

Memo Program

The Memo program is a simple program that validates a string of UTF-8 encoded characters and verifies that any accounts provided are signers of the transaction. The program also logs the memo, as well as any verified signer addresses, to the transaction log, so that anyone can easily observe memos and know they were approved by zero or more addresses by inspecting the transaction log from a trusted provider.

Background

Solana's programming model and the definitions of the Solana terms used in this document are available at:

- <https://docs.solana.com/apps>
- <https://docs.solana.com/terminology>

Source

The Memo Program's source is available on [GitHub](#)

Interface

The on-chain Memo Program is written in Rust and available on crates.io as [spl-memo](#) and [docs.rs](#).

The crate provides `abuild_memo()` method to easily create a properly constructed Instruction.

Operational Notes

If zero accounts are provided to the signed-memo instruction, the program succeeds when the memo is valid UTF-8, and logs the memo to the transaction log.

If one or more accounts are provided to the signed-memo instruction, all must be valid signers of the transaction for the instruction to succeed.

Logs

This section details expected log output for memo instructions.

Logging begins with entry into the program: `Program MemoSq4gqABAXKb96qnH8TysNcWxMyWCqXgDLGmfcHr invoke [1]`

The program will include a separate log for each verified signer: `Program log: Signed by`

Then the program logs the memo length and UTF-8 text: `Program log: Memo (len 4): " "`

If UTF-8 parsing fails, the program will log the failure point: `Program log: Invalid UTF-8, from byte 4`

Logging ends with the status of the instruction, one of: `Program MemoSq4gqABAXKb96qnH8TysNcWxMyWCqXgDLGmfcHr success` `Program MemoSq4gqABAXKb96qnH8TysNcWxMyWCqXgDLGmfcHr failed: missing required signature for instruction` `Program MemoSq4gqABAXKb96qnH8TysNcWxMyWCqXgDLGmfcHr failed: invalid instruction data`

For more information about exposing program logs on a node, head to the [developer docs](#)

Compute Limits

Like all programs, the Memo Program is subject to the cluster's [compute budget](#). In Memo, compute is used for parsing UTF-8, verifying signers, and logging, limiting the memo length and number of signers that can be processed successfully in a single instruction. The longer or more complex the UTF-8 memo, the fewer signers can be supported, and vice versa.

As of v1.5.1, an unsigned instruction can support single-byte UTF-8 of up to 566 bytes. An instruction with a simple memo of 32 bytes can support up to 12 signers. [Edit this page](#) [Previous](#) [Token Upgrade Program](#) [Next](#) [Name Service](#) * [Background](#) * [Source](#) * [Interface](#) * [Operational Notes](#) * * [Logs](#) * * [Compute Limits](#)