

# Truffle Quick Guide

## Contents

Quick Start.....	2
Creating a project: .....	2
Testing .....	2
Migrating (deploying) with Ganache .....	3
Interacting with the contract.....	3

## Quick Start

<https://trufflesuite.com/docs/truffle/>

<https://trufflesuite.com/tutorial/>

### With Truffle, you get:

- Contract compilation, linking, deployment & testing
- Network management for deploying to any number of public & private networks
- Package management with EthPM & NPM, using the ERC190 standard.
- Interactive console for direct contract communication.
- Configurable build pipeline with support for tight integration.
- External script runner that executes scripts within a Truffle environment.

*Installing Truffle: `npm install -g truffle`*

### Creating a project:

`truffle unbox metacoin`

You can use the `truffle unbox <box-name>` command to download any of the other Truffle Boxes.

To create a bare Truffle project with no smart contracts included, use `truffle init`

Open the `contracts/Migrations.sol` file. This is a separate Solidity file that manages and updates the status of your deployed smart contract

Open the `migrations/1_initial_migration.js` file. This file is the migration (deployment) script for the Migrations contract found in the `Migrations.sol` file

Open the `migrations/2_deploy_contracts.js` file. This file is the migration script for the MetaCoin contract. (Migration scripts are run in order, so the file beginning with 2 will be run after the file beginning with 1.)

### Testing

Run the Solidity test: `truffle test ./test/TestMetaCoin.sol`

Run the JavaScript test: `truffle test ./test/metacoin.js`

*Compiling: `truffle compile`*

## Migrating (deploying) with Ganache

Open truffle-config.js and replace the content with the following:

```
module.exports = {  
  networks: {  
    development: {  
      host: "127.0.0.1",  
      port: 8545,  
      network_id: "*"  
    }  
  }  
};
```

Run: **truffle migrate**

## Interacting with the contract

To interact with the contract, you can use the Truffle console. The Truffle console is similar to Truffle Develop, except it connects to an existing blockchain (Ganache)

Run: **truffle console**

```
truffle(development)> let instance = await MetaCoin.deployed()  
truffle(development)> let accounts = await web3.eth.getAccounts()  
truffle(development)> let balance = await instance.getBalance(accounts[0])  
truffle(development)> balance.toNumber()  
truffle(development)> let ether = await instance.getBalanceInEth(accounts[0])  
truffle(development)> ether.toNumber()  
truffle(development)> instance.sendCoin(accounts[1], 500)
```