

# changelog.md for @push.rocks/smartacme

## 2026-03-27 - 9.3.1 - fix(acme)

parse issued certificate expiry from X.509 metadata and update build compatibility for dependency upgrades

- Store certificate validity using the actual X.509 expiration date instead of a fixed 90-day estimate, with a fallback if PEM parsing fails.
- Add reflect-metadata imports and declare it as a direct dependency to support @peculiar/x509 v2.
- Update TypeScript and build configuration for newer toolchain requirements, including Node types and renamed project config file.

## 2026-03-19 - 9.3.0 - feat(readme)

document built-in ACME directory server and CA capabilities

- Update package metadata to describe client and server support, including built-in CA functionality
- Add README sections for ACME server quick start, configuration, endpoints, challenge verification, and trust setup
- Expand architecture notes and project hints to cover ts\_server modules and tsbuild tsfolders usage

## 2026-03-19 - 9.2.0 - feat(server)

add an embedded ACME directory server and certificate authority with challenge, order, and certificate endpoints

- exports a new server module with AcmeServer, AcmeServerCA, server error types, and related interfaces
- implements in-memory account and order storage, nonce management, JWS verification, routing, challenge validation, and CSR signing for RFC 8555 style flows
- adds end-to-end tests for account creation, order processing, challenge handling, certificate issuance, and error scenarios
- updates the build configuration to include tsfolders and package file patterns for ts\_\* sources

## 2026-02-16 - 9.1.3 - fix(smrtacme)

Include base domain alongside wildcard when building identifiers for wildcard certificate requests

- When isWildcardRequest is true, the base domain (e.g. example.com) is now added in addition to the wildcard (\*.example.com) so the issued certificate covers both apex and wildcard entries.
- Prevents missing SAN for the apex domain when requesting wildcard certificates.

## 2026-02-15 - 9.1.2 - fix(docs)

document built-in concurrency control, rate limiting, and request deduplication in README

- Added a new 'Concurrency Control & Rate Limiting' section to the README describing per-domain mutex, global concurrency cap, and sliding-window account rate limiting (defaults: 1 per domain, 5 global, 250 per 3 hours).
- Documented new SmartAcme options in the interface: maxConcurrentIssuances, maxOrdersPerWindow, and orderWindowMs.
- Added example code showing configuration of the limits and an example of request deduplication behavior (multiple subdomain requests resolving to a single ACME order).
- Added an example subscription to certIssuanceEvents and updated the components table with TaskManager entry.
- Change is documentation-only (README) — no code changes; safe patch release.

## 2026-02-15 - 9.1.1 - fix(deps)

bump @push.rocks/smrttime to ^4.2.3 and @push.rocks/taskbuffer to ^6.1.2

- @push.rocks/smarttime: ^4.1.1 -> ^4.2.3
- @push.rocks/taskbuffer: ^6.1.0 -> ^6.1.2
- Only package.json dependency version updates; no code changes

## 2026-02-15 - 9.1.0 - feat(smartacme)

Integrate @push.rocks/taskbuffer TaskManager to coordinate ACME certificate issuance with per-domain mutex, global concurrency cap, and account-level rate limiting; refactor issuance flow into a single reusable cert-issuance task, expose issuance events, and update lifecycle to start/stop the TaskManager. Add configuration for concurrent issuances and sliding-window order limits, export taskbuffer types/plugins, and update tests and docs accordingly.

- Added dependency @push.rocks/taskbuffer and re-exported ITaskEvent/ITaskMetadata in ts/index.ts; also imported/exported taskbuffer in ts/plugins.ts.
- Replaced interestMap coordination with TaskManager + TaskConstraintGroup(s): 'cert-domain-mutex' (per-domain mutex, resultSharingMode: 'share-latest'), 'acme-global-concurrency' (global concurrency cap), and 'acme-account-rate-limit' (sliding-window rate limiter).
- Introduced a single reusable Task named 'cert-issuance' and moved the ACME issuance flow into performCertificateIssuance(), splitting progress into named steps (prepare/authorize/finalize/store) and using notifyStep() for observable progress.
- Exposed certIssuanceEvents via SmartAcme.certIssuanceEvents and wired TaskManager.start()/stop() into SmartAcme.start()/stop().
- Added new ISmartAcmeOptions: maxConcurrentIssuances, maxOrdersPerWindow, orderWindowMs to control concurrency and rate limiting.
- Updated tests to remove interestMap stubs and adapt to the taskbuffer-based flow; cleaned up client/retry stubbing in tests.
- Updated readme.hints.md with guidance on concurrency, rate limiting, and taskbuffer integration.

## 2026-02-15 - 9.0.1 - fix(acme-http-client)

Destroy keep-alive HTTP agents and DNS client on shutdown to allow process exit; add destroy() on AcmeHttpClient and AcmeClient, wire agents into requests, and call client/smartdns destroy during SmartAcme.stop; documentation clarifications and expanded README (error handling, examples, default retry values).

- ts/acme/acme.classes.http-client.ts: added per-protocol http/https agents (keepAlive: false), use agent for outgoing requests, and added destroy() to explicitly destroy agents and free sockets.
- ts/acme/acme.classes.client.ts: added destroy() that forwards to the HTTP client to allow transport cleanup.
- ts/smartacme.classes.smartacme.ts: SmartAcme.stop now calls client.destroy() and smartdns.destroy() (when present) to ensure child processes and sockets are terminated before exit; also ensures certmanager.close() is awaited.
- readme.md: documentation improvements and clarifications (Let's Encrypt spelling, added RFC 8555 compliance note, error handling / AcmeError usage examples, default retry parameter docs, UI/emoji improvements, and other wording/formatting updates).

## 2026-02-15 - 9.0.0 - BREAKING CHANGE(acme)

Replace external acme-client with a built-in RFC8555-compliant ACME implementation and update public APIs accordingly

- Add complete TypeScript ACME implementation under ts/acme (AcmeClient, AcmeCrypto, AcmeHttpClient, AcmeAccount, AcmeOrderManager, AcmeChallengeManager, AcmeError, interfaces, ACME\_DIRECTORY\_URLS).
- Implement JWK/JWK-thumbprint, JWS creation, nonce management, bad-nonce retries, Retry-After handling, CSR generation via @peculiar/x509 and node:crypto.
- Update SmartAcme to use the new AcmeClient/AcmeCrypto API (e.g. plugins.acme.AcmeClient, accountKeyPem) and add AcmeError-aware retry/backoff logic.
- Remove dependency on the external acme-client and other unused packages; add @peculiar/x509 and bump multiple dependency/devDependency versions.
- Add/adjust tests (unit tests for crypto, challenge, error handling) and update test imports to @git.zone/tstest; update README/readme.hints and npmextra.json to reflect implementation and publishing changes.

## 2025-05-19 - 8.0.0 - BREAKING CHANGE(smartacme)

Make wildcard certificates opt-in to fix HTTP-01 only configurations

- BREAKING CHANGE: Wildcard certificates are no longer automatically requested for all domains
- Added 'includeWildcard' option to `getCertificateForDomain()` to explicitly request wildcard certificates
- HTTP-01 only configurations now work correctly as they do not try to request wildcard certificates automatically
- Updated certificate CSR generation to match the requested domain configuration

## 2025-05-19 - 7.4.0 - feat(smrtacme)

Make wildcard certificates opt-in to fix HTTP-01 only configurations

- BREAKING CHANGE: Wildcard certificates are no longer automatically requested for all domains
- Added `includeWildcard` option to `getCertificateForDomain()` to explicitly request wildcards
- HTTP-01 only configurations now work correctly as they no longer attempt wildcard certificates
- Wildcard certificates require DNS-01 handler and must be explicitly requested
- Updated certificate CSR generation to match the requested domain configuration

## 2025-05-18 - 7.3.4 - fix(smrtacme)

Refine documentation and tests for improved clarity in ACME certificate management

- Enhanced the README with detailed usage, configuration, and example sections
- Refined test cases for certificate matching and challenge handlers across DNS-01 and HTTP-01
- Updated TypeScript definitions and inline comments for better developer experience

## 2025-05-05 - 7.3.3 - fix(SmartAcme)

Remove duplicate challengeHandlers declaration from SmartAcme class

- Eliminated the redundant private declaration of challengeHandlers since it is already defined as a public property
- Ensures a single source of truth and clearer interface for challenge handler configuration

## 2025-05-05 - 7.3.2 - fix(test)

Add missing checkWetherDomainIsSupported implementation to DummyHandler for interface compliance in tests

- Implemented the missing checkWetherDomainIsSupported method in the DummyHandler to satisfy IChallengeHandler interface requirements
- Ensured that tests now correctly instantiate the DummyHandler without interface errors

## 2025-05-05 - 7.3.1 - fix(core)

Refactor import paths and update dependency references

- Replaced deprecated 'smartacme.plugins.js' with the new 'plugins.js' across cert managers, handlers, and core classes
- Added missing dependencies (@push.rocks/smartfile and @push.rocks/smartnetwork) in package.json
- Updated HTTP challenge handlers to include domain support checks via checkWetherDomainIsSupported
- Adjusted import paths in MongoCertManager, MemoryCertManager, and DNS-01 handler for consistency

## 2025-05-05 - 7.3.0 - feat(index)

Bump @tsclass/tsclass to 9.2.0 and update module exports to include handlers

- Upgrade @tsclass/tsclass dependency from 9.1.0 to 9.2.0 in package.json
- Add explicit export of handlers in ts/index.ts to improve module accessibility

# 2025-05-05 - 7.2.5 - fix(smartacme)

Refactor module exports and update wildcard certificate support documentation

- Updated readme.plan.md to streamline and remove obsolete wildcard plan details
- Normalized certmanager imports by consolidating exports in ts/index.ts and updating tests accordingly
- Reordered ISmartAcmeOptions interface properties for clarity (accountEmail moved to the top)

# 2025-05-04 - 7.2.4 - fix(test)

Refactor wildcard certificate test to properly stub SmartAcme.start and getCertificateForDomain for robust integration.

- Temporarily override SmartAcme.start and getCertificateForDomain to simulate wildcard certificate behavior.
- Restore original prototype methods post-test to prevent side effects.
- Improve test clarity for wildcard certificate integration.

# 2025-05-01 - 7.2.3 - fix(docs)

Improve certificate manager documentation with detailed examples and custom implementation guide

- Added usage examples for MemoryCertManager and MongoCertManager
- Provided a custom ICertManager implementation guide
- Enhanced overall documentation clarity for certificate storage configuration

# 2025-05-01 - 7.2.2 - fix(readme)

Update readme documentation: switch installation instructions to pnpm and clarify usage with MongoCertManager and updated SmartAcme options

- Replaced npm/yarn commands with pnpm commands for installation and testing.
- Added guidance to ensure the project is set up for TypeScript and ECMAScript Modules.
- Updated usage examples to include initialization of MongoCertManager instead of legacy mongoDescriptor.
- Revised challenge handlers examples to reference the current API signatures.

## 2025-05-01 - 7.2.1 - fix(smartacme)

Centralize interest map coordination and remove redundant interestMap from cert managers

- Removed interestMap property and related logic from MemoryCertManager and MongoCertManager
- Refactored SmartAcme to instantiate its own interestMap for coordinating certificate requests
- Updated getCertificateForDomain to use the new interestMap for checking and adding certificate interests

## 2025-05-01 - 7.2.0 - feat(core)

Refactor SmartAcme core to centralize interest coordination and update dependencies

- Moved interest coordination mechanism out of ICertManager implementations and into SmartAcme core
- Updated certificate managers (MemoryCertManager and MongoCertManager) to remove redundant interestMap handling
- Upgraded @push.rocks/tapbundle from 6.0.1 to 6.0.3 in package.json
- Revised readme.plan.md to reflect the new interest coordination approach

## 2025-04-30 - 7.1.0 - feat(certmanagers/integration)

Add optional wipe methods to certificate managers and update integration tests, plus bump tapbundle dependency

- Introduce wipe() in ICertManager to support integration testing by clearing stored certificates
- Implement wipe() in MemoryCertManager and MongoCertManager for resetting internal state
- Refactor SmartAcme constructor to consider wiping certificates in integration mode (commented out for now)
- Update integration test assertions and add console logging for domain certificate retrieval
- Upgrade @push.rocks/tapbundle from ^6.0.0 to ^6.0.1

## 2025-04-30 - 7.0.0 - BREAKING CHANGE(SmartAcme (Cert Management))

Refactor certificate management and challenge handling API to use a unified certManager interface, remove legacy storage, and update challenge workflows.

- Introduce ICertManager interface with MemoryCertManager and MongoCertManager implementations.
- Remove the legacy SmartacmeCertManager and update SmartAcme to require a certManager option instead of mongoDescriptor.
- Adjust certificate renewal logic to delete and store certificates through the new certManager API.
- Refine DNS-01 challenge handling by removing in-handler DNS propagation waiting and relying on external checks.
- Increase retry settings for robustness during challenge verification and certificate issuance.
- Update integration and unit tests to use the new certManager configuration.

## 2025-04-30 - 6.2.0 - feat(handlers)

Add in-memory HTTP-01 challenge handler and rename file-based handler to Http01Webroot

- Renamed Http01Handler to Http01Webroot in both implementation and documentation
- Introduced Http01MemoryHandler for diskless HTTP-01 challenges
- Updated tests and README examples to reflect handler name changes and new feature

# 2025-04-30 - 6.1.3 - fix(Dns01Handler)

Update dependency versions and refine Dns01Handler implementation

- Bump '@apiclient.xyz/cloudflare' to ^6.4.1 and '@tsclass/tsclass' to ^9.1.0 in package.json
- Remove duplicate Cloudflare import in smartacme.plugins.ts
- Refactor Dns01Handler to use IConvenientDnsProvider and add checkWetherDomainIsSupported method
- Align devDependencies versions for improved consistency

# 2025-04-27 - 6.1.2 - fix(repo)

Update repository metadata by replacing the LICENSE file with a license.md file for improved consistency.

- Removed the old LICENSE file.
- Introduced license.md as the new license documentation file.

# 2025-04-27 - 6.1.1 - fix(readme)

Fix license link reference in documentation

- Updated the license link from [license](#) to [license.md](#) in the License and Legal Information section

# 2025-04-27 - 6.1.0 - feat(readme)

Update documentation with detailed built-in challenge handlers and custom handler examples

- Expanded readme to include sections on Dns01Handler and Http01Handler usage
- Added examples for creating and registering custom ACME challenge handlers
- Improved clarity of ACME certificate management instructions using SmartAcme

# 2025-04-27 - 6.0.1 - fix(readme)

Remove extraneous code fence markers from license section in readme

- Removed unnecessary triple backticks wrapping the license information
- Improved clarity of the license section in the documentation

# 2025-04-27 - 6.0.0 - BREAKING CHANGE(SmartAcme)

Refactor challenge handling by removing legacy setChallenge/removeChallenge in favor of pluggable challengeHandlers and update documentation and tests accordingly

- Removed legacy challenge methods and introduced new 'challengeHandlers' and 'challengePriority' options
- Updated readme examples to demonstrate usage with DNS-01 (and HTTP-01) handlers
- Refactored internal SmartAcme flow to select and process challenges via the new handler interface
- Adjusted tests (including integration tests) to align with the updated challenge handling mechanism

# 2025-04-27 - 5.1.0 - feat(smartacme)

Implement exponential backoff retry logic and graceful shutdown handling in SmartAcme; update acme-client dependency to v5.4.0

- Added retry helper with exponential backoff for ACME client operations
- Introduced retryOptions in ISmartAcmeOptions for configurable retry parameters
- Enhanced graceful shutdown handling by cleaning up pending DNS challenges on signal
- Updated acme-client dependency from v4.2.5 to v5.4.0

# 2025-04-26 - 5.0.1 - fix(build)

Update CI workflows, bump dependency versions, and refine import and TypeScript configuration

- Changed CI workflow image and npmci package from '@shipzone/npmci' to '@ship.zone/npmci', and updated repository URLs
- Bumped several dependency versions in package.json (e.g. @api.global/typedserver, @push.rocks/lik, @push.rocks/smardata, @push.rocks/smardns, @tsclass/tsclass) to newer releases
- Adjusted smardns import to use the smardnsClient module for proper module resolution
- Updated tsconfig.json to add emitDecoratorMetadata and baseUrl settings
- Minor markdown and formatting tweaks in readme and gitignore files, and slight improvements in test async handling

## 2024-06-16 - 5.0.0 - No significant changes

This release contains no user-facing changes.

## 2024-06-16 - 4.0.8 - Structure and configuration updates

- BREAKING CHANGE(structure): renamed classes to avoid confusion
- update description
- update tsconfig
- update npmextra.json: githost

## 2024-01-28 - 4.0.7-4.0.6 - Internal fixes and updates

- A series of releases with routine bug fixes and maintenance updates.

## 2023-07-21 - 4.0.5-4.0.4 - Internal fixes and updates

- Multiple releases addressing internal issues and maintenance improvements.

## 2023-07-10 - 4.0.3 - Organizational changes

- switch to new org scheme

## 2022-09-27 - 4.0.0-4.0.2 - Internal fixes and updates

- Routine maintenance and internal bug fixes.

## 2022-09-27 - 3.0.15 - Breaking changes

- BREAKING CHANGE(core): update

## 2021-01-22 - 3.0.9-3.0.14 - Internal fixes and updates

- A range of releases focused on routine internal updates.

## 2020-11-18 - 3.0.0–3.0.8 - Internal fixes and updates

- Routine maintenance and internal bug fixes.

## 2020-02-10 - 2.1.2 - Breaking changes

- BREAKING CHANGE(core): streamline scope to certificate retrieval using dns challenge

## 2020-02-10 - 2.1.0–2.1.1 - Internal fixes and updates

- Routine fixes and updates.

## 2019-02-06 - 2.0.36 - New feature

- feat(Cert): now has validity check

## 2019-01-18 - 2.0.2–2.0.35 - Internal fixes and updates

- Routine internal updates and maintenance.

# 2018-10-07 - 2.0.0-2.0.1 - Internal fixes and updates

- Routine internal updates and maintenance.

# 2018-10-07 - 1.1.4 - Breaking changes

- BREAKING CHANGE(scope): change to @pushrocks

# 2018-08-12 - 1.1.1 - NPM publishing fix

- fix(npm publishing): update

# 2018-08-11 - 1.1.0 - Certificate issuance update

- fix(core): now creating certs all right

# 2018-08-11 - 1.0.11 - Feature update

- feat(switch to acme-v2): switch to letsencrypt v2

## 2017-04-28 - 1.0.10 - CI improvements

- add updated ci config

## 2017-04-28 - 1.0.9 - Standards update

- update to latest standards

## 2017-01-27 - 1.0.8 - Basic functionality

- basic functionality

## 2017-01-25 - 1.0.7 - Response and validation improvements

- now getting a valid response
- update validation
- improve README

## 2017-01-15 - 1.0.6 - Async and documentation improvements

- improve README
- add async checkDNS

## 2017-01-15 - 1.0.5 - Standards and process updates

- update to new standards
- now has working requestValidation method
- fix som things
- start better segregation of concerns
- start with certificate signing process

## 2017-01-01 - 1.0.4 - Certificate acquisition improvements

- now getting certificates
- can now agree to TOS
- remove test keys

## 2017-01-01 - 1.0.3 - NPM extra configuration

- add npmextra.json

## 2017-01-01 - 1.0.2 - README and integration update

- add better readme
- switch to rawacme for more basic letsencrypt access

# 2016-11-17 - 1.0.1 - Promise fix

- fix promise

# 2016-11-17 - 1.0.0 - Major initial release changes

- remove superfluous key creation
- switch to acme core
- prepare switch to le-acme-core
- improve upon keyCreation
- update to use more promises
- add README
- first version

---

Revision #2

Created 2026-03-28 13:08:09 UTC by foss.global Team

Updated 2026-03-29 16:51:01 UTC by foss.global Team