in a column labeled 'Release'. The response time for the subsequent response is independent of the release time (i.e., it begins at the onset of the stim and consequently includes the key release time).

Clear Screen During an RT

Location:

...\DirectRT\samples\5 other features\07 clear screen during an rt

Sample Description

Normally, when you request an RT, the stimulus remains on the screen until the response is received. In some cases, you may want to fix the time the stimulus is displayed regardless of how long it takes the subject to respond. You can do this using the 'clear' function. This instructs DirectRT to clear the screen after a specified number of milliseconds even if the subject has not yet responded.

Try running the sample contained in this folder. You will find that when you are required to respond, you will see the last stimulus for only 250ms, no matter how long you take to hit a key. Even though the stimulus has disappeared, the RT clock keeps going. This is accomplished with the input file as follows:

Q	R	S
stim	loc	time
\$pos	0,0,1	rt:44,53,clear250
\$neg	0,0,1	rt:53,44,clear250
\$pos	0,0,1	rt:44,53,clear250

After listing the valid keys, we enter 'clear250' to indicate that the screen should be cleared after 250ms. This has only a visual impact-it has no effect on the RT being collected.

You might wonder, 'why not just display the stim for 250ms and then take the RT?' The answer to this is, 'what if the subject can respond in less than 250ms?' The benefit of doing it this way is that it allows a response to occur both before **and** after the moment when the stim is cleared. If we were to display the stim for 250ms and **then** take the RT, we could only get responses greater than 250ms. The usefulness of this is even more pronounced when fixed display times become greater (e.g., clear500, clear1000, etc.).

Key Press Feedback

Location:

...\DirectRT\samples\5 other features\08 keypress feedback

Sample Description

Response Feedback allows you to display the response key that is hit as a form of visual feedback to subjects. You can request that the key name be displayed after an RT is taken by entering 'feedback' in a subsequent Stim column. Using the Location column you can specify where the feedback should appear and whether the screen is cleared or not before the feedback appears. Using the Time column you can specify how long the feedback should be displayed before continuing to the next stimulus or trial. For example, following a reaction time, if you enter a Stim value of 'feedback', a Loc value of '.5,.5,0' and a Time value of '2000', then the subject's response will appear in the middle of the screen for 2000ms. The style used to present the feedback will be the default style for the trial unless you specify an alternate style in angular brackets (e.g., 'feedback <2>' will present the feedback in style #2 from the experiment's style file).

Try running the input file contained in this sample folder. You will see that each time you answer an question, they key you pressed will appear on the screen. This is accomplished with the input file as follows:

I	J	K	L	M	N
stim	loc	time	stim	loc	time
#hobbies1	1,350,0	rt:30,48,46,32,18	feedback	375,85,0	2000
#hobbies2	1,350,0	rt:30,48,46,32,18	feedback	375,85,0	2000
#q1	1,350,0	rt:30,48,46,32,18	feedback	375,85,0	2000
#q2	1,350,0	rt:30,48,46,32,18	feedback	375,85,0	2000

On each trial, a question is asked that requires one of the keys, a though e, be pressed. After that, we request that response feedback be displayed for 2000ms by entering 'feedback' in the Stim column and 2000 in the Time column. The Loc column specifies where the feedback should be displayed (375 pixels to the right, and 85 pixels down) and that the screen should not be cleared (0) so that the feedback appears on the same screen as the question. A Loc value of '375,85,1' would have the same result with the exception that the feedback would appear on a fresh screen.

See also: Using open-ended responses as stimuli

Multiple RTs

Location:

...\DirectRT\samples\5 other features\09 multiple rts

Sample Description

DirectRT will record multiple RTs on a single trial if you request it. Take a look at the sample input file in this folder and you will see that each trial requests 8 RTs. The following excerpt shows two of the eight sequential RTs:

I	J	K	L	M	N
stim	loc	time	stim	loc	time
~Stan went to the movies.	0,0,1	rt:57	~jumlik	0,0,1	rt:44,53
~He had no money.	0,0,1	rt:57	~movie	0,0,1	rt:44,53
~The cashier laughed at him.	0,0,1	rt:57	~cuthet	0,0,1	rt:44,53
~He went home very sad.	0,0,1	rt:57	~phone	0,0,1	rt:44,53

After running this sample, take a look at the resulting data file. You will see that only one RT has been recorded for each trial-the final one. In order to view and analyze data from input files that assess **multiple** RTs, you will need to use the **log** data file, located in the 'Log' subfolder of the data folder. Open the log version of the data file and you will see the data for each of the eight RTs on every trial, e.g.,

F	G	H	I	J	K	L
Block	Trial	Order	Resp	Name	Stim	RT
1	1	2.01	57	SPACE	The lazy re	941
1	1	2.02	57	SPACE	Stan went	751
1	1	2.03	53	/	jumlik	10656
1	1	2.04	44	Z	lawn	1091
1	1	2.05	44	Z	orange	1733
1	1	2.06	53	/	kergh	1091
1	1	2.07	53	/	monji	1352
1	1	2.08	44	Z	dog	1112
1	2	3.01	57	SPACE	The lazy b	882
1	2	3.02	57	SPACE	He had no	871

The log file allows you to differentiate between the multiple RTs with the 'Order' column. Normally, the order column tells you the ordinal position in which the trial occurred in the session. In this sample, the trial labeled Block 1, Trial 1 was the second trial to occur (the instruction screen was the first but since it has a Block value of 0, it is not written to the data file). When recording multiple RTs, the order column also tells you which RT the data refer to. 2.01 refers to the first RT, 2.02 refers to the second, 2.03 to the third and so on up to 2.08 which refers to the eighth and final RT for that trial.

Using Previous Stim

Location:

...\DirectRT\samples\5 other features\10 using previous stim

Sample Description

In special cases you might want to instruct DirectRT to use a stimulus from a previous trial without knowing exactly what that stimulus will be when you design your input file. This could happen for example, when a stimulus is randomly determined on one trial but then you want to repeat that trial. On the later trial you can simply tell DirectRT which stimulus you want to repeat. Essentially saying, 'use the third stimulus from block 2, trial 5--whatever it was!' To do so, you can use the '?' operator which means to use a previous stimulus, no matter whether it was randomly chosen or not. To refer to a previous stimulus, you just need to know the block (b), trial (t) and stimulus (s) IDs. For example if you want to use block 1, trial 3, stimulus 4 in a later trial, you would request: '?b1t3s4' in the stim column.

The sample included in this folder illustrates this feature. Try running it. Notice the prime-target pairings used in the first 8 trials-the pairings are randomly chosen.

Then watch the next 8 trials. The order of the trials is randomized-but the pairings are the SAME as they were in the first 8 trials!

How was this accomplished?

In this sample, primes and targets for the first 8 trials are all randomly selected without replacement from a stimlist. That's nothing unusual if you've been looking through the many priming samples. What's new is that we want to repeat this block of trials again with the same prime target combinations. The graphic below shows the targets which are the 4th stimulus on each trial. On the first 8 trials, you can see that they are randomly selected from a stim list (i.e., \$pos and \$neg). The next 8 trials use the '?' operator to find out what the previous targets were and inserts them. For example, on the 9th trial, the Stim column reads '?b1t1s4' -this means show the 4th stimulus from Block 1, Trial 1. We don't show it here, but the prime on this trial reads '?b1t1s2' which means to show the 2nd stimulus from Block 1, Trial 1. This results in Block 2, Trial 1 being identical to Block 1, Trial 1. Using block-specific WGR values, each block is randomized separately. Voila! We're done.

Q	R	S
stim	loc	time
\$pos	0,0,1	rt:44,53
\$neg	0,0,1	rt:53,44
\$pos	0,0,1	rt:44,53
\$neg	0,0,1	rt:53,44
\$pos	0,0,1	rt:44,53
\$neg	0,0,1	rt:53,44
\$pos	0,0,1	rt:44,53
\$neg	0,0,1	rt:53,44
?b1t1s4	0,0,1	rt:44,53
?b1t2s4	0,0,1	rt:53,44
?b1t3s4	0,0,1	rt:44,53
?b1t4s4	0,0,1	rt:53,44
?b1t5s4	0,0,1	rt:44,53
?b1t6s4	0,0,1	rt:53,44
?b1t7s4	0,0,1	rt:44,53
?b1t8s4	0,0,1	rt:53,44

Play Sounds

Location:

...\DirectRT\samples\5 other features\11 play sounds

Sample Description

DirectRT allows you to play sound files and to synchronize them with other events and stimuli in your session. Try running the sound.csv sample in this folder. If your speakers are on (or your headphones are plugged in) and your volume is turned up, then you should hear the sound files play during the session. On most machines, you can adjust the volume by clicking on the little yellow speaker icon on the Windows task bar. If you do not have this icon, then you can type 'sndvol32' from the Run command on the Windows Start menu to adjust the volume.

Playing sound files in DirectRT is accomplished the same way every other stimulus is presented-by listing the sound file name in the Stim column. To tell DirectRT that it should be looking for a sound file, precede the filename with '!'

e.g., this is taken from the sound.csv sample:

I	J	K	L	M	N
stim	loc	time	stim	loc	time
!tone	0,0,0	1500	!dog	0,0,1	rt:30<97,3
!tone	0,0,0	1500	!giraffe	0,0,1	rt:30<97,3
!tone	0,0,0	1500	!monkey	0,0,1	rt:30<97,3
!tone	0,0,0	1500	!zebra	0,0,1	rt:30<97,3

On these trials, DirectRT will first look in the stim folder for a sound file called 'tone.wav'. It will play this file and from the moment it begins, 1500ms will elapse before DirectRT continues with the next stimulus. In this case, the next stimulus is another sound file (e.g., dog.wav, giraffe.wav). Note that the second sound file on each trial request an RT rather than a time interval in the Time column. When this happens, DirectRT will start timing the RT from the moment the sound file begins playing.

Note that you will usually enter '0,0,0' in the Loc column if you simply want to play the sound file without clearing the screen, and '0,0,1' if you want the screen cleared when the sound begins. DirectRT has advanced sound functions though which you can control by entering values different from these in the Loc column.

Advanced Sound Functions

By using special values in the Loc column (i.e., instead of 0,0,0 or 0,0,1), you can control three aspects of the how the sound files are played: Frequency, Pan and Volume. Try running the dichotic.csv sample. To get the full effect, try to use stereo speakers--or even better, a pair of head phones. You'll notice that certain sounds play in the left channel and some in the right. On the last two trials, the sounds are either sped up or slowed down. How are these effects accomplished? Here is an excerpt from the input file:

I	J	K	L	M	
stim	loc	time	stim	loc	time
lgiraffe	0,-100,1	0	ldog	0,100,0	rt:30<
laustralia	0,-100,1	0	ltelevision	0,100,0	rt:30<
lzebra	0,-100,1	100	lmonkey	0,100,0	rt:30<
lmaryland	0,-100,1	100	lauto	0,100,0	rt:30<
lmonkey	0,-100,1	200	lzebra	0,100,0	rt:30<
ltelevision	0,-100,1	200	laustralia	0,100,0	rt:30<
lauto	55000,-100,1	400	ldog	55000,100,0	rt:30<
ldog	33000,-100,1	400	lgiraffe	33000,100,0	rt:30<

To speed up or slow down the sound file, you can set the frequency in hertz (Hz) at which the sound is played by entering a value from 100 to 100000. In the first six trials, this value is set to 0-which indicates to use the default frequency of the sound file. However, the seventh trial, the frequency of the sounds are set to 55000Hz-significantly higher than the default frequency of 44000Hz. In contrast, on the eighth trial, the frequency of the sounds are set to 33000Hz-significantly lower than the default frequency.

To control whether a given sound is dominant in the right or left channel, you can set a Pan value anywhere from -100 (left only) to 100 (right only). Note that the sounds in the first stim column have a pan value of -100. Consequently, they will play in the left channel only. In contrast, the sound files in the second stim column have a pan value of +100-meaning they will play only in the right channel. Other examples: A value of 50 would result in the right channel being at full volume and the left at half volume. A value of -75 would result in the left channel being at full volume and the right channel being at 25% volume.

Finally, to control the overall volume at which the sound file is played, you can enter a value from 0 (normal) to -100 (completely muted). Sounds can only be reduced in volume and not amplified.

Note that in this sample an added effect of delay is achieved by entering variable values in the Time column of the first sound. When this value of 0, the two sounds play nearly simultaneously. When the value is 100, there is a 100ms delay between the sounds. When it's 200, there is a 200ms delay and so on. This occurs because the value in the Time column for a sound determines how many milliseconds DirectRT will let pass before continuing with the next stimulus.

Voice Responses

Location:

...\DirectRT\samples\5 other features\12 voice responses

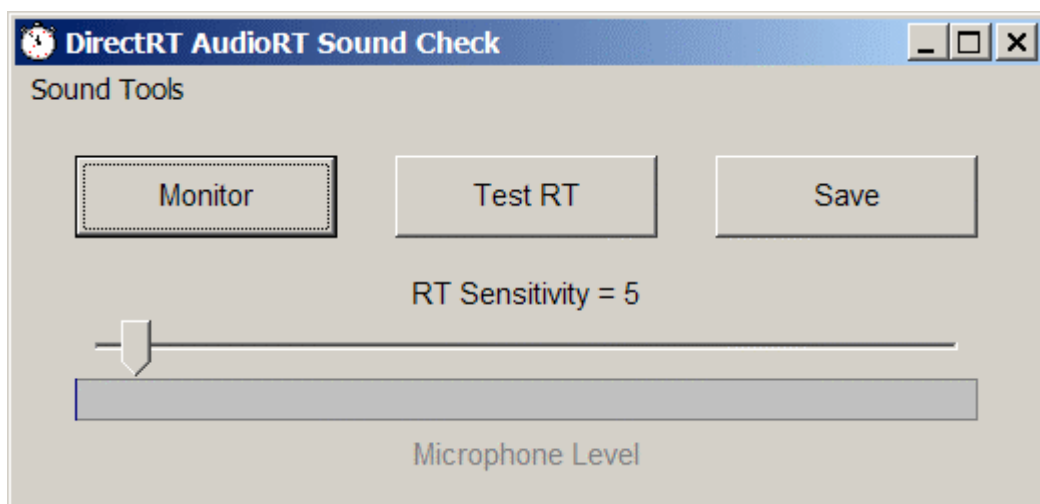
Sample Description

These samples illustrate how to assess reaction times through a microphone connected to your sound card and how to optionally record voice responses as sound files on your hard drive.

Collecting Voice Responses

With all of the voice response methods, you will need to make sure you have a microphone connected to the microphone input connector on your sound card. You will also need to make sure that the microphone is active and able to record. Finally, you will need to tell DirectRT how sensitive it should be in detecting response times.

To make all this easier, DirectRT comes with a tool called SoundCheck, which can be located on the Tools menu of DirectRT. When you run SoundCheck, this is what you'll see:



After making sure your microphone is plugged into your soundcard, click 'Monitor' and speak into the microphone. If all is well, you should see the bar labeled 'Microphone Level' respond immediately to your voice. If that works, then click 'Stop' to end the monitoring. Now try a sample voice rt. Click 'Test RT' and then speak into the microphone. SoundCheck should report the response time in milliseconds.

If you need to speak too loudly to set off the test, then try moving the sensitivity slider toward the left. If the test is set off too easily, then try moving the sensitivity slider toward the right. Try this until you find an RT sensitivity that works well for you. When you are happy with the sensitivity, click 'Save.' This will make the new sensitivity level available to DirectRT when you run a session with voice responses.

It is advisable to run a few test RTs on any machine before running a DirectRT session with voice responses for the first time. When running the samples in this folder, if you repeatedly get the message 'please wait for the stimulus' this means the RT sensitivity value is set too low. Alternatively, if you repeatedly get the message 'please try to respond faster' then the RT sensitivity value is set too high.

Note that the SoundCheck program provides access to other sound utilities by clicking on 'SoundTools' in the top left corner. Access to these additional tools is provided here in case you have problems with your test RTs.

Voice Response Methods

Automatic Continuation Method

The easiest to implement of all the voice response methods, this traditional method instructs DirectRT to simply wait until a voice response occurs and then to immediately continue with the next stimulus or trial. This is illustrated in the sample 'stroop automatic.csv'

stim	loc	time
~blue	0,0,1	rt:aud
~green	0,0,1	rt:aud
~green	0,0,1	rt:aud

Notice that for a voice RT, all we need to do is enter 'rt:aud' in the time column. In place of keycodes, we tell DirectRT to wait for auditory input from the sound card.

Fixed Time Method

With the fixed time method, you specify the amount of time you want DirectRT to monitor the microphone. This time period can range from 100ms up to much longer periods (e.g., 5 minutes or 300000ms). You can also request that DirectRT save the incoming sound as a .wav sound file in a folder so you can listen to it later.

Take a look at the sample, 'stroop fixed.csv' in this folder. We see that on each trial, a voice response is requested by entering 'rt:aud' in the Stim column, followed by the fixed amount of time we want DirectRT to monitor the microphone. In this case, we ask DirectRT to monitor the microphone for 1500ms. In this sample, we also ask DirectRT to save the incoming sound on each trial as a sound file by inserting 'rec' (short for 'record') which could optionally be omitted.

stim	loc	time
~blue	0,0,1	rt:aud,rec,1500
~green	0,0,1	rt:aud,rec,1500
~green	0,0,1	rt:aud,rec,1500

When the response is set to begin, DirectRT begins monitoring the incoming sound and keeps doing so until the time limit you set has expired. When the time limit expires, DirectRT will continue with next stimulus or trial. If you requested that DirectRT save the sound (using 'rec'), then it will be saved to the data folder so that you can listen to it later for coding, response verification, or some other purpose.

Combo Method

A third option combines the first two. In this method, DirectRT will wait until a voice response has occurred and then-from that point-will wait an additional period of time that you specify. For example, if you want DirectRT to continue 500ms after each response then you can specify this the same way as with the Fixed Time method, but with a *negative* time value. The negative time value tells DirectRT that you want the Combo Method rather than the Fixed Time method. This is illustrated in the sample 'stroop combo.csv'

stim	loc	time
~blue	0,0,1	rt:aud,rec,-500
~green	0,0,1	rt:aud,rec,-500
~green	0,0,1	rt:aud,rec,-500

This method is especially useful if you want to proceed immediately after subjects respond, but you also want to save subjects' verbal responses as sound files. By specifying an additional fixed time, DirectRT will continue recording just long enough to capture the verbal response (otherwise, you would record only the first few milliseconds!). Larger negative values will cause DirectRT to wait longer after the initial sound before continuing.

See also: MinMax Values (which apply as usual just like any other RT)

TTL Signals

Location:

...\DirectRT\samples\5 other features\13 ttl signals

Sample Description

In order to communicate with external equipment (e.g., physiological equipment, response boxes, other computers), you can send and receive TTL (Transistor-Transistor Logic) signals through any of your computer's accessible ports (e.g., parallel or serial ports). In English, this means you can ask DirectRT to send a value from 0-255 to your external equipment any time you like, and/or you can have DirectRT wait for any value from 0-255 to come from the external equipment. To accomplish sending a value, you list a TTL signal just like any other stimulus in the Stim column of your input file. To receive a value, you list a TTL signal just like any other RT. Consequently, you can synchronize the TTL signals with other stimuli in your experiment and use TTL signals as way to record precise RTs from external equipment.

In jargonese, a TTL signal is an 8-bit value (e.g., 00101001) represented in decimal form as a value from 0-255. You should check the documentation that came with your external equipment to see what values it can send and receive. Since 8-bit values can be represented either as their individual bit values (e.g., 00101001) or as their decimal equivalent (e.g., 41), you may need to convert between them (since they're simpler to comprehend, DirectRT uses decimal values exclusively). You can use the Windows calculator program to do this quite easily. Type 'calc' from the Run command on the Start menu. From the view menu, click 'Scientific.' To convert from Binary to Decimal, click 'Bin' and type in the bit value (e.g., 00101001) and then click 'Dec'. You will see the decimal equivalent appear (e.g., 41). To convert from Decimal to Binary, click 'Dec' and type in the decimal value (e.g., 41) and then click 'Bin'. You will see the binary equivalent appear (e.g., 00101001). For future reference you can also use this method to convert to and from Hexadecimal format as well.

Ports

So far, we know that DirectRT can exchange values from 0-255 with your external equipment. The second thing we need to know is where that exchange will take place. This meeting place is going to be a 'port' located on the same computer as DirectRT. A port is a place where values can be written and read. Ports that are accessible to outside equipment include the serial and parallel ports. You will need to know which port your external equipment connects to, and the port address. With this information, you will be set to go. For example, you will be able to tell DirectRT to 'send the value 155 to port 330' or 'go to port 330 and wait for the values 155, 156, or 157.'

Initial Testing with your External Equipment

DirectRT comes with a tool called IOTest located in the DirectRT\Tools folder. This program uses the same code as DirectRT to send and receive byte values from specified ports, but is simpler to work with during initial testing. If you can get your external equipment to work properly with IOTest, then it should work well with DirectRT. Try running IOTest and follow these directions:

When IOTest starts, you will see a pull down list of ports-these will include port addresses that IOTest immediately recognizes as being available for read/write access. You can choose one of these, or you can enter a port address of your choosing. Note that IOTest as well as DirectRT will expect a port address in Decimal form. Since ports addresses are often referred to in Hexadecimal format, you may need to convert a hexadecimal address to decimal. You can use the Windows program calc.exe to do this (see above).

After selecting a port address, click the 'Read' button next to it. Below where it says 'Monitor' you can watch the value of this port. For example, if you select port 888 (often the address for parallel port LPT1), you can see its current value. If the value at this port address changes, you will see it change here. You can test to see if your external equipment can send values to this port address. Try it and see. If successful, you will see the value change to whatever value you send from your external equipment.

You can also test to see if DirectRT can send values to your external equipment. In the text box where it says 'Byte Value (0-255)' try entering a value, and then click the 'Send Byte Now' button. You should see the value update on the monitor, and you should see your external equipment react accordingly if it is able to read the data from that port address successfully. You can also send a byte value for a specified number of milliseconds by entering a value in the 'Duration' field. Try entering a new byte value (e.g., 123) and a value of 100 in the Duration field. Then click 'Send Byte Now.' You should see the monitor briefly display the new value and then return to its original value.

That's it. If the testing was successful, then you should be able to proceed with the samples.

Samples

Samples were not finalized at the time of writing this manual. Contact support@empirisoft.com to obtain the current TTL samples.

Implicit Attitude Test (IAT)

Location:

...\DirectRT\samples\5 other features\14 iat

Sample Description

"I've tried the DirectRT implementation of the IAT. It worked like a charm and has an appearance very similar to the lab procedures we've been using for the last few years."

Anthony Greenwald
Co-Creator of the IAT
University of Washington

The IAT sample was designed as a template for conducting IAT sessions with a variety of stimuli. Since the typical design of an IAT is rather complex, you can use this sample as a reference when building your own IAT, or you can simply copy it and substitute your own stimuli and scales—and voila! A nearly instant IAT!

If you're new to DirectRT I suggest that you run through the tutorials before attempting to follow the IAT input file (this is kind of advanced stuff).

Some things to notice:

The sample contains 5 blocks of trials. In this sample, we are only interested in data from blocks 3 and 5—consequently, we use a block ID of 0 for the other blocks so that they will not be written to the data file.

We use an optional comment column (!Step) to identify which block (or Step) the trials belong to.

The Bgr column contains all 0's because the order of the blocks is not randomized. The Wgr column contains identical values for all items within blocks 1, 2 and 4 so that the items within each of those blocks will be randomized. In blocks 3 and 5 alternating patterns of Wgr values allow us to alternate between randomly ordered adjectives and targets. Note that the instruction screens always have a Wgr value of 0 so that they will stay where they are.

In the time column, some trials specify 'rt:23,18' and others specify 'rt:18,23'. This tells DirectRT which response is correct as the correct response always comes first.

In this sample, we have:

Adjective1 = Good, Adjective2 = Bad
Target1 = White, Target2 = Black

These could be switched with whatever you like.

The first Stim column contains the scale that should appear on screen. On the first trial, the scale is 'a2-a1.' Take a look at the image file in the stim folder called 'a2-a1.bmp.' It's a label that reads 'bad - good'. That's because a2-a1 means adjective2 (bad) on the left and adjective1 (good) on the right.

Later the scales get more complicated because you have to have an adjective and a target on each side. E.g., look at 'a2t1-a1t2.bmp.' You'll see adjective2 (bad) and target1 (white) on the left and adjective1 (good) and target2 (black) on the right.

Do you see how this could act as a template? You wouldn't have to change the input file—just the stimuli. If your targets were 'me' and 'other', instead of 'white' and 'black', you could just replace 'white' with 'me' and 'black' with 'other' on each of the scale graphics. Does that make sense?

The same goes for the second stim column. In the column you see that a random stimulus is drawn from the text files 'adjective1.txt', 'adjective2.txt', 'target1.txt' and 'target2.txt'. Take a look at the adjective1.txt file. Since adjective1 = 'good,' the stimuli in this stim list are words like 'JOY', 'LOVE' and 'PEACE.' You could simply replace these with words representing whatever you're using as adjective1.

A couple final notes, you may want to shorten the trial interval in this experiment to make the 'flashing' between trials less pronounced. To do this, click on 'Options' in the style file, IAT.drt. Remember though that DirectRT prepares the next trial during the trial interval, so be sure to give it at least 50-100ms.

The scales in this study were made with a graphics programs called Fireworks by Macromedia. It's a great program for putting together formatted text and saving it as .bmp graphics that you can use in DirectRT. A free trial version is available from <http://www.macromedia.com/software/fireworks/trial/>. If you get it, you can just modify the source files for this sample which are located in the Fireworks subfolder.

Advanced Graphics

Location:

...\DirectRT\samples\5 other features\15 advanced graphics

Sample Description

For the truly masochistic and/or the individual who feels that DirectRT is just too darn easy, here is a sample that will surely challenge you. Put together with Geoffrey Cohen and Mitch Prinstein at Yale, this study simulates a group interaction study over the web. Uses just about every graphics trick that DirectRT has to offer. Too much to explain here. If you see something you like and can't figure it out, feel free to email me and I'll be happy to elaborate. –Blair

Note: In your display settings, you should select 'advanced' and then 'large fonts' in order for this sample to look exactly as it should.

Key Code Quick Reference (see also 'Time')

Key	Code	Key	Code	Key	Code	Key	Code	Key	Code	Key	Code
1	1	Pad1	79	F1	59	A	30	N	49	Enter	28
2	2	Pad2	80	F2	60	B	48	O	24	Space	57
3	3	Pad3	81	F3	61	C	46	P	25	LShift	42
4	4	Pad4	75	F4	62	D	32	Q	16	ICtrl	29
5	5	Pad5	76	F5	63	E	18	R	19	lAlt	56
6	6	Pad6	77	F6	64	F	33	S	31	rShift	54
7	7	Pad7	71	F7	65	G	34	T	20	rCtrl	157
8	8	Pad8	72	F8	66	H	35	U	22	rAlt	184
9	9	Pad9	73	F9	67	I	23	V	47	;	39
0	10	Pad0	82	F10	68	J	36	W	17	'	40
				F11	87	K	37	X	45	,	51
				F12	88	L	38	Y	21	.	52
						M	50	Z	44	/	53

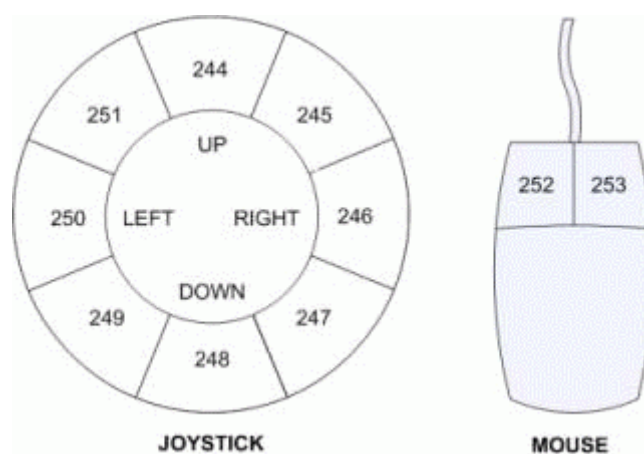
Key	Code	Key	Code	Key	Code
-	12	Home	199	PgDn	209
[	26	Insert	210	PgUp	201
]	27	Left	203	Right	205
`	41	lWin	219	rWin	220
=	13	NumLock	69	ScrollLock	70
Apps	221	PadCom	91	SysRq	183
Backspace	14	PadDec	83	Tab	15
CapsLock	58	PadDivide	181	Up	200
Delete	211	PadEnter	156		
Down	208	PadMinus	74		
End	207	PadMult	55		
Esc	---	Pause	116		

In addition to the specific key codes, other allowable keypress events are:

- **any** = any key
- **num** = any number key on the regular keyboard
- **numpad** = any number on the number pad
- **alpha** = any character of the 26-character alphabet
- **funct** = any function key

Joystick & Mouse Codes

Joystick	Code	Mouse	Code
Button 1	241	Left-Click	252
Button 2	242	Right-Click	253
Button 3	243	Mid-Click	254
Up	244	Other-Click	255
Up-Right	245		
Right	246		
Down-Right	247		
Down	248		
Down-Left	249		
Left	250		
Up-Left	251		



Note: When assessing response accuracy, the first key code listed defines the 'correct' response. If pressed, the 'Correct' column in the data file will read 'true'.

See 'Time' for more detail

Index

!	23
#	27
\$	27
%	27
&	23
?	23
~	27
=	23

A

Alignment	22, 32, 48
-----------	------------

B

Block	17
BMP	23, 29

C

Center Stimuli	32
Clear Screen	32, 35, 81
Codes	8, 36, 37, 38, 39, 96
Colour	22, 29, 45, 46
Command Prompt	53
Compound-Screen	35
Continuous Joystick	36, 39
Correct Responses	35, 79
Creating Graphics	30
CSV	7, 8, 10, 11, 16, 17

D

Data	9, 11, 55, 56, 57, 58
Directx	7
DRT Files	9, 22
Duration of Stimulus	35

E

Excel	7, 16, 17
External Equipment	28

F	
Features.....	4, 6
Feedback.....	48
Feedback.....	36, 39, 82, 83
File Format	16
Font.....	22, 47
Frequency	25, 34
G	
GIF.....	29
H	
Horizontal.....	32
I	
IAT	93
Image.....	23, 28, 29
Input	35
Input Files	9, 10, 11, 16, 52, 53
Introduction and Key Concepts.....	9
J	
Joystick.....	35, 36, 38, 76, 77, 96
Joystick Continuous	77
JPG.....	29
Jumping	41
K	
Key Release Times.....	35, 80
Keycodes.....	35
Keypress	35, 36, 39, 96
L	
Loc	10, 34
Log Data File.....	55
M	
Main Menu Commands	14
Manual for Printing	3
Maximum RT.....	48
Medialab	53, 54

Merge Data	15, 55
Messages	48, 49
Microphone	35
Minimum RT	48
Minmax	48
Mouse	76, 77
Mouseclick	35
Multiple Response Times	36, 39, 83
N	
Name (Styles)	45
O	
Open Ended Responses	35, 74
Optional Fields	18, 19, 20
Options	34
P	
Pan	25, 34
Physiological Equipment	28, 91
Pixels	32, 34
Port	34
Powerpoint	30
Printscreen	31
Prior Response	23, 27
Prior Stimulus	23, 84
R	
Random	26
Randomization	20, 21, 40
Randomization Between Groups	20
Randomization Within Groups	21
RBG	20
Refresh Rate	9, 15
Resolution	32, 50
Response Boxes	28
Running a Session	52
RWG	21

S

Samples	59, 60
Scales	31
Screen Size	50
Screens	10
Security	7, 8
Sequential	26
Simultaneous Stimuli	35
Skipping	39
Sound	86, 87, 88
Sound	23, 25, 28, 34, 35
Status Bar	15
Stim	10, 23, 25, 26, 28
Stimlist	25, 26, 27, 28
Stimulus	23, 24, 25, 26, 27, 28
Stimulus Location	32
Styles	11, 22, 42, 44
System requirements	7
System Settings	8

T

Text	22, 23, 24, 25, 27, 28
Text File	23, 24
Time	9, 10, 11, 35, 36, 39
Trial	18
Truerandom	26
TTL	28, 34, 35, 36, 39, 91

U

Unconditional Skipping	40
------------------------------	----

V

Vertical	32
Voice Response	15, 36, 39, 88, 89, 90
Volume	25, 34