Magento Test Automation Framework Installation Guide

The Magento Test Automation Framework (MTAF) is a system of software tools used for running repeatable functional tests against the Magento application being tested.

MTAF is used for both writing test automation scripts and for performing the actual testing. Test automation scripts created within the framework can be used for testing most Magento functionality which does not relate to an external system. This is a cross-platform solution (not dependent on a specific operating system). MTAF allows QA specialists to quickly develop all kinds of tests for the current Magento version, and the tests can be reused at any time. Framework users can run a single test independently, a bunch of tests together (a test suite), or all available tests.

This guide provides instructions on installing and configuring all components required for the MTAF to work properly.

Note
Installing and configuring the Magento Test Automation Framework properly requires advanced knowledge of XML, YAML, and PHP.

About this Document

This section is intended to clarify the structure, content and presentation of information in this document.

Conventions

The following text formatting conventions are used to accentuate specific types of information:

- **Bold** is used to highlight paths to files.
- ***Bold Italics*** is used to highlight file names.
- `Code Style` is used to highlight samples of code.

Glossary

All specialized terminology in this document is defined in the Glossary section of the Magento Test Automation Framework User's Guide.

Scope

This section presents a brief overview of each chapter's contents.

- Preparing the Environment for Using MTAF describes the process for installing and configuring all necessary applications.
- MTAF Configuration describes the process of configuring the Magento Test Automation Framework.
- IDE Configuration describes the process of adding the MTAF to NetBeans or PhpStorm.

Preparing the Environment for Using MTAF

For the Magento Test Automation Framework to function properly, the following software must be installed and configured:

- PHP 5.2.0 or later
- PHPUnit 3.6.8 or later
- Java Run-time Environment (JRE) 1.6 or later
- An Integration Development Environment (NetBeans 6.9.1 or later is recommended, alternatively Zend, Eclipse, etc., will suffice)
- Selenium Remote Control (RC) 1.0.3 or later, 2.0 rc2
- TortoiseGIT (recommended, but optional)
- Magento Community Edition 1.5 or later
- Web browsers:
    - Mozilla Firefox 2.x or 3.x
    - Google Chrome
    - Internet Explorer
    - Apple Safari 2.x

Before installing and configuring the required software, ensure that the environment meets all the system requirements for Magento CE.

Recommended File Structure for Windows OS

For convenience and clarity, we recommend using the following file structure:

```
C:\
|__Programs\ - Folder where environment is located
   |-------PHP\ - Folder for PHP installation
   |-------Apache\ - Folder for Apache installation
   |-------MySQL\ - Folder for MySQL installation
   |-------WWW\ - Folder for HTML\PHP content
   |-------magento-afw-x.x.x\ - Directory with unarchived MTAF
   |-------Firefox\
```

Installing MySQL Server

Download MySQL Server from the MySQL Server Download page and install it in the location recommended above.

Download Recommendations

- MySQL Community Server
- MSI Installer (32-bit OR 64-bit)

For installation instructions, refer to the MySQL Documentation.

Installing Apache

Download Apache WEB Server 2.2.x from the Apache Download page and install it in the location recommended above.

Download Recommendations

- MSI Installer (*.msi) (Win Binary)
- Pack with OpenSSL 0.9.8r

Install Recommendations
For installation instructions, refer to the Apache Server Documentation.

- Network domain: localhost.com
- Server name: www.localhost.com
- Admin email: admin@localhost.com

Configuration Recommendations

- Change the 'DocumentRoot' in the ***httpd.conf*** to **C:/Programs/www**
- Add `127.0.0.1 localhost localhost.com www.localhost.com` to the ***hosts*** file located in **C:\Windows\System32\drivers\etc**.
- RESTART the Apache server after the previous modifications.

Installing PHP

Download PHP from the PHP Download page and install it in the location recommended above.
Download Recommendations

- MSI Installer (*.msi) (Win Binary)
- Pack with Thread Safe

Install Recommendations
For installation instructions, refer to the PHP Installation and Configuration Documentation.

- Web server setup: Apache 2.2.x Module
- Apache configuration directory: **C:\Programs\Apache\conf**

Upgrading PEAR

First, it is recommended to set Windows security level to lowest and grant full permissions to **C:\Programs** (the directory where the necessary PHP software will be installed).

> **Useful Information**
> Change or add the following **Windows** PATH environment variables:
> **Win 7**: Control Panel \ System and Security \ System \ Environment Variable
> **Win XP**: Control Panel \ System \ Advanced \ Environment Variable

You may also need to add the following environment variables:

Additional System Variables
PHP_PEAR_BIN_DIR — C:\Programs\PHP
PHP_PEAR_DATA_DIR — C:\Programs\PHP\PEAR\data
PHP_PEAR_DOC_DIR — C:\Programs\PHP\PEAR\docs
PHP_PEAR_INSTALL_DIR — C:\Programs\PHP\pear
PHP_PEAR_PHP_BIN — C:\Programs\PHP\php.exe
PHP_PEAR_SYSCONF_DIR — C:\Programs\PHP
PHP_PEAR_TEST_DIR — C:\Programs\PHP\PEAR\tests
PHP_PEAR_INCLUDE_PATH — C:\Programs\PHP\PEAR

To update your PEAR installation, open http://pear.php.net/go-pear.phar in your browser and save the output to **C:\Programs\PHP\PEAR\go-pear.phar**.
NOTE: *This will replace the existing* ***go-pear.phar****.*

After changing the current directory to a PHP executable directory (in this case, **C:\Programs\PHP**), run the command `php -d phar.require_hash=0 PEAR/go-pear.phar` from the command line:

```
C:\Programs\PHP>php -d phar.require_hash=0 PEAR/go-pear.phar
```

System PATH Variables
Check that the PATH in the System Variables contains the direct path to PHP (**C:\Programs\PHP**).

After successfully completing the previous command, run the `pear upgrade` command in the same directory:

```
C:\Programs\PHP>pear upgrade
```

Installing PHPUnit

Open the command line window and run the following:

```
C:\Programs\PHP>pear channel-discover pear.phpunit.de
C:\Programs\PHP>pear channel-discover pear.symfony-project.com
C:\Programs\PHP>pear channel-discover components.ez.no
C:\Programs\PHP>pear install phpunit/PHPUnit
C:\Programs\PHP>pear install phpunit/PHPUnit_Selenium
C:\Programs\PHP>pear install phpunit/DbUnit
```

Linux note
Ubuntu: To get the PHPUnit executable file you may want to simply install it via APT instead of doing `phpize`.
NOTE: *This does not render the above step unnecessary.*

```
sudo apt-get install phpunit
```

You may also discover that a necessary Symfony component is not installed. Please make sure the `SymfonyComponents` directory exists in your PEAR directory (usually **/usr/share/php**). If you don't have it there, install the component using this command:

```
pear install symfony/YAML
```

sfYaml may conflict with the default PHP YAML component. In this case you'll need to install the component in a separate location, then manually copy the **SymfonyComponents** directory to the PEAR folder. Use this PEAR command to install the component to another location:

```
pear install -R /usr/share/php/sfYAML symfony/YAML
```

Copy the necessary folder back to a location where it can be found via the `include_path`:

```
cp -R /usr/share/php/sfYaml/usr/share/php/SymfonyComponents /usr/share/php
```

Installing Java SE Development Kit

Download the Java SE Development Kit from the Oracle Software Downloads page and install it in the desired location.

Installing NetBeans IDE 6.9.1 with the PHP pack

Download the NetBeans IDE 6.9.1 from the NetBeans IDE Download page and install it in the desired location.

Installing Selenium RC

Download Selenium Server 2.x (formerly Selenium RC Server) from the Selenium Downloads page and install it in the recommended location.

Installation & Running

1. Download the installation file.
2. Unpack the file.
3. Copy the ***selenium-server.jar*** file to a location which is convenient for running it through the command line (for example: **C\:**).
4. Run command line.
    a. Run command: `C:\>java -jar selenium-server.jar`.
    b. Confirm that the expected output is produced:

```
c:\>java -jar selenium-server.jar
23:46:04.852 INFO - Java: Sun Microsystems Inc. 19.0-b09
23:46:04.865 INFO - OS: Windows 7 6.1 x86
23:46:04.934 INFO - v2.0 [a2], with Core v2.0 [a2]
23:46:05.347 INFO - RemoteWebDriver instances should connect to:
http://127.0.0.1:4444/wd/hub
23:46:05.348 INFO - Version Jetty/5.1.x
23:46:05.349 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
23:46:05.351 INFO - Started HttpContext[/selenium-server,/selenium-server]
23:46:05.352 INFO - Started HttpContext[/,/]
23:46:05.502 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@19134f4
23:46:05.502 INFO - Started HttpContext[/wd,/wd]
23:46:05.517 INFO - Started SocketListener on 0.0.0.0:4444
23:46:05.517 INFO - Started org.openqa.jetty.jetty.Server@b6ece5
```

This output means the server started successfully and is ready to work.

c. Clicking **Ctrl+C** will SHUTDOWN Selenium Server.

Installing TortoiseGIT for Updating Your MTAF Copy from the Repository

Download GIT from the msysgit Downloads page and install it in the desired location.

Download TortoiseGIT from the TortoiseGIT Download page and install it in the desired location.

Downloading and Installing the MTAF Project

To install the Magento Test Automation Framework, clone the MTAF source tree from GIT:

```
git clone http://git.magento.com/taf DIR_for_CLONE
```

Downloading, Installing and Configuring Magento Community Edition

Choose one of the following ways of downloading Magento Community Edition:

- As a source tree from SVN: **Instructions for creating SVN checkout**
- As an archived pack: **Download the latest build of Magento Community Edition**.

Install and configure

- Unpack the downloaded archive.
- Install and configure Magento Community Edition according to the instructions in **Magento Knowledge Base > Installation & Configuration**.

Attention
When Magento Community Edition has been installed you SHOULD disable use of 'Secret Key in URLs':

- Login to the Magento Admin Panel.
- Open System > Configuration.
- Open Advanced > Admin > Security.
- Change the value of 'Add Secret Key to URLs' to "No".
- Click the 'Save Config' button.

Configuring Mozilla FireFox

Download Mozilla FireFox 3.6.x from the Firefox Download page and install it in the location recommended above.

Important
You need to install **Mozilla Firefox 3.x**. Selenium RC is currently NOT compatible with Firefox 4.

Each time Selenium starts running a test script, a new browsing session is created. This means that a new Firefox process is started with a new, "fresh" profile. Thus issues with previously accepted cookies, stored passwords, etc., are avoided and each test runs in an identical browser environment.

Making Firefox Available to Selenium RC

For the browser to run successfully, Selenium RC server must be able to locate the ***firefox.exe*** file. Please add the full path to the system Path variable.

In some operating systems (including Windows 7 and Windows XP), Selenium RC will not be able to locate ***firefox.exe*** when the full path contains spaces (for example, **C:\Program Files\Mozilla\Firefox**). There are two possible solutions for this:

- Reinstall Firefox to a different directory (for example, **C:\Programs\Firefox**).
  or
- Use the built-in Windows7 mklink utility as follows:

```
mklink /D  C:\Firefox "C:\Program Files (x86)\Mozilla Firefox\"
```

System PATH Variables
Add the direct path to Firefox or its link (**C:\Programs\Firefox**) to the PATH in System Variables.

Using a Custom Firefox Profile

To use a custom Firefox profile for Selenium testing, you should use the `-firefoxProfileTemplate` parameter.

You must start (or restart) Selenium Server using the custom profile:

```
c:\>java -jar selenium-server.jar -firefoxProfileTemplate “<Selenium Profile Directory>”
```

Installing Add-ons for Firefox

To install add-ons for Firefox, go to Tools > Add-ons and then select the following add-ons (as needed):

- **Firebug** allows you to edit, debug, and monitor CSS, HTML, and JavaScript in the application being tested.
- **Selenium IDE** allows you to record, edit, and debug Selenium tests.
- **ScreenGrab** saves entire webpages as images.
- **FirePath** is a Firebug extension that adds a development tool for editing, inspecting, and generating XPath 1.0 expressions, CSS 3 selectors and JQuery selectors.
- **Web Developer** is an extension which adds various web developer tools to the browser.
- **Remember Certificate Exception** allows you to automatically add Certificate Exceptions.

Creating a Selenium Profile

It is strongly recommended that you create a pre-configured Firefox profile for running Selenium tests.

Before you begin, make sure all of your Firefox instances are closed.

1. Click 'Windows button' > 'Run' (for Windows XP) or 'Win+R' (all Windows versions).
2. Type `firefox.exe -ProfileManager -no-remote`.
3. In the 'Choose User profile' window click “Create Profile”.
4. Click “Next”.
5. Enter a new profile name (e.g., selenium).
6. Select a directory folder to store your new profile.
7. Click “Finish”.
8. Confirm that the “Don’t ask at startup” check box is selected.
9. Click “Start Firefox” for the profile being created and configure the settings based on the Suggested Settings for Selenium Profile\.
   NOTE: *After starting Firefox with this profile it becomes the default. To reset the previous “default” profile to be the default again, start Firefox with that profile (if needed).*

Suggested Settings for Selenium Profile

The following actions are performed in the Firefox toolbar menu.

- Click “View" > "Toolbars” > and clear the “Bookmarks Toolbar” check box.
- Right-click the toolbar and click “Customize”.
    - Remove “Google search” by dragging it to the 'Customize Toolbar' window.
    - In the 'Customize Toolbar' window, select the “Use Small Icons” check box and then click “Done”.
- Click “Tools" > "Options” and set the following in the 'Options' window:
    - On the “General” tab:
        - Set 'Home Page' to “about:blank”.
        - Clear the “Show the Downloads...” option.
    - On the “Tabs” tab:
        - Select "Open new windows in a new tab instead" (Firefox 3.6) OR "a new window" for new pages (older FF versions).
        - Clear all warning options.
    - On the “Content” tab:
        - Clear the “Block pop-up windows" option.
    - On the “Privacy” tab:
        - From the "Firefox will" drop-down list, select "Use custom settings for history".
            - Clear all “History” options.
    - On the “Security” tab:
        - Clear all “Security” options.
        - Click “Settings” and clear all warning options.
    - On the “Advanced” tab:
        - Select the “General” subtab.
            - Clear the “Use autoscrolling” option under the 'Browsing' group.
        - Select the “Update” subtab.
            - Clear all options under "Automatically check for updates to" group.
            - Clear the “Warn me ...” option.
        - Select the "Encryption" subtab
            - In the "When a server requests my personal certificate" group:
                - Select the "Select one automatically" option button.
                - Click the "View Certificates" button.
                    - In the "Certificate Manager" window:
                        - On the "Servers" tab:
                            - Click the "Add Exception..." button.
                            - In the "Location" field, enter your main URL (e.g. **https://www.mydomain.com/**).
                            - Click the "Get Certificate" button.
                            - Ensure that the "Permanently store this exception" check box is selected.
                            - Click the "Confirm Security Exception" button.
                            - In the "Certificate Manager" window, click "OK".
- Click "OK" in the "Options" window.
- Verify that the Cookies are enabled:
    - On the "Web Developer" toolbar select "Cookies" > "Disable Cookies".
        - Ensure that "All Cookies" is NOT selected.

It can also be helpful to do the following:

- Install useful add-ons (FireBug, Selenium IDE, ScreenGrab, Web Developer, Remember Certificate Exception).
- Accept SSL-certificates for working websites before using.

MTAF Configuration

Selenium Server Configuration

Before you start using the MTAF, you need to properly configure the Selenium Server and the Selenium Client.

To configure the necessary Selenium Server settings, simply run the proper set of commands in the command line. For example, it might be similar to the following:

Example of Selenium Server Settings

```
java -jar selenium-server-standalone-2.4.0.jar -trustAllSSLCertificate
```

Where `trustAllSSLCertificate` specifies overriding all https warnings.

For more information on configuring Selenium Server, refer to the Selenium Server website.

Selenium Client Configuration

To configure the Selenium Client, you should use either the ***config.yml*** file or the ***local.yml*** file. Both files are located in the **config** sub-folder of your local MTAF installation directory (for example, `..\magento-afw-x.x.x\config\config.yml`).

> **Warning**
> Be aware that the ***config.yml*** file (the original configuration template) is always under version control. This means that if you upload a new MTAF installation from the version controlrepository, all the settings you have previously made in this file will be overridden. On the other hand, if you commit changes made in the ***config.ym*** file to the repository, it may later affect other MTAF users. Therefore, it is strongly recommended to make local configuration changes ONLY within the ***local.yml*** file. The ***local.yml*** file is similar to the ***config.yml*** file and, when it exists, is used in its place. Moreover, this file is not managed by version control, so you can always perform your own configuration settings without effecting the original configuration template.

The following is a typical example of the "default" ***config.yml*** file and a description of its contents.

Example of browsers.yml File

```
// List of browsers which can be used
browsers:
 // Default browser settings
    // Browser name
    chrome: &chrome
        name: Firefox
        browser: '*chrome'
        // Host where it is installed and will run
        host: 127.0.0.1
         // Port on the Host where this browser will be available
        port: 4444
        // Parameter which sets up the frequency of browser running:
        // false - Browser starts every time when new tests are run and shuts down in the end of each
test (recommended).
        // true - Browser starts ONLY before the first test and shuts down after the last one.
        doNotKillBrowsers: false
        // Parameter sets up the count of tests which must be run
        // sequentially before the browser will forcibly shut down.
        browserTimeoutPeriod: 40000
    firefox: &firefox
        name: Firefox
        browser: '*firefox'
        host: 127.0.0.1
        port: 4444
        doNotKillBrowsers: false
        browserTimeoutPeriod: 40000
    iexplore: &iexplore
        name: Internet Explorer
        browser: '*iexplore'
        host: 127.0.0.1
        port: 4444
        doNotKillBrowsers: false
        browserTimeoutPeriod: 40000
    googlechrome: &googlechrome
        name: Google Chrome
        browser: '*googlechrome'
        host: 127.0.0.1
        port: 4444
        doNotKillBrowsers: false
        browserTimeoutPeriod: 40000
    default: *chrome
default:
    //Cache settings
    cache:
        frontend:
            name: 'core'
            options:
                caching: false
                cache_id_prefix: 'selenium_'
```

```
            automatic_serialization: true
    backend:
        name: 'file'
        options:
        cache_dir: 'tmp/cache'
uimaps:
//Path to uimap root folder
    basePath: uimaps
// List of applications which can be tested
applications:
    // Settings of Magento application
    magento: &magento
        // Login to the backend
        adminLogin: admin
        // Password to the backend
        adminPassword: admin
        storeName: Store Name
        areas:
            admin:
                // General link to the backend
                url: 'http://www.localhost.com/magento/admin/'
                //Relative path to uimap files for the area
                uimap_path: 'admin'
            frontend:
                // General link to the frontend
                url: 'http://www.localhost.com/magento/'
                //Relative path to uimap files for the area
                uimap_path: 'frontend'
            paypal-sandbox:
                //General link to third-party site
                url: 'https://www.sandbox.paypal.com/'
                //Relative path to uimap files for the area
                uimap_path: 'third-party/paypal-sandbox'
            paypal-developer:
                url: 'https://developer.paypal.com/'
                uimap_path: 'third-party/paypal-developer'
    // Pointer to the default application setting.
```

```
        // Used when more than one application is described
        // and one of which is required to be used by default
        default: *magento
```

- `browsers` lists and configures the browsers that can be used to run both Magento client and admin branches.
  - `browser: '*chrome'` is the browser's identification accepted by Selenium RC. Keep in mind that the operating system, not Magento, is set up for a specific 'default' browser.
  - `host` is the identification of the machine where Selenium RC is installed.
  - `port` is used when one needs to support several servers on a single host, and thus needs to use different ports.
  - `default` identifies the default browser to be used for running a test or suite of tests. This 'browser identification' option is used only to specify the required browser for a single test run. Thus, you do not need to specify other browsers, simply classify the necessary one.

In fact, as you can see from the above example, the ***browser.yml*** file presents a description of browsers supported, and a description of the Magento application itself - its location and installation paths.

Tip
Selenium RC can work on remote computers, which means that the locations of the real tests, the Magento version being tested, and the testing environment are irrelevant . For example, it may happen that the three mentioned instances are located on three different virtual machines. Even so, if you properly configure the settings such as `frontendUrl`, `adminUrl`, `adminLogin`, `adminPassword` and `browsers`, the MTAF engine will work properly.

PHPUnit Configuration

- ***phpunit.xml*** contains the list of test scripts (test cases) to be run. (`..\magento-afw-x.x.x\phpunit.xml`)
  phpunit.xml is the standard PHPUnit configuration file. The file defines a test suite (the list of test scripts/test cases) to be run. In other words, it contains the suite's name, a list of suffixes, and a list of directories that the necessary PHP files will be run from. For example:

Example of phpunit.xml File

```
<phpunit
    bootstrap="bootstrap.php"
    colors="false"
    convertErrorsToExceptions="true"
    convertNoticesToExceptions="true"
    convertWarningsToExceptions="true"
    stopOnFailure="false"
    syntaxCheck="false"
    verbose="true"
    strict="false"
    printsummary="true">
    <testsuites>
        <testsuite name="All Tests">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
    <logging>
        <log type="coverage-html" target="./tmp/report" charset="UTF-8" yui="true" highlight="false"
         lowUpperBound="35" highLowerBound="70"/>
        <log type="coverage-xml" target="./tmp/coverage.xml"/>
        <log type="graphviz" target="./tmp/logfile.dot"/>
        <log type="json" target="./tmp/logfile.json"/>
        <log type="metrics-xml" target="./tmp/metrics.xml"/>
        <log type="plain" target="./tmp/logfile.txt"/>
        <log type="pmd-xml" target="./tmp/pmd.xml" cpdMinLines="5" cpdMinMatches="70"/>
        <log type="tap" target="./tmp/logfile.tap" logIncompleteSkipped="true"/>
        <log type="junit" target="./tmp/logfile.xml" logIncompleteSkipped="false"/>
        <log type="testdox-html" target="./tmp/testdox.html"/>
        <log type="testdox-text" target="./tmp/testdox.txt"/>
    </logging>
</phpunit>
```

Where:

- `<testsuite name="All Tests">` is the name of the suite to be run

- `suffix="Test.php"` is the suffix
- `tests` is the name of the directory from which the tests will run F

Based on these specific settings in the configuration file, the suite to be run will include all PHP files (tests) from the **tests** directory which have the `Test.php` suffix. If there are no files with the specified suffix, only the mentioned file will be run as a test.

Warning
All PHP files (test cases) to be executed must be named using CamelCase style (the initial letter of each element's within the compound capitalized, and the first letter either upper or lower case). This means that if you have PHP files named ***File1Test.php***, or ***Test1Test.php***, those will be run within the test suite. However, if you name a test case ***test222.php*** it will not be a part of the sample suite, because it does not meet either suffix or style rules.

IDE Configuration

The installed IDE must be configured to access the MTAF. If you are using an IDE for which information is not provided below, refer to the documentation for that product for appropriate instructions.

Adding MTAF to the NetBeans IDE

- Run NetBeans IDE.
- Click "File" > "New Project".
- Step 1. Choose Project:
  - Select Categories > 'PHP'.
  - Select Projects > 'PHP Application with Existing Sources'.
  - Click 'Next'.
- Step 2. Name and Location:
  - In the 'Sources Folder' field, set the path to the project.
  - In the 'PHP version' field, select PHP 5.3.
  - Click "Next".
- Step 3. Run Configuration:
  - In the 'Run As' field, select the 'Script (run in command line)' value.
  - Enter the path to the ***php.exe*** file in the 'PHP Interpreter' field.
  - Ensure the 'Index File' field is blank.
  - Click 'Finish'.
- In the Projects window, right-click the node for your project and select "Properties".
  - On the 'Sources' tab:
    - In the 'Test Folder' field, set the path to the folder with tests.
  - On the 'PHPUnit' tab:
    - Select 'Use XML Configuration' and in the 'XML Configuration' field, select ***phpunit.xml*** file from the project's folder.
  - Click 'Ok'.

Adding MTAF to PhpStorm IDE

It is supposed that you already have a project with Magento and it is opened.

- Click 'Run' > 'Edit Configurations'.
- Click 'Add New Configuration' (the yellow "**+**" at the top left corner).
- Choose 'PHPUnit' (not to be confused with 'PHPUnit on server').
- Enter 'Name' (e.g. "Selenium").
- On 'Configuration' subtab choose 'XML File'.
- Select path to ***phpunit.xml*** in 'Use XML configuration file'.
- Click 'OK'.

You can now run PHPUnit tests by choosing "Selenium" (or whatever name you've entered) from the drop-down menu near the "Run" and "Debug" buttons, then clicking the "Run" button.