```
BIP: 370
Layer: Applications
Title: PSBT Version 2
Author: Andrew Chow <achow101@gmail.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0370
Status: Draft
Type: Standards Track
Created: 2021-01-14
License: BSD-2-Clause
```

Introduction

Abstract

This document proposes a second version of the Partially Signed Bitcoin Transaction format described in BIP 174 which allows for inputs and outputs to be added to the PSBT after creation.

Copyright

Motivation

Partially Signed Bitcoin Transaction Version 0 as described in BIP 174 is unable to have new inputs and outputs be added to the transaction. The fixed global unsigned transaction cannot be changed which prevents any additional inputs or outputs to be added. PSBT Version 2 is intended to rectify this problem.

An additional beneficial side effect is that all information for a given input or output will be provided by its or . With Version 0, to retrieve all of the information for an input or output, data would need to be found in two locations: the / and the global unsigned transaction. PSBT Version 2 now moves all related information to one place.

Specification

PSBT Version 2 (PSBTv2) only specifies new fields and field inclusion/exclusion requirements.

`PSBT_GLOBAL_UNSIGNED_TX` must be excluded in PSBTv2. `PSBT_GLOBAL_VERSION` must be included in PSBTv2 and set to version number 2[1].

The new global types for PSBT Version 2 are as follows:

[1] **What happened to version number 1?** Version number 1 is skipped because PSBT Version 0 has been colloquially referred to as version 1. Originally this BIP was to be version 1, but because it has been colloquially referred to as version 2 during its design phrase, it was decided to change the version number to 2 so that there would not be any confusion

Name			Description	
Transaction Version	`PSBT_GLOBAL_TX_VERSION = 0x02`	None	No key data	`<32-bit`
Fallback Locktime	`PSBT_GLOBAL_FALLBACK_LOCKTIME = 0x03`	None	No key data	`<32-bit`
Input Count	`PSBT_GLOBAL_INPUT_COUNT = 0x04`	None	No key data	
Output Count	`PSBT_GLOBAL_OUTPUT_COUNT = 0x05`	None	No key data	
Transaction Modifiable Flags	`PSBT_GLOBAL_TX_MODIFIABLE = 0x06`	None	No key data	`<8-bit u`

The new per-input types for PSBT Version 2 are defined as follows:

Name			Description	
Previous TXID	`PSBT_IN_PREVIOUS_TXID = 0x0e`	None	No key data	`<3`
Spent Output Index	`PSBT_IN_OUTPUT_INDEX = 0x0f`	None	No key data	`<3`
Sequence Number	`PSBT_IN_SEQUENCE = 0x10`	None	No key data	`<3`
Required Time-based Locktime	`PSBT_IN_REQUIRED_TIME_LOCKTIME = 0x11`	None	No key data	`<3`
Required Height-based Locktime	`PSBT_IN_REQUIRED_HEIGHT_LOCKTIME = 0x12`	None	No key data	`<3`

The new per-output types for PSBT Version 2 are defined as follows:

Name			Description	
Output Amount	`PSBT_OUT_AMOUNT = 0x03`	None	No key data	`<64-bit little endian int amount>`
Output Script	`PSBT_OUT_SCRIPT = 0x04`	None	No key data	

Determining Lock Time

The nLockTime field of a transaction is determined by inspecting the PSBT_GLOBAL_FALLBACK_LOCKTIME and each input's PSBT_IN_REQUIRED_TIME_LOCKTIME and PSBT_IN_REQUIRED_HEIGHT_LOCKTIME fields. If none of the inputs have a PSBT_IN_REQUIRED_TIME_LOCKTIME and PSBT_IN_REQUIRED_HEIGHT_LOCKTIME, then PSBT_GLOBAL_FALLBACK_LOCKTIME must be used. If PSBT_GLOBAL_FALLBACK_LOCKTIME is not provided, then it is assumed to be 0.

If one or more inputs have a PSBT_IN_REQUIRED_TIME_LOCKTIME or PSBT_IN_REQUIRED_HEIGHT_LOCKTIME, then the field chosen is the one which is supported by all of the inputs. This can be determined by looking at all of the inputs which specify a locktime in either of those fields, and choosing the field which is present in all of those inputs. Inputs not specifying a lock time field can take both types of lock times, as can those that specify both. The lock time chosen is then the maximum value of the chosen type of lock time.

If a PSBT has both types of locktimes possible because one or more inputs specify both PSBT_IN_REQUIRED_TIME_LOCKTIME and PSBT_IN_REQUIRED_HEIGHT_LOCKTIME, then locktime determined by

looking at the PSBT_IN_REQUIRED_HEIGHT_LOCKTIME fields of the inputs must be chosen.[2]

Unique Identification

PSBTv2s can be uniquely identified by constructing an unsigned transaction given the information provided in the PSBT and computing the transaction ID of that transaction. Since PSBT_IN_SEQUENCE can be changed by Updaters and Combiners, the sequence number in this unsigned transaction must be set to 0 (not final, nor the sequence in PSBT_IN_SEQUENCE). The lock time in this unsigned transaction must be computed as described previously.

Roles

PSBTv2 introduces new roles and modifies some existing roles.

Creator

In PSBTv2, the Creator initializes the PSBT with 0 inputs and 0 outputs. The PSBT version number is set to 2. The Creator should also set PSBT_GLOBAL_FALLBACK_LOCKTIME. If the Creator is not also a Constructor and will be giving the PSBT to others to add inputs and outputs, the PSBT_GLOBAL_TX_MODIFIABLE field must be present and the Inputs Modifiable and Outputs Modifiable flags set appropriately; moreover, the transaction version number must be set to at least 2. [3] If the Creator is a Constructor and no inputs and outputs will be added by other entities, PSBT_GLOBAL_TX_MODIFIABLE may be omitted.

Constructor

This Constructor is only present for PSBTv2. Once a Creator initializes the PSBT, a constructor will add inputs and outputs. Before any input or output may be added, the constructor must check the PSBT_GLOBAL_TX_MODIFIABLE field. Inputs may only be added if the Inputs Modifiable flag is True. Outputs may only be added if the Outputs Modifiable flag is True.

When an input or output is added, the corresponding PSBT_GLOBAL_INPUT_COUNT or PSBT_GLOBAL_OUTPUT_COUNT must be incremeted to reflect the number of inputs and outputs in the PSBT. When an input is added, it must

[2] **Why choose the height based locktime?** In the event of a tie for the locktime type, signers need to be able to know which locktime to use as their signatures will commit to the locktime in the transaction, so choosing the wrong one will result in an invalid transaction. Height based locktime is preferred over time based as Bitcoin's unit of time is the block height, so a height makes more sense in the context of Bitcoin.

[3] **Why does the transaction version number need to be at least 2?** The transaction version number is part of the validation rules for some features such as OP_CHECKSEQUENCEVERIFY. Since it is backwards compatible, and there are other ways to disable those features (e.g. through sequence numbers), it is easier to require transactions be able to support these features than to try to negotiate the transaction version number.

have PSBT_IN_PREVIOUS_TXID and PSBT_IN_OUTPUT_INDEX set. When an output is added, it must have PSBT_OUT_VALUE and PSBT_OUT_OUTPUT_SCRIPT set. If the input has a required timelock, Constructors must set the requisite timelock field. If the input has a required time based timelock, then PSBT_IN_REQUIRED_TIME_TIMELOCK must be set If the input has a required height based timelock, then PSBT_IN_REQUIRED_HEIGHT_TIMELOCK must be set. If an input has both types of timelocks, then both may be set. In some cases, an input that can allow both types, but a particular branch supporting only one type of timelock will be taken, then the type of timelock that will be used can be the only one set.

If an input being added specifies a required time lock, then the Constructor must iterate through all of the existing inputs and ensure that the time lock types are compatible. Additionally, if during this iteration, it finds that any inputs have signatures, it must ensure that the newly added input does not change the transaction's locktime. If the newly added input has an incompatible time lock, then it must not be added. If it changes the transaction's locktime when there are existing signatures, it must not be added.

If the Has SIGHASH_SINGLE flag is True, then the Constructor must iterate through the inputs and find the inputs which have signatures that use SIGHASH_SINGLE. The same number of inputs and outputs must be added before those inputs and their corresponding outputs.

A Constructor may choose to declare that no further inputs and outputs can be added to the transaction by setting the appropriate bits in PSBT_GLOBAL_TX_MODIFIABLE to 0 or by removing the field entirely.

A single entity is likely to be both a Creator and Constructor.

Updater

For PSBTv2, an Updater can set the sequence number.

Signer

For PSBTv2s, a signer must update the PSBT_GLOBAL_TX_MODIFIABLE field after signing inputs so that it accurately reflects the state of the PSBT. If the Signer added a signature that does not use SIGHASH_ANYONECANPAY, the Input Modifiable flag must be set to False. If the Signer added a signature that does not use SIGHASH_NONE, the Outputs Modifiable flag must be set to False. If the Signer added a signature that uses SIGHASH_SINGLE, the Has SIGHASH_SINGLE flag must be set to True.

Transaction Extractor

For PSBTv2s, the transaction is constructed using the PSBTv2 fields. The lock time for this transaction is determined as described in the Determining Lock

Time section. The Extractor should produce a fully valid, network serialized transaction if all inputs are complete.

Backwards Compatibility

PSBTv2 shares the same generic format as PSBTv0 as defined in BIP 174. Parsers for PSBTv0 should be able to deserialize PSBTv2 with only changes to support the new fields.

However PSBTv2 is incompatible with PSBTv0, and vice versa due to the use of the PSBT_GLOBAL_VERSION. This incompatibility is intentional so that PSBT_GLOBAL_UNSIGNED_TX could be removed in PSBTv2. However it is possible to convert a PSBTv2 to a PSBTv0 by creating an unsigned transaction from the PSBTv2 fields.

Test Vectors

The following are invalid PSBTs:

- Case: PSBTv0 but with PSBT_GLOBAL_VERSION set to 2.
  - Bytes in Hex:
    `70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408`
  - Base64 String:
    `cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAry8AAAA`
- Case: PSBTv0 but with PSBT_GLOBAL_TX_VERSION.
  - Bytes in Hex:
    `70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408`
  - Base64 String:
    `cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAry8AAAA`
- Case: PSBTv0 but with PSBT_GLOBAL_FALLBACK_LOCKTIME.
  - Bytes in Hex:
    `70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408`
  - Base64 String:
    `cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAry8AAAA`
- Case: PSBTv0 but with PSBT_GLOBAL_INPUT_COUNT.
  - Bytes in Hex:
    `70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408`
  - Base64 String:
    `cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAry8AAAA`
- Case: PSBTv0 but with PSBT_GLOBAL_OUTPUT_COUNT.
  - Bytes in Hex:
    `70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408`
  - Base64 String:
    `cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAry8AAAA`

- Case: PSBTv0 but with PSBT_GLOBAL_TX_MODIFIABLE.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_IN_PREVIOUS_TXID.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_IN_OUTPUT_INDEX.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_IN_SEQUENCE.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_IN_REQUIRED_TIME_LOCKTIME.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_IN_REQUIRED_HEIGHT_LOCKTIME.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_OUT_AMOUNT.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv0 but with PSBT_OUT_SCRIPT.
  - Bytes in Hex:
    70736274ff01007102000000010b0ad921419c1c8719735d72dc739f9ea9e0638d1fe4c1eef0f994408
  - Base64 String:
    cHNidP8BAHECAAAAAQsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/IAAAAAAD+////AgAIry8AAAA
- Case: PSBTv2 missing PSBT_GLOBAL_INPUT_COUNT.
  - Bytes in Hex:

70736274ff01020402000000010304000000000105010201fb040200000000010052020000001c1aa2
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEFAQIB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG4

- Case: PSBTv2 missing PSBT_GLOBAL_OUTPUT_COUNT.
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101fb040200000000010052020000001c1aa2
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG4

- Case: PSBTv2 missing PSBT_IN_PREVIOUS_TXID.
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102 01fb040200000000010052020000
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/4

- Case: PSBTv2 missing PSBT_IN_OUTPUT_INDEX.
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102 01fb040200000000010052020000
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/4

- Case: PSBTv2 missing PSBT_OUT_AMOUNT.
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102 01fb040200000000010052020000
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/4

- Case: PSBTv2 missing PSBT_OUT_SCRIPT.
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102 01fb040200000000010052020000
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/4

- Case: PSBTv2 with PSBT_IN_REQUIRED_TIME_LOCKTIME less than 500000000.
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201fb040200000000010052020000001c1aa256e214
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG41Fdc

- Case: PSBTv2 with PSBT_IN_REQUIRED_HEIGHT_LOCKTIME greater than or equal to 500000000.
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201fb040200000000010052020000001c1aa256e214
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG41Fdc

The following are valid PSBTs

- Case: 1 input, 2 output PSBTv2, required fields only.
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201fb040200000000010e200b0ad921419c1c871973
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEOIAsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/
- Case: 1 input, 2 output updated PSBTv2.
  - Bytes in HEx:
    70736274ff01020402000000010401010105010201fb040200000000010052020000000001c1aa256e214
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG41Fdc
- Case: 1 input, 2 output updated PSBTv2, with PSBT_IN_SEQUENCE.
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201fb040200000000010052020000000001c1aa256e214
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEAUgIAAAABwaolbiFLlqGCL5PeQr/ztfP/jQUZMG41Fdc
- Case: 1 input, 2 output updated PSBTv2, with PSBT_IN_SEQUENCE,
  and all locktime fields
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102010fb040200000000010052020000
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/4C
- Case: 1 input, 2 output updated PSBTv2, with Inputs Modifiable Flag
  (bit 0) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060101fb040200000000010052020000000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEBAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE
- Case: 1 input, 2 output updated PSBTv2, with Outputs Modifiable Flag
  (bit 1) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060102fb040200000000010052020000000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgECAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE
- Case: 1 input, 2 output updated PSBTv2, with Has SIGHASH_SINGLE
  Flag (bit 2) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060104fb040200000000010052020000000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEEAfsEAgAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE
- Case: 1 input, 2 output updated PSBTv2, with an undefined flag (bit 3)
  of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:

70736274ff01020402000000010401010105010201060108 01fb0402000000000100520200000001c1a

  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEIAfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with both Inputs Modifiable Flag (bit 0) and Outputs Modifiable Flag (bit 1) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060103 01fb0402000000000100520200000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEDAfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with both Inputs Modifiable Flag (bit 0) and Has SIGHASH_SINGLE Flag (bit 2) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060105 01fb0402000000000100520200000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEFAfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with both Outputs Modifiable Flag (bit 1) and Has SIGHASH_SINGLE FLag (bit 2) of PSBT_GLOBAL_TX_MODIFIABLE set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060106 01fb0402000000000100520200000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEGAfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with all defined PSBT_GLOBAL_TX_MODIFIABLE flags set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060107 01fb0402000000000100520200000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgEHAfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with all possible PSBT_GLOBAL_TX_MODIFIABLE flags set
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201060 1ff01fb0402000000000100520200000001c1a
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIBBgH/AfsEAgAAAAABAFICAAAAAcGqJW4hS5ahgi+T3kK/87Xz/40FGTE

- Case: 1 input, 2 output updated PSBTv2, with all PSBTv2 fields
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010101050102010601 07 01fb0402000000000100 5
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQEBBQECAQYBBwH7BAIAAAAAAQBSAgAAAAHBqiVuIUuWoYIvk95Cv/C

The following tests are the timelock determination algorithm.

The timelock for the following PSBTs should be computed to be 0:

- Case: No locktimes specified
  - Bytes in Hex:
    70736274ff01020402000000010401010105010201fb040200000000010e200b0ad921419c1c8719735
  - Base64 String:
    cHNidP8BAgQCAAAAAQQBAQEFAQIB+wQCAAAAAAEOIAsK2SFBnByHGXNdctxzn56p4GONH+TB7vD5lECEgV/
- Case: Fallback locktime of 0
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758db
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA

The timelock for the following PSBTs should be computed to be 10000:

- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000, Input 2 has no locktime fields
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758db
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA
- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000, Input 2 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 9000
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758db
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA
- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000, Input 2 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 9000 and PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048460
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758db
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA
- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000 and PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048459, Input 2 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 9000 and PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048460
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758db
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA

The timelock for the following PSBTs should be computed to be 1657048460:

- Case: Input 1 has PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048459, Input 2 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 9000 and PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048460
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758d
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA
- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000 and PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048459, Input 2 has PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048460
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758d
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA
-
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758d
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA

The timelock for the following PSBTs cannot be computed:

- Case: Input 1 has PSBT_IN_REQUIRED_HEIGHT_LOCKTIME of 10000, Input 2 has PSBT_IN_REQUIRED_TIME_LOCKTIME of 1657048460
  - Bytes in Hex:
    70736274ff01020402000000010304000000000104010201050101fb040200000000010e200f758d
  - Base64 String:
    cHNidP8BAgQCAAAAAQMEAAAAAAEEAQIBBQEBAfsEAgAAAABDiAPdY2/vU2nwWyKMwnDyB4RAPVh6mRttbA

Rationale

Reference implementation

The reference implementation of the PSBT format is available at https://github.com/achow101/bitcoin/tree/psbt2.