Resolution:2560×1440(2K)
atomicals-js:c405bee22e4cb70e3951ad747fa3ed9ae2dcd745, https://github.com/bitcoinjs/bitcoinjs-lib/tree/v6.1.5

Entries

run
atomicals-js/lib/commands/mint-interactive-dft-command.ts

```
const atomicalBuilder = new AtomicalOperationBuilder({
atomicalBuilder.setBitworkCommit(mint_bitworkc);
atomicalBuilder.setBitworkReveal(mint_bitworkr);
atomicalBuilder.addOutput({
const result = await atomicalBuilder.start(this.fundingWIF);
```

atomical-operation-builder

run
atomicals-js/lib/utils/atomical-operation-builder.ts

```
for (let i = 0; i < concurrency; i++) {
worker.on("message", (message: WorkerOut) => {
    let psbtStart = new Psbt({ network: NETWORK });
    psbtStart.addInput({
    psbtStart.addOutput({
    this.addCommitChangeOutputIfRequired(
    psbtStart.signInput(0, fundingKeypair.tweakedChildNode);
    psbtStart.finalizeAllInputs();
    const interTx = psbtStart.extractTransaction();
    const rawtx = interTx.toHex();
    AtomicalOperationBuilder.finalSafetyCheckForExcessiveFee(
    if (!this.broadcastWithRetries(rawtx)) {
const messageToWorker = {
worker.postMessage(messageToWorker);
workers.push(worker);
```

miner-worker

parentPort.on("message"
atomicals-js/lib/utils/miner-worker.ts

```
do {
    Math.min(sequence + 10000, MAX_SEQUENCE)
    let psbtStart = new Psbt({ network: NETWORK });
    psbtStart.addInput({
    psbtStart.addOutput(fixedOutput);
    psbtStart.signInput(0, fundingKeypair.tweakedChildNode);
    psbtStart.finalizeAllInputs();
    prelimTx = psbtStart.extractTransaction();
    const checkTxid = prelimTx.getId();
    if (
        workerPerformBitworkForCommitTx &&
        hasValidBitwork(
    sequence++;
parentPort!.postMessage({
```

psbt

signInput
bitcoinjs-lib/ts_src/psbt.ts

```
if (isTaprootInput(input)) {
    return this._signTaprootInput(
return this._signInput(inputIndex, keyPair, sighashTypes);
```

_signInput
bitcoinjs-lib/ts_src/psbt.ts

```
const { hash, sighashType } = getHashAndSighashType(
signature: bscript.signature.encode(keyPair.sign(hash),
sighashType),
this.data.updateInput(inputIndex, { partialSig });
```