Release notes - devnet v0.1.0

February 19, 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

This devnet release is Midnight's first public alpha version, which is being made broadly available for the developer community. We aim to solicit the community's feedback about Midnight's technology and development progress. It is one of many iterations of the functionality on the roadmap to Midnight mainnet.

These release notes present the core features of Midnight that are being made available at this time and list some of the known issues our team has mapped and is working to resolve. We will use Midnight's Discord channel to update the community on resolutions and any new issues we find.

NOTE: the previous version of Devnet, 0.0.2, will be deactivated after this new devnet release, and Midnight's documentation (docs.midnight.network) has been updated to reflect the new version. Communication of Devnet 0.0.2 sunset will be issued via Midnight's Discord.

Impacted users

Devnet v0.1.0 release notes apply to all Midnight devnet users.

Impacted components

Component	Version	Change/impact
Blockchain	0.2.0	Stabilization upgrade, bug fixes
Pub-sub indexer	1.0.5	Stabilization upgrade
Wallet (Lace)	1.1.0 (App) 3.5.5 (Engine) 3.3.1 (API)	Support for Midnight Native Shielded Tokens
DApp connector	1.1.0	Common prover and balance methods
Proof server	2.0.7	Bug fixes
Compact compiler	0.9.2	Support to Midnight Native Shielded Tokens
Visual Studio plugin	0.2.12	Accompanying Compact language updates
Faucet	0.6.1	Initial release

Additional resources

Developer support

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

Documentation

Midnight documentation is accessible online (https://docs.midnight.network/) and is regularly updated. In case of suggestions, questions, or requests, please contact 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/

What is in this release?

- This is an alpha version of Midnight, which is open to all developers.
- Build data-protection smart contracts using the Compact domain-specific language (DSL) and Compact compiler. Generate JavaScript, cryptographic materials, and *circuit* descriptions needed by the proof to create the ZK proofs that enforce the terms of a smart contract while shielding the users' private data.
- Store, manage, and interact with Midnight assets and decentralized applications (DApps) directly within the Google Chrome web browser using a Midnight alpha version of the popular Lace wallet exclusive for Devnet.
- View and manage listed Midnight Native Tokens in their Midnight Lace Wallet.
- Pay for transactional costs and move value peer-to-peer using shielded tDUST tokens (tDUST is a test token used for Midnight devnet testing purposes only).
- Create ZK proofs and perform ZK computations using the local proof generator software to facilitate the submission of transactions and proof data from clients.
- Index the Midnight blockchain data to support wallet and DApp functionality using the pub-sub indexer to query data directly from the ledger.
- Write Midnight DApps in TypeScript and Compact DSL code supported by the Visual Studio Code plugin.

Bug fixes

Item	Issue/bug
PM-7595	The wallet history shows only transaction fees to preserve the confidentiality of the transaction. Other visualization methods will be made available in the future.
PM-7815 PM-7830 PM-7826 PM-7790	The following error messages or behaviors of the wallet require the user to resync the wallet by using the resync button next to the syncing status in the menu: ● The wallet locks funds when an error happens after a transaction is balanced/built but before it is submitted. ● When a 'Not sufficient funds' error appears, and no errors are present in the node logs, it might be the wallet locking funds due to an error before submitting a transaction. ● When a ‘Wallet was not synced. Connection was lost’ error appears, it might be because the wallet is locking funds due to an error before submitting a transaction. ● The wallet locks funds when a valid transaction is submitted but fails to execute in the node.
PM-7832	The wallet displays a popup indicating that it cannot ‘fetch ADA price’. When this occurs, the user can close this message and proceed further. Ada is not a token available for use on the Midnight devnet.
PM-7820 PM-7817	The following error messages or behaviors of the wallet require the user to create a new wallet: ● RuntimeError: unreachable at wasm ● Not sufficient funds to balance token
PM-7594	When the headless wallet is stopped, it might show an error saying ‘node:events:495 throw er; // Unhandled 'error''. It's an internal abrupt close of connection, but nothing is wrong. Users can ignore this message.

PM-7593	When the headless wallet connects to the pub-sub indexer, it might 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

- Increased stability for all components.
- New blocks are produced every 6 seconds.
- Users are now able to implement, visualize and transact with different shielded token types.

Known issues

Item	Issue
PM-8694	[Wallet] Native tokens missing after resyncing or restoring wallet. When that happens, users are required to mint new tokens.
PM-8523	[Pubsub] Missing block height number validation before querying. This should not impact users.
PM-8476	[Wallet] Deploying contract error after increasing wallet balance. In case of insufficient balance, users should increase wallet balance, resync their wallet, then redeploy the contract.
PM-8199	[Compactc] Compiler might not add a trailing slash to COMPACT_PATH. Users should add the trailing '/' at the end of the variable, if not using the provided wrapper script.
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-7798	[Faucet] Previous error message remains visible while processing request. Users can ignore the message and proceed.