**Safe Contracts**

**==============**

**[![npm version](**[**https://badge.fury.io/js/%40gnosis.pm%2Fsafe-contracts.svg**](https://badge.fury.io/js/%40gnosis.pm%2Fsafe-contracts.svg)**)](**[**https://badge.fury.io/js/%40gnosis.pm%2Fsafe-contracts**](https://badge.fury.io/js/%40gnosis.pm%2Fsafe-contracts)**)**

**[![Build Status](**[**https://github.com/safe-global/safe-contracts/workflows/safe-contracts/badge.svg?branch=development**](https://github.com/safe-global/safe-contracts/workflows/safe-contracts/badge.svg?branch=development)**)](**[**https://github.com/safe-global/safe-contracts/actions**](https://github.com/safe-global/safe-contracts/actions)**)**

**[![Coverage Status](**[**https://coveralls.io/repos/github/safe-global/safe-contracts/badge.svg?branch=development**](https://coveralls.io/repos/github/safe-global/safe-contracts/badge.svg?branch=development)**)](**[**https://coveralls.io/github/safe-global/safe-contracts**](https://coveralls.io/github/safe-global/safe-contracts)**)**

**> :warning: \*\*This branch contains changes that are under development\*\* To use the latest audited version make sure to use the correct commit. The tagged versions that are used by the Safe team can be found in the [releases](**[**https://github.com/safe-global/safe-contracts/releases**](https://github.com/safe-global/safe-contracts/releases)**).**

**Usage**

**-----**

**### Install requirements with yarn:**

**```bash**

**yarn**

**```**

**### Testing**

**To run the tests:**

**```bash**

**yarn build**

**yarn test**

**```**

**Optionally, if you want to run the ERC-4337 compatibility test, it uses a live bundler and node, so it contains some pre-requisites:**

**1. Define the environment variables:**

**```**

**ERC4337\_TEST\_BUNDLER\_URL=**

**ERC4337\_TEST\_NODE\_URL=**

**ERC4337\_TEST\_SINGLETON\_ADDRESS=**

**ERC4337\_TEST\_SAFE\_FACTORY\_ADDRESS=**

**MNEMONIC=**

**```**

**2. Pre-fund the executor account derived from the mnemonic with some Native Token to cover the deployment of an ERC4337 module and the pre-fund of the Safe for the test operation.**

**### Deployments**

**A collection of the different Safe contract deployments and their addresses can be found in the [Safe deployments](**[**https://github.com/safe-global/safe-deployments**](https://github.com/safe-global/safe-deployments)**) repository.**

**To add support for a new network follow the steps of the ``Deploy`` section and create a PR in the [Safe deployments](**[**https://github.com/safe-global/safe-deployments**](https://github.com/safe-global/safe-deployments)**) repository.**

**### Deploy**

**> :warning: \*\*Make sure to use the correct commit when deploying the contracts.\*\* Any change (even comments) within the contract files will result in different addresses. The tagged versions that are used by the Safe team can be found in the [releases](**[**https://github.com/safe-global/safe-contracts/releases**](https://github.com/safe-global/safe-contracts/releases)**).**

**> \*\*Current version:\*\* The latest release is [v1.3.0-libs.0](**[**https://github.com/safe-global/safe-contracts/tree/v1.3.0-libs.0**](https://github.com/safe-global/safe-contracts/tree/v1.3.0-libs.0)**) on the commit [767ef36](**[**https://github.com/safe-global/safe-contracts/commit/767ef36bba88bdbc0c9fe3708a4290cabef4c376**](https://github.com/safe-global/safe-contracts/commit/767ef36bba88bdbc0c9fe3708a4290cabef4c376)**)**

**This will deploy the contracts deterministically and verify the contracts on etherscan using [Solidity 0.7.6](**[**https://github.com/ethereum/solidity/releases/tag/v0.7.6**](https://github.com/ethereum/solidity/releases/tag/v0.7.6)**) by default.**

**Preparation:**

**- Set `MNEMONIC` in `.env`**

**- Set `INFURA\_KEY` in `.env`**

**```bash**

**yarn deploy-all <network>**

**```**

**This will perform the following steps**

**```bash**

**yarn build**

**yarn hardhat --network <network> deploy**

**yarn hardhat --network <network> sourcify**

**yarn hardhat --network <network> etherscan-verify**

**yarn hardhat --network <network> local-verify**

**```**

**#### Custom Networks**

**It is possible to use the `NODE\_URL` env var to connect to any EVM based network via an RPC endpoint. This connection then can be used with the `custom` network.**

**E.g. to deploy the Safe contract suite on that network you would run `yarn deploy-all custom`.**

**The resulting addresses should be on all networks the same.**

**Note: Address will vary if contract code is changed or a different Solidity version is used.**

**#### Replay protection (EIP-155)**

**Some networks require replay protection, making it incompatible with the default deployment process as it relies on a presigned transaction without replay protection (see**[**https://github.com/Arachnid/deterministic-deployment-proxy**](https://github.com/Arachnid/deterministic-deployment-proxy)**).**

**Safe contracts use a different deterministic deployment proxy (**[**https://github.com/safe-global/safe-singleton-factory**](https://github.com/safe-global/safe-singleton-factory)**). To make sure that the latest version of this package is installed, make sure to run `yarn add @gnosis.pm/safe-singleton-factory` before deployment. For more information, including how to deploy the factory to a new network, please refer to the factory repo.**

**Note: This will result in different addresses compared to hardhat's default deterministic deployment process.**

**### Verify contract**

**This command will use the deployment artifacts to compile the contracts and compare them to the onchain code**

**```bash**

**yarn hardhat --network <network> local-verify**

**```**

**This command will upload the contract source to Etherescan**

**```bash**

**yarn hardhat --network <network> etherscan-verify**

**```**

**Documentation**

**-------------**

**- [Safe developer portal]([http://docs.safe.global](http://docs.safe.global/" \t "_blank))**

**- [Error codes](docs/error\_codes.md)**

**- [Coding guidelines](docs/guidelines.md)**

**Audits/ Formal Verification**

**---------**

**- [for Version 1.4.0/1.4.1 by Ackee Blockchain](docs/audit\_1\_4\_0.md)**

**- [for Version 1.3.0 by G0 Group](docs/audit\_1\_3\_0.md)**

**- [for Version 1.2.0 by G0 Group](docs/audit\_1\_2\_0.md)**

**- [for Version 1.1.1 by G0 Group](docs/audit\_1\_1\_1.md)**

**- [for Version 1.0.0 by Runtime Verification](docs/rv\_1\_0\_0.md)**

**- [for Version 0.0.1 by Alexey Akhunov](docs/alexey\_audit.md)**

**Security and Liability**

**----------------------**

**All contracts are WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.**

**License**

**-------**

**All smart contracts are released under LGPL-3.0**