midnight

Testnet release notes

October 1, 2024

Midnight is a regulatory-friendly data protection-based blockchain that safeguards sensitive commercial and personal data, protecting fundamental freedoms of association, commerce, and expression for developers, companies, and individuals. Midnight utilizes a novel data protection-first programming model and zero-knowledge (ZK) proofs while operating as a Cardano partner chain.

Overview

Midnight `testnet` is the evolution from our developer-centric `devnet` version to the upcoming public `mainnet` release; it is being made broadly available to the independent and partner developer communities. We solicit feedback about our technology and development progress.

These release notes present the core features of `testnet` along with some of the known issues our team has mapped and is working to resolve. We will use Midnight's Discord channel (discord.gg/midnightnetwork) to update the community on these and any new issues we find.

NOTE: The Midnight documentation (docs.midnight.network) has been updated to reflect `testnet`. The `devnet` will be sunset on a schedule to be shared via Discord.

Impacted users

These release notes apply to all Midnight users.

Impacted components

Component	Version	Change/impact
Blockchain	0.6.3	Stabilization upgrade, bug fixes
Indexer	2.2.1	Stabilization updates and hard fork improvements

midnight

Component	Version	Change/impact
Wallet (Lace)	1.2.2 (App) 3.6.3 (Engine) 3.4.21 (API)	Support for `testnet` (not backwards compatible with `devnet`)
DApp connector	1.2.1	Common prover and balance methods
Ledger	3.0.3	Ledger now uses halo2 proving system, supporting SNARK upgradeability
Compact compiler	0.18.2	Accompanying Compact language updates
Compact	0.10.1	Syntax changes, namespace changes, and bug fixes to make the language more analogous to TypeScript and more user-friendly
Visual Studio plugin	0.2.12	Accompanying Compact language updates
Mn.JS	0.2.4	API changes to support snark upgrade
Faucet	0.8.0	`testnet` support (not backward compatible with `devnet`). Improved performance and UX.

Additional resources

Developer support

The Developer Relations team can be reached through Midnight's Discord channel (discord.gg/midnightnetwork), where further communication about community engagement will be made.

Documentation

Midnight documentation (docs.midnight.network) is regularly updated. Please address suggestions, questions, and requests to the Developer Relations team on Discord.

Social channels

- Midnight on Twitter/X: https://twitter.com/MidnightNtwrk
- Midnight on LinkedIn: https://www.linkedin.com/showcase/midnight-ntwrk/

midnight

What is in this release?

- SNARK Upgradeability: Midnight's tech stack is now capable of updating the proving without resetting the chain. Contracts can be easily updated to accommodate proving system changes, bug fixes, and performance enhancements. As of now, there is no need to move contract funds if maintenance authority is set.
- Midnight Block Producer Guide and System Requirements: SPOs may become block producers for the public Midnight Testnet to confirm they meet the hardware requirements to maintain consistent node performance and contribute effectively to Midnight's security and stability. These requirements help maintain a level playing field for all participants and promote a healthy ecosystem.
- Stake Pool Operator (SPO) Registration and Block Production, A new consensus mechanism was designed to enhance network decentralization while maintaining a degree of centralized control. It has been fully integrated into the block stack and is now available for external evaluation, allowing developers and the community to test and provide feedback.
- Chain Hard Fork Capability: Midnight internal developers are now able to publish backward incompatible changes without resetting the chain. DApp and Wallet developers can accommodate these changes with minimal inconvenience to their users. Finally, it provides a mechanism for SPOs and Block Producers to upgrade to the new consensus rules.
- Halo2 as a proving system instead of Plonk: Halo2, a popular library for zero-knowledge proofs, offers advantages like custom gates, lookup arguments, and universal SNARKs. Midnight uses a modified version of Halo2, supporting efficient recursion and pairing-friendly curves.
- SPO nodes automatically vote for runtime upgrades: Validators should be able to signal to one another that they have upgraded to a client version compatible with the latest ledger version and the upcoming runtime upgrade. This signaling is achieved by broadcasting a specific message to the network, indicating the client version and a unique identifier for the runtime upgrade. Only validators with compatible clients can participate in the upgrade process, ensuring a smooth transition to the new runtime.

midnight

Bug fixes

Item	Issue/bug
PM-8523	[Indexer] Missing block height number validation before querying. This should not impact users.
PM-8135	[Faucet] Error messages are ambiguous in case of insufficient balance or invalid address. Users should ensure a valid wallet address is entered (e.g. by copy and paste from Lace Wallet) and try again after a few minutes.
PM-7843	[Welcome DApp] DApp halts if invalid contract address is set. Users should end the current CLI DApp instance and start a new one ensuring to use a valid Contract Address.

PM-7593	When the wallet engine connects to the indexer, it can fail with error message 'java.security.SecureRandom is not supported on this platform because it provides neither `crypto.getRandomValues` nor Node.js' 'crypto' module.' The users should apply the following workaround:
	Unset
	`import { webcrypto } from "node:crypto";`
	`// @ts-ignore`
	`global.crypto = webcrypto;`

Enhancements

Ledger

- Ledger now supports the Halo2 proving system.
- Ledger supports upgradeable contracts to support SNARK upgradeability.
- Ledger parameters has been added.
- Added a mechanism for block rewards.
- Moved the network ID variable from a state variable to a parameter of serialization functions.
- Switched to SHA-256 for our persistent hash function.
- Bug fixes and improvements.

Compact

- Ledger declaration syntax and namespace changes:
  - Ledger fields are now declared individually with the keyword `ledger`.
  - There is no need to prefix a ledger field with `ledger.` since ledger field declarations are now top-level.
  - Kernel operations now require `kernel.` prefix instead of `ledger.` prefix.
  - Ledger field update shortcuts are now statements.
- Unsigned integer sizes can now be generic and can take either numeric literal or a generic parameter name.
- The static typing of subtraction now provides a more precise bound on the result type.
- Compact now matches the precedence of relational operators in TypeScript and JavaScript.
- Due to the lack of direct analog in TypeScript, Generics can no longer be parameterized over generics.
- The identifier `contract` is now a reserved word due to adding support for a new in-progress feature that allows contracts to call other contracts.

More details on these changes, why we made them, and how you can fix your code can be found in the Compact 0.10.1 notes.

CompactC

- CompactC can now be run with the command `compactc` without the user having to set their `$PATH` environment.

MN.JS

- Added convenience functions for performing SNARK upgrades of contracts.
- Added convenience functions for creating unproven transactions.
- Added convenience functions for submitting transactions directly.
- Replaced function parameters with configuration objects at the main SDK entry points.
- Improved type inference for contracts and API clients.

- Added support for storing and retrieving contract maintenance authority signing keys.
- Introduced an explicit distinction between public and private data in the transaction model.

Blockchain

- Session length increased from 6 minutes to 2 hours to improve stability.
- Simplified configuration for Midnight node with pre-configured settings for each network, reducing setup time and the risk of configuration errors.
- Users are now able to implement, visualize, and transact with different shielded token types, providing enhanced privacy and flexibility for various use cases.
- Users can now view initial faucet funding transactions directly within the genesis block using polkadot.js, improving transparency.
- Docker Compose scripts now available for SPOs that would prefer not to use Kubernetes.
- Several RPC methods now return `json` rather than `string`.
- The Midnight node image is now Debian-based for better compatibility and easier management.
- Node Version of block author now included in block header.
- Full set of substrate node CLI commands available.
- Node now monitors the available disk space and will gracefully shut down if that value drops below the threshold.
- Increased stability for all components.

Indexer

- Enhanced stability and performance through updated dependencies, optimized database schema, and improved logging.
- Implemented support for the upcoming network upgrade, enabling seamless upgrades, providing flexibility for developers, and better UX for end-users.
- Enhanced compatibility with the latest Halo2 proving system, allowing for smoother transactions and reduced error rates.

Wallet

- Midnight-Lace
    - Enhanced stability and performance improvements.
    - Updated to the latest version of Wallet Engine for improved security, compatibility, and feature support.
- Wallet Engine
    - The Halo2 proving system has been integrated, enabling more efficient proving transaction performance.
    - Implemented support for network upgrades, enabling seamless upgrades, providing flexibility for developers, and better UX for end-users.
    - Network ID is now a required build parameter to ensure proper network configuration and prevent transactions from ending up in the wrong network.
    - Introduced option to omit transaction history via build parameter. This can be used to reduce storage requirements and improve performance in certain use cases.

Known issues

Item	Issue	Impact	Workaround
PM-8694	[Wallet] Native tokens missing after resyncing or restoring wallet. When that happens, users are required to mint new tokens.	Native Tokens can go missing in Midnight Lace Wallet when restoring a wallet using the same seed phrase on multiple devices. The problem occurs when transferring the wallet setup (using the seed phrase) to a different browser/computer.	Workaround: Keep Midnight-Lace and its associated seed phrase on the same browser/computer to preserve native tokens.
PM-12387	Partner-chain-cli missing deregister command	Deregister must be performed by invoking the sidechain-main-cli that's packaged with partner-chain-cli.	Invoke "./sidechain-main-cli deregister" to perform deregistration.
PM-12428	Unexpected "Not sufficient funds to balance token" message	SNARK upgrades initiated using Midnight.js will fail intermittently.	Increase the delay between each SNARK upgrade attempt.
PM-12365	[wallet] Timed out trying to connect if an invalid network ID provided.	Wallet will not connect.	Verify if the network ID, indexer, and node URLs are correct.

Item	Issue	Impact	Workaround
PM-3974	Retain the ability to develop smart contracts when a hard fork happens.	MN.JS won't support HF at this time.	Launch HF capability on all components. MN.jS updates will happen on the next release.