**MATRX 一对多交易测试用例**

**版本: V1.0**

**作者：MATRIX 基金会北京研究院**

**日期：2018.07.19**

任何情况下，与本系统及其衍生产品，以及与之相关的全部文件（包括本文件及其任何附件中的全部信息）相关的全部知识产权（包括但不限于版权、商标和技术秘密）皆属于 MATRIX 基金会所有。

本文件中的信息是保密的，且仅供用户指定的接收人内部使用。未经 MATRIX 基金会事先书面同意，本文件的任何用户不得对本产品和本文件中的信息向任何第三方（包括但不限于除用户指定接收人以外的管理人员、员工和关联公司）进行披露、出借、许可、转让、出售、分发、传播或进行与本产品和本文件相关的任何其他处置，也不得使该等第三方以任何形式使用本产品和本文件中的信息。

未经 MATRIX 基金会事先书面允许，不得为任何目的、以任何形式或任何方式对本文件进行复制、修改或分发。本文件的任何用户不得更改、移除或损害本文件所使用的任何商标。

本文件按“原样”提供，就本文件的正确性、准确性、可靠性或其他方面，MATRIX 基金会并不保证本文件的使用或使用后果。本文件中的全部信息皆可能在没有任何通知的情形下被进一步修改，MATRIX 基金会对本文件中可能出现的任何错误或不准确之处不承担任何责任。

在任何情况下，MATRIX 基金会均不对任何因使用本软件产品和本文件中的信息而引起的任何直接损失、间接损失、附带损失、特别损失或惩罚性损害赔偿（包括但不限于获得替代商品或服务、丧失使用权、数据或利润；或商业中断），责任或侵权（包括过失或其他侵权）承担任何责任，即使 MATRIX 基金会事先获知上述损失可能发生。

MATRIX 基金会保留所有权利

# 目 录

[目 录 1](#_Toc520216945)

[变更记录 2](#_Toc520216946)

[1 以太坊钱包兼容性测试 3](#_Toc520216947)

[2 一对多交易相同地址验证 4](#_Toc520216948)

[3 新交易格式的验证 5](#_Toc520216949)

[4 验证一对多交易中账户为空 6](#_Toc520216950)

[5验证一对多交易中子交易最大不超过1000 7](#_Toc520216951)

[6 验证一对多交易gas验证失败 8](#_Toc520216952)

[7验证一对多交易余额不足 9](#_Toc520216953)

[8一对一交易给Txtype或者localHieght赋值检查结果 10](#_Toc520216954)

[9一对多交易字段为空验证 11](#_Toc520216955)

[10发送交易类型字段验证 12](#_Toc520216956)

[11发送交易填写nonce验证 13](#_Toc520216957)

[12广播交易时nonce验证 14](#_Toc520216958)

[13接受到广播交易后nonce验证 15](#_Toc520216959)

[14转账金额验证 16](#_Toc520216960)

# 变更记录

| 日期 | 版本号 | 修订章节 | 描述 | 作者 | 评审者 | 评审日期 |
| --- | --- | --- | --- | --- | --- | --- |
| 2018-7-19 | V0.0.1 |  | 一对多交易测试用例 | 叶营 |  |  |
|  |  |  |  |  |  |  |

# 以太坊钱包兼容性测试

| **用例编号** | **TC\_Matrix\_交易\_以太坊钱包兼容性** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 按照标准以太坊交易格式发送交易，节点能够识别，并正确打包 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 执行eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(0.01)})命令   预期测试结果检查：  1.能够验证通过该笔交易；  2.能够通过双方账户余额判断转账是否成功； | | | | | |
| 期望输出 |  | | 实际输出 | |  |
| 测试结果分析和建议 |  | | | | |

表格 1 [用例名称]

# 一对多交易相同地址验证

| **用例编号** | **TC\_Matrix\_交易\_一对多交易稳定性测试** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 一对多交易中出现多个相同的被转账地址 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa ","value":web3.toWei(0.01)}]}) 3. 查看该交易是否能够验证成功   预期测试结果检查：  1. 不允许出现多个相同的被转账地址出现； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

表格 3 [用例名称]

# 新交易格式的验证

| **用例编号** | **TC\_Matrix\_交易\_一对多交易稳定性测试** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 所有交易类型（一对一和一对多）均功能测试通过 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})   3.查看该笔交易是否能够执行成功  预期测试结果检查：  1.一对多和一对一交易都能执行成功；  2.一对多和一对一交易中账户的余额能够正确增减； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

# 验证一对多交易中账户为空

| **用例编号** | **TC\_Matrix\_交易\_一对多交易账户不为空** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 创建一对多交易，其中有些账户为空 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "","value":web3.toWei(0.01)}]})   3.查看该笔交易是否能够执行成功  预期测试结果检查：  1.该笔交易能执行成功；  2.该笔交易中所有的交易都能转账成功；  3.所涉及的账户余额变化； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

# 5验证一对多交易中子交易最大不超过1000

| **用例编号** | **TC\_Matrix\_交易\_限制一对多子交易数** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 验证一对多交易中子交易数最大不超过1000个。本用例是用来测试逻辑所以测试时可以将1000降低 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})   3.查看该笔交易是否能够执行成功  预期测试结果检查：  1.如果转账的地址多于设计的最高个数则转账失败，各个账户余额无变化； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限**

# 6 验证一对多交易gas验证失败

| **用例编号** | **TC\_Matrix\_交易\_**验证失败 | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送一对多交易其中gas小于每笔gas之和，测试交易验证失败 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),gas:“21000”"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})   3.或者不输入gas  4.查看该笔交易是否能够执行成功  预期测试结果检查：  1.交易验证失败无法执行该交易； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 7验证一对多交易余额不足

| **用例编号** | **TC\_Matrix\_交易\_验证余额** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送一对多交易验证账户余额是否充足 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})   3.查看该笔交易是否能够执行成功  预期测试结果检查：  1.交易验证失败提示余额不足； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 8一对一交易给Txtype或者localHieght赋值检查结果

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送一对一交易给txtype或者localHieght赋值，然后验证是否按照一对一的流程正确执行交易（一对一交易走一对多流程） | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(0.01)，txType:0})   3.查看该笔交易是否能够执行成功  预期测试结果检查：   1. 能够正确转账并通过账户余额来验证，同时走的是一对多交易 | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 9一对多交易字段为空验证

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送一对多交易分别将交易的字段比如：txtype，localheight,extra 为空验证执行结果 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：   1. 启动以太坊节点（3台docker节点） 2. 创建一笔交易eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":,"lockTime":,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})   3.查看该笔交易是否能够执行成功  预期测试结果检查：  1.交易执行成功；  2.可以给第一个账户转账成功并通过余额验证；  3.给扩展交易的账户转账成功通过余额验证； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 10发送交易类型字段验证

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送交易将txtype设为0，将localHeight和extra正确填写验证执行结果 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：  1.创建一笔交易一对一一对多都可以  2.查看该交易是否能够验证成功  3.查看该笔交易是否能够执行成功  测试结果检查：  1.本笔交易第一个账户能够执行成功并通过余额进行验证；  2.其他的扩展交易也能执行成功； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 11发送交易填写nonce验证

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 发送交易时如果手动填写Nonce需要验证nonce是否大于0x8000000000000000 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：  1.创建一笔交易一对一一对多都可以并手动输入Nonce  2.查看该交易是否能够验证成功  3.查看该笔交易是否能够执行成功  测试结果检查：  1.如果输入的Nonce是原来以太坊的格式则验证失败交易不执行；  2.如果按照Matrix格式输入Nonce则验证成功并能正确执行交易，通过余额进行判断； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 12广播交易时nonce验证

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 创建一笔正常交易，然后广播交易，验证Nonce | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：  1.创建一笔交易一对一一对多都可以  2.查看该交易是否能够验证成功  3.查看该笔交易是否能够执行成功  测试结果检查：  1.通过打印日志来判断广播前是否将Nonce & 0x7FFFFFFFFFFFFFFF，同时是否验证通过并执行； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 13接受到广播交易后nonce验证

| **用例编号** | **TC\_Matrix\_交易\_验证交易** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 接收到广播交易验证是否将Nonce | 0x8000000000000000 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：  1.创建一笔交易一对一一对多都可以  2.查看该交易是否能够验证成功  3.查看该笔交易是否能够执行成功  测试结果检查：  1.通过打印日志来判断是否将Nonce | 0x8000000000000000，同时是否验证通过并执行； | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**

# 14转账金额验证

| **用例编号** | **TC\_Matrix\_交易\_验证转账金额** | **用例类型** | | **功能测试** | |
| --- | --- | --- | --- | --- | --- |
| 用例描述 | 创建一笔交易（一对多或者一对一都可以）输入一个超级大（最大数的一半）的金额，检查是否会有溢出 | | | | |
| 设计人 | 叶营 | 设计日期 | | 2018-07-14 | |
| 执行人 |  | 执行日期 | |  | |
| 关联的需求名称 |  | 涉及UI | |  | | |
| 测试条件 | 3台docker节点 | | | | | |
| 输入数据 |  | | | | |
| 测试过程  测试步骤：  1.创建一笔交易一对一一对多都可以  2.查看该交易是否能够验证成功  3.查看该笔交易是否能够执行成功  测试结果检查：   1. 转账失败 应该是账户余额不足。 2. 内存并无明显变化 | | | | | |
| 期望输出 |  | | 实际输出 | |  |

**限制**