

CCIP Manual Execution

Prerequisites

Read the CCIP [Concepts](#) and [Architecture](#) pages to understand all the concepts discussed on this page.

In general, messages are successfully delivered and processed via CCIP as described in the [Architecture](#) page. CCIP has a built-in gas bumping mechanism called Smart Execution to ensure that messages are reliably delivered to the destination blockchain. Read the [CCIP Execution latency](#) section to learn more about Smart Execution. However, some exceptional conditions might require users to manually execute the transaction on the destination blockchain:

- The receiver contract on the destination blockchain reverted due to an unhandled exception such as a logical error.
- The receiver contract on the destination blockchain reverted due to the gas limit being insufficient to execute the triggered function (Note: The gas limit value is set in the [extraArgs](#) param of the message).
- The message could not be executed on the destination chain within CCIP's Smart Execution time window, which is currently set to 8 hours. This could happen, for example, during extreme network congestion and resulting gas spikes.

The flowchart below displays the process of a cross-chain transaction, detailing the steps involved in the manual execution:

CCIP execution

1. A sender contract or EOA initiates a CCIP message on the source blockchain.
2. [CCIP Committing DON](#) awaits [finality](#) on the source blockchain.
3. Post finality, the [CCIP Committing DON](#) assembles a batch of eligible transactions, computes a Merkle root, and records it to the [CommitStore contract](#) on the destination blockchain.
4. After successful verification, the [Risk Management Network](#) blesses the committed Merkle root.
5. After the committed Merkle root is blessed, the [CCIP Executing DON](#) proceeds with the execution on the destination blockchain:
6. During the Smart Execution time window, the Executing DON assesses and, if necessary, adjusts the gas price (a process known as gas bumping) to facilitate successful transaction execution on the destination blockchain as part of the Smart Execution process.
7. If the Smart Execution time window has passed and the CCIP message has still not been executed, then Manual Execution is enabled. See the [Manual execution](#) section for more details.
8. The execution on the destination blockchain works as follows:
9. If the message involves token transfers, the tokens are first transferred to the receiver.
10. If the receiver is an EOA, then this transaction is considered complete with no further processing.
11. If the receiver is a smart contract, the [ccipReceive](#) function is invoked after the token transfer. The `ccipReceive` function processes the CCIP message and any user-specified logic in the receiver contract. The execution of the CCIP message is atomic (all or none). If the `ccipReceive` function successfully executes, then all aspects of the transaction are complete, and there is no revert. If, however, there is an issue in the receiver execution due to insufficient gas limit or unhandled exceptions, the attempted token transfer will also revert. The transaction then becomes eligible for manual execution.
12. If the message does not involve token transfers, only arbitrary messaging, and the receiver execution fails due to gas limits or unhandled exceptions, the transaction becomes eligible for manual execution.

Manual execution

As described above, a CCIP message becomes eligible for manual execution for various reasons. Manual execution means that a user has to manually trigger the execution of the destination transaction after the issue that caused manual execution has been resolved.

When a CCIP message is eligible for manual execution, the [CCIP explorer](#) shows the following information:

- Ready for manual execution status
- The possibility to override the gas limit and a `Trigger Manual Execution` button

Depending on the situation, you can take one of the following steps:

- Insufficient gas limit: The executor can connect their wallet, override the gas limit parameter, increase the gas limit for this particular execution, and trigger a manual execution. If this new gas limit override is sufficient, the transaction will go through successfully. Note: This gas limit override applies only to this manually executed transaction.
- Unhandled exception (logical error) in the receiver contract: If the receiver contract is [upgradeable](#), developers must

correct the logic, re-deploy the logic contract, and then manually execute the same transaction. If the receiver contract is not upgradeable, developers must deploy a new receiver contract, and then users can send a new CCIP message. Non-upgradable contracts will not benefit from manual execution. **Note:** Always make sure to test your smart contracts thoroughly. As a best practice, implement fallback mechanisms in the CCIP receiver contracts to manage unhandled exceptions gracefully. Read the [Defensive example](#) to learn more.

- If the issue was due to extreme gas spikes or network conditions, and CCIP was not able to successfully transfer the message despite gas bumping for the entire duration of the Smart Execution time window (currently 8 hours), then you can try manual execution.

When manual execution is initiated, a Merkle proof is generated for the message to be executed. During execution, the CCIP explorer submits the Merkle proof and the new gas limit (if the initial failure was due to a low gas limit). This Merkle proof is verified by the [OffRamp contract](#) against the Merkle root in the [CommitStore contract](#), and that was blessed by the [Risk Management Network](#). This mirrors the automated execution performed by the [CCIP Executing DON](#), with the addition that the execution is resubmitted using the gas limit you provide.

Frequently asked questions

1. Can anyone execute a transaction on the CCIP explorer even if they are not the initiator of the transaction?

Yes, any EOA can manually execute a CCIP message that is eligible for manual execution. However, the executing account must have sufficient native gas tokens (such as ETH on Ethereum or MATIC on Polygon) to cover the gas costs associated with the delivery of the CCIP message. 2. If a user sends multiple messages and the first message isn't successfully delivered and goes into a manual execution mode, does that mean all subsequent messages from the user will also be stuck?

It depends. If a message goes into manual execution mode due to receiver errors (unhandled exceptions or gas limit issues), subsequent messages don't get automatically blocked, unless they would encounter the same error. However, suppose a message goes into manual execution mode after the Smart Execution time window expires (currently 8 hours). In that case, subsequent messages must wait for the first message to be processed to maintain the default sequence. 3. If the maximum gas limit is 2M (2,000,000) on mainnet, but it turns out that the destination blockchain requires more than that, will an override of > 2M still work?

Yes, but only for this execution. This works because the gas limit for this execution instance isn't passing through the CCIP validation for the gas limit, for which the CCIP executing DON pays the gas. However, if you consistently need more than 2M for your use case, please reach out to us via this [contact form](#). 4. Will Chainlink Labs reimburse us for manual execution fees?

Since most manual execution situations are due to insufficient gas limit or an unhandled exception in the receiver contract, Chainlink Labs does not reimburse these fees. If you are a dApp developer, please ensure you test thoroughly to avoid manual executions to the extent possible. 5. Do I have to manually execute via the CCIP explorer? Are there any other ways to do this?

The CCIP explorer provides the easiest way to execute manually. It handles all the complexity of submitting the Merkle proof needed for successful transaction execution. 6. How do I know if my receiver error is due to a gas limit issue or an unhandled exception?

If you see a `ReceiverError` with a revert reason as empty (0x), this is likely due to a gas limit issue. You can look at the transaction trace for the destination transaction on a tool such as [Tenderly](#), and you will likely see a `gas` reason mentioned in such cases. Determine the gas limit that would work for your transaction and manually override it. Read the [manual execution](#) tutorial to analyze an example of an exception due to a low gas limit. 7. How can I write contracts that avoid manual execution situations in the first place?

- Test thoroughly to ensure logical conditions for all paths are gracefully handled in your receiver contract.
- Set a gas limit that works for complex execution paths, not just the simplest ones. Read the [best practices](#) for gas estimation.
- Refer to the [Defensive example](#) tutorial for an example of how to design a programmable token transfer that handles errors gracefully.
- My transaction meets one of the conditions to trigger manual execution, but I do not see this option on the CCIP explorer. How am I supposed to execute this manually?

This should not happen, but in the unlikely scenario that this does happen, please reach out directly to Chainlink Labs by filling out this [form](#). Make sure to include the CCIP Message ID, your preferred contact details, and the issue you faced in your communication.