

Easy IAT: An Open-Source and Customizable Implicit Association Test

Easy IAT Manual

Chanita Intawan and Graham William Thompson

Last Updated: July 2015

Abstract

The Easy IAT is an open-source implementation of the implicit association test (IAT; Greenwald, McGhee, and Schwartz 1998) that allows researchers to easily build and run custom IATs online. It is meant to be a free alternative to proprietary programs such as DirectRT or Inquisit by Millisecond Software. The Easy IAT program is written in standard html5 and javascript and can be run from any server without additional dependencies. These technologies have been shown to produce consistent and reasonably accurate response times for online behavioral tasks across platforms (Reimers and Stewart 2014). Easy IAT provides an easy, simple setup for all types of IAT including the traditional IAT and single target IAT and accepts any text and all image file types for stimuli. Python scripts for calculating D-scores are provided for all standard IAT types. Additionally, powerful configuration files allow researchers to easily create and deploy custom IATs with any number of blocks, trials, and stimuli. The standard interface is modern, sleek looking and scales to window size to provide a consistent experience for participants. Easy IAT can also be easily embedded into larger experiments or websites and provides unique participant identifiers to easily match participant IAT data to data collected from other surveys or experimental applications. Full access and rights are given to the source code to allow interested researchers to modify the application as much as they may need. Easy IAT provides a simple method to run implicit association tasks online or locally and is more flexible and modern than current alternatives.

Table of Contents:

1. IAT Structure.....	3
2. Easy IAT Requirements.....	5
3. Easy IAT Setup.....	6
4. Creating Your Own IATS.....	7
5. Data Collection.....	9
6. Data Output.....	10

1. IAT Structure

The implicit association test is a quasi-experimental procedure to measure the strength of implicit associations between paired concepts. For example, the classic race IAT pairs GOOD-BAD with WHITE-BLACK and uses the reaction times in categorizing stimuli with these concepts to produce a D-Score which expresses the relative association of those concepts. To implement the IAT, it is important to understand its underlying structure. We suggest these three research papers for a more in-depth review of the IAT's mechanics. To try it out first hand many sites such as Project Implicit have active versions online.

Greenwald, Nosek, and Banaji 2003

Nosek, Greenwald, and Banaji 2005

Lane, Banaji, Nosek, and Greenwald 2007

A brief overview of the IAT is given below.

Blocks & Trials

The IAT is composed of blocks, which are then composed of trials. The standard IAT (Greenwald, McGhee, and Schwartz 1998) has 7 blocks. There are different types of IATs that may have a different number of blocks. For example, the ST-IAT (single target IAT; Wigboldus, Holland, and van Knippenberg 2004) has 5 blocks. There are a specific number of trials in each block. In the Easy IAT's standard configuration, practice blocks contain 20 trials and non-practice blocks contain 40 trials.

Categories & Stimuli

During each trial, there are concept categories displayed at the top corners of the screen and a stimulus to be categorized that is presented in the middle of the screen. The stimulus is randomly selected from a set of stimuli. Participants must categorize the stimulus into the correct category as given by the instructions. Participants use the "e" and "i" keys on the keyboard to categorize the stimulus into the left and right categories. If the stimulus is correctly categorized, the next trial begins. If the stimulus is incorrectly categorized, a red X will appear, and the participant must press the other key that correctly categorizes the stimulus to move on to the next trial. The particular block number displays a particular set of categories, which are described in detail below.

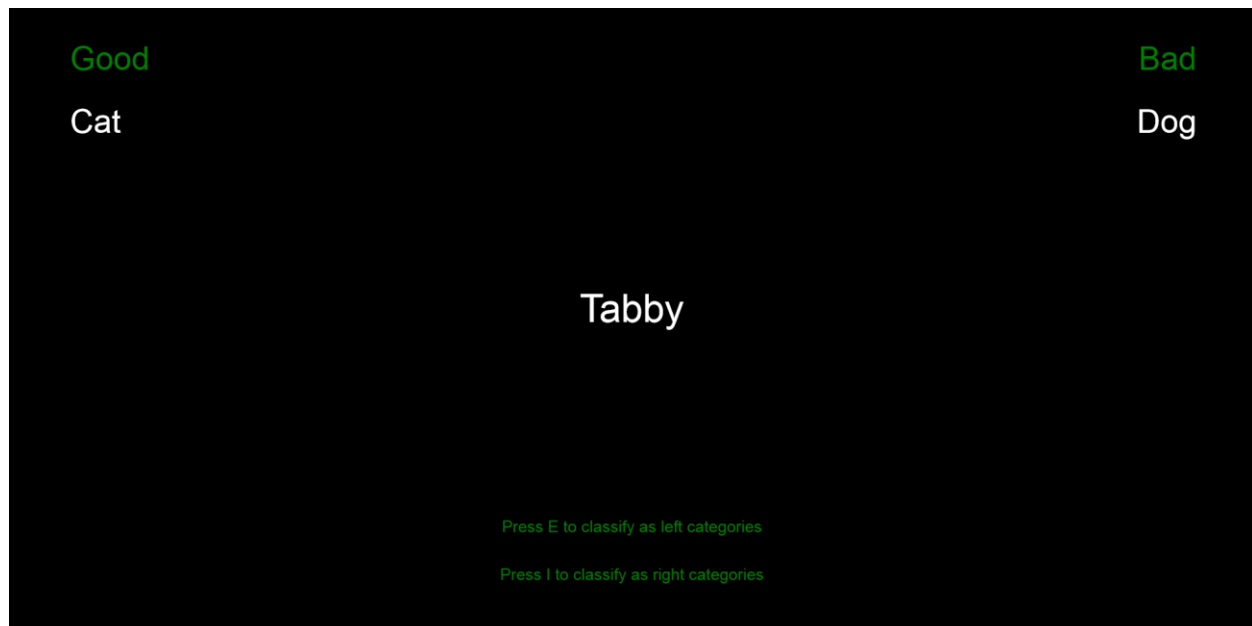
Block Structure

In the first block, participants categorize stimuli for the attributes. In the example, the attributes are "good" and "bad." In the second block, participants categorize stimuli for the target concepts. In the example, the target concepts are "cat" and "dog." In the third block, the attributes and target concepts are paired. In the example, "good" is paired with "cat", while "bad" is paired with "dog." The fourth block is a longer block with double the number of trials as the third block. In the fifth block, the pairings are switched. In the example, "cat" has switched positions with "dog." In the sixth block, the attributes and switched concepts are paired, so that "good" is paired with "dog" and "bad" is paired with "cat." The seventh block is a longer block with double the number of trials as the sixth block. Participants must categorize the stimuli rapidly and accurately as possible. In the Easy IAT, there is a built-in maximum

response window of 5 seconds (5,000 milliseconds) per trial. If the participant takes longer than 5 seconds to respond, a pop-up screen will appear that instructs the participant to respond faster. For each trial, the reaction times are recorded in milliseconds.

<u>Block #</u>	<u># of Trials</u>	<u>Categories</u>	<u>Examples</u>
Block 1	20 trials	Attributes	Good/Bad
Block 2	20 trials	Target Concepts	Cat/Dog
Block 3	20 trials	Attributes + Target Concepts	Good/Cat + Bad/Dog
Block 4	40 trials	Attributes + Target Concepts	Good/Cat + Bad/Dog
Block 5	20 trials	Target Concepts Switched	Dog/Cat
Block 6	20 trials	Attributes + Target Concepts-Switched	Good/Dog + Bad/Cat
Block 7	40 trials	Attributes + Target Concepts-Switched	Good/Dog + Bad/Cat

Example Screen:



2. Easy IAT Requirements

The Easy IAT is server based and must be hosted using server software either on a web facing machine or on a local machine for local use. There are many commercial options for servers, which charge a yearly subscription fee. Researchers at universities may have access to university servers at minimal or no cost.

To be able modify the files on your server, you will need an FTP (file transfer protocol) client. There are many options for FTP clients. WinSCP is a recommended open source FTP client for Windows. This can be downloaded for free at: <https://winscp.net/eng/download.php>

To modify the contents of the configuration files, you will need a text editor. Notepad or other default editors will work, but to avoid formatting issues we recommend a dedicated code editor such as Notepad++ or TextWrangler. These can be downloaded for free at: <https://notepad-plus-plus.org/> and <http://www.barebones.com/products/textwrangler/download.html>

On your server you will need PHP, Python, and write permissions for web user in directory. Most server software will have PHP and Python installed by default so you will only need to modify permissions. For client-side (your participants' browser and computer), the requirement is a recent HTML5 and Javascript capable web browser. We have tested Easy IAT on different browsers. Mozilla Firefox and Google Chrome work well with the Easy IAT. Opera and Apple Safari have reported issues, so it is strongly recommended that you do not allow participants to use these browsers for running the IAT. Inform participants that they can only use a Firefox or Chrome browser.

Hardware requirements are a physical keyboard and reasonably sized screen. While the IAT will run on any browser capable device, it is recommended to only use standard computers not smartphones, tablets without keyboards, or other smart devices. It is unknown whether virtual keyboards result in different latencies than physical keyboards for the IAT task.

3. Easy IAT Setup

1) Copy the unpacked directory from Github onto your server. Each different IAT will need to have its own folder.

IMPORTANT: In the contents of the folder, you will need to set write permissions for the savefile.php and the output folder. This ensures that the data actually outputs.

2) In the config folder, you will need to open the main iat_config.cfg file and change the local server path to match your own server. This path is where your main IAT folder is located.

3) Set write permissions for the IAT. Navigate to the main directory of your IAT and right click the savefile.php file and output folder. Using the properties menu, check write and execute permissions for owner, group, and others.

4) Customize your IAT stimuli and experiment. Navigate to the config folder and use your text editor to modify the global configuration file (iat_config.cfg) and the block configuration files (blockConfig.cfg) as needed.

There are many different options for customization. For basic IATs, you will only need to modify the contents in the config folder and the images folder if you are using picture stimuli.

For advanced users, you can modify the structure of the IAT itself. This is only recommended if you have a solid understanding of the IAT and Javascript coding. To modify the structure of the IAT, you can modify the iat_main.js code.

4. Creating Your Own IATs

The structure and content of your IAT is determined by the parameters in your configuration files.

- The main config file (iat_config.cfg) contains the global parameters that set the number of blocks and the location of their config files, time between trials and the text for the instruction and debriefing screens.
- The block configuration files (blockConfig.cfg) contain the categories for that block, the stimuli to be categorized, the number of trials to run, and the text for instructions.

To add your own categories and stimuli, open up each block folder, and look for the text labeled “categories” and “targets.” For categories, “Position1” and “Position2” are the attribute categories. In the example IAT, the attribute categories are “good” and “bad.” “Target1” and “Target 2” are the target categories. In the example IAT, the target categories are “dog” and “cat.”

The stimuli to be categorized are listed in the targets1 and targets2 arrays. For a standard IAT both are used to contain the attribute set and target set of stimuli respectively. For a single target IAT only the first is used as you do not need to alternate between attribute and target stimuli. The list of possible stimuli and their settings are given by a line of text in the array.

Here is the template for “targets1” and “targets2.”

```
“txt    word          correct_category”
```

```
“img    image_link    correct_category”
```

The correct category can be either the attribute category or the target category. To use the attribute category, change the correct_category to be the label associated with your attribute category. For example

```
"categories":{"Position1":"Good", "Position2":"Bad", "Target1":"Cat", "Target2":"Dog"}
```

```
"targets1": ["txt dreadful Bad",
```

```
"txt horrible Bad",
```

```
"txt terrible Bad",
```

```
"txt awful Bad",
```

```
"txt unpleasant Bad",
```

```
"txt marvelous Good",
```

```
"txt superb Good",
```

```
"txt fantastic Good",
```

```
"txt glorious Good",
```

```
"txt wonderful Good"
```

```
]
```

To use the target category, change the correct_category to be either “Target1” or “Target2.”

```
"categories":{"Position1":" ", "Position2":" ", "Target1":"Cat", "Target2":"Dog"}

"targets1": ["txt Feline Target1",
"txt Tabby Target1",
"txt Siamese Target1",
"txt Calico Target1",
"txt Poodle Target2",
"txt Labrador Target2",
"txt Terrier Target2",
"txt Beagle Target2"
]
```

For image stimuli, you will need to include the image link from the image folder. If you are using images, they must be present in the image folder and have unique file names. Change the first identifier to “img” and substitute the file location for the stimuli name.

```
"categories":{"Position1":" ", "Position2":" ", "Target1":"Cat", "Target2":"Dog"}

"targets1": ["img http://yourserverhere.com/IAT/images/cat1.jpg Target1",
"img http://yourserverhere.com/IAT/images/cat2.jpg Target1",
"img http://yourserverhere.com/IAT/images/cat3.jpg Target1",
"img http://yourserverhere.com/IAT/images/cat4.jpg Target1",
"img http://yourserverhere.com/IAT/images/dog1.jpg Target2",
"img http://yourserverhere.com/IAT/images/dog2.jpg Target2",
"img http://yourserverhere.com/IAT/images/dog3.jpg Target2 ",
"img http://yourserverhere.com/IAT/images/dog4.jpg Target2"
]
```

There are other customization options that are discussed below. These do not need to be changed unless you are making a specialized IAT. The “alternating” option indicates whether the attribute and target stimuli need to alternate. You can change this to “yes” or “no.” The “numberOfTrials” option indicates the number of trials. You can change the number to a custom number of trials. The “flipTargetCategories” option indicates whether the categories need to change sides from the previous block. You can change this to “yes” or “no.” The “trialInstructions” are the instructions displayed in each trial. The “blockEndInstructions” are the instructions displayed at the end of the block, which are actually instructions for the next block. You can change the text instructions.

For advanced users, you can customize the number of blocks in an IAT. You will need to add or delete block folders. In addition, the block numbers need to match the “blockDirs” in the iat_config.cfg file. For example, if you have 5 block folders, then the “blockDirs” needs to have 5 blocks listed.

5. Data Collection

It is important to thoroughly test the IATs after you have created them. It is also essential that you make sure data files output correctly. To begin data collection, you will need to make the IAT link available to participants. There are a variety of ways to do this. For example, you can embed the link in an online survey. Data collection can be done locally or online. Whether the data collection is in a lab or through the internet, the participant accesses the IAT through the web. When participants click the link, they are first shown the instruction page and then proceed to do the IAT task. When the participant is finished, the participant is given a randomly generated 4 digit verification code. This code is useful for combining IAT data with external survey data.

Note: Since the code is randomly generated, there is potential for repeats especially with large samples.

For advanced users, the verification code can be customized. The code can be modified in the `iat_main.js` file.

6. Data Output

The IAT data is saved in the output folder. Each text (.txt) file is for each participant. The name of the text (.txt) file has the verification code identifier followed by a randomly generated number as a time stamp. When you open the (.txt) file for a single participant, each piece of information is comma separated. Each row represents information for one particular trial.

Column 1: Block number (e.g. block1)

Column 2 and 3: Attribute category 1 & 2 (called position1 and position2 in the blockconfig.cfg file)

Column 4 and 5: Target category 1 & 2 (called target1 and target2 in the blockconfig.cfg file)

Column 6: Initial randomization of the categories, 0 is the original position and 1 is the reversed position

Column 7: Type of stimuli (text or image), the stimulus that is displayed for that trial and the correct category for that stimulus.

Column 8: Reaction time in milliseconds

There is a Python script included in the output folder that loops through all the files in the directory and converts the reaction times into D-scores. You can create your own script for converting the reaction times into D-scores. An article written by Greenwald, Nosek, and Banaji (2003) explains how to calculate D-scores. The D-score output file is called SummaryScores.out. In the SummaryScores.out file, each row with the exception of the header represents data for a single participant. The first column is the verification code identifier, the second column is the original file name, and the third column is the D-score. The summary scores can then be used in an analysis program.

Happy researching! 😊 -Chanita & Graham

References

- Greenwald, Anthony G., Debbie E. McGhee, and Jason L.K. Schwartz. 1998. "Measuring Individual Differences in Implicit Cognition: The Implicit Association Test." *Journal of Personality and Social Psychology* 74(6): 1464-1480.
- Greenwald, Anthony G, Brian A. Nosek, and Mahzarin R. Banaji. 2003. "Understanding and Using the Implicit Association Test: I. An Improved Scoring Algorithm." *Journal of Personality and Social Psychology* 85: 197-216.
- Lane, Kristin A., Mahzarin R. Banaji, Brian A. Nosek, and Anthony G. Greenwald. 2007. "Understanding and Using the Implicit Association Test: IV." *Implicit Measures of Attitudes*: 59-102.
- Nosek, Brian A., Anthony G. Greenwald, and Mahzarin R. Banaji. 2005. "Understanding and Using the Implicit Association Test: II. Method Variables and Construct Validity." *Personality and Social Psychology Bulletin* 31(2): 166-180.
- Reimers, Stian, and Neil Stewart. 2014. "Presentation and Response Timing Accuracy in Adobe Flash and HTML5/JavaScript Web Experiments." *Behavior Research Methods* 47(2): 309-327.
- Wigboldus, Daniel H. J., Rob W. Holland, and Ad van Knippenberg. 2004. "Single target implicit associations." Unpublished manuscript.