

XCOFDK-py - Release Notes

Version	:	1.9
---------	---	-----

Date	:	24.10.2025
------	---	------------

©Copyright	:	2023-2025 Farzad Safa (farzad.safa@xcofdk.de)
------------	---	--

All rights reserved.		
----------------------	--	--

Table of Contents

- [Table of Contents](#)
 - [Release Highlights](#)
 - [Release Highlights v3.2](#)
 - [Release Highlights v3.1](#)
 - [Release Highlights v3.0](#)
 - [Release Highlights v2.1](#)
 - [Release Notes](#)
 - [Release Notes v3.2 - 24.10.2025](#)
 - [Release Notes v3.1 - 30.09.2025](#)
 - [Release Notes v3.0 - 12.06.2025](#)
 - [Release Notes v2.1 - 04.02.2025](#)
 - [Release Notes v2.0.1 - 19.11.2024](#)
 - [Release Notes v2.0 - 11.11.2024](#)

Release Highlights

Release Highlights v3.2

- The framework now officially supports free-threaded (FT) Python:
 - The framework considers the stable version `3.14.0` the first Python version officially supporting *free-threaded (FT)*.
 - Also, the framework now considers Python versions `3.13` and pre-releases of the stable version `3.14.0` supporting *experimental free-threaded*.

If built with the build configuration for disabling GIL, for these interpreter versions use of the RTE configuration policy via the API function below still remains mandatory:

```
rtecfg.RtePolicyBypassExperimentalFreeThreadingGuard()
```
 - See also new API functions below:
 - `fwutil.IsFTPythonVersion()`
 - `fwutil.IsExperimentalFTPythonVersion()`

Release Highlights v3.1

- Introduction of public api function `fwutil.GetLcFailure()`.
- Introduction of RTE configuration policy for disabling console output:
 - `rtecfg.RtePolicyDisableLogRDConsoleSink()`
- Introduction of RTE configuration policy for redirection of logging output to file sink:
 - `rtecfg.RtePolicyEnableLogRDFileSink()`
- Introduction of RTE configuration policy for redirection of logging output to TCP sink:
 - `rtecfg.RtePolicyEnableLogRDTcpSink()`
- Task/process alias name is now allowed to be a printable, non-empty string literal without spaces.
- [DEPRECATED] Task state property `ITask.isAborting` is deprecated now:
 - with `False` is always returned,
 - `ITask.isFailed` shall be used instead.

TOC

Release Highlights v3.0

- Complete refactoring of the public API of package `xcofdk`:
 - re-organized the API on a *by-subsystem* basis,
 - interface classes have been introduced:
 - to avoid possible, undesired issue of cyclic import,
 - to avoid possible layout conflicts when task classes part of a multiple inheritance,
 - to ease access to task (or other) object instances created elsewhere
 - to improve type annotations.
- Introduction of pre-start configuration of the framework by RTE policy items and related API functions:
 - `rtecfg.RtePolicyGetConfig()`
 - `rtecfg.RtePolicyConfigure()`
 - `rtecfg.RtePolicyEnableAutoStop()`
 - `rtecfg.RtePolicyEnableTerminalMode()`
 - `rtecfg.RtePolicyEnableForcedAutoStop()`
 - `rtecfg.RtePolicyDisableSubSystemMessaging()`
 - `rtecfg.RtePolicyDisableSubSystemMultiProcessing()`
 - `rtecfg.RtePolicyBypassExperimentalFreeThreadingGuard()`
 - `rtecfg.RtePolicyDisableExceptionTrackingOfChildProcesses()`
- Introduction of class `RCTask` (and its derived classes) for rapid construction of task instances by simply passing callable objects (as opposed to mandatory subclassing of the abstract classes `XTask` and `XMainTask`):
 - `class SyncTask`
 - `class AsyncTask`

- `class SyncCommTask`
- `class AsyncCommTask`
- `class MessageDrivenTask`
- `class XFSyncTask`
- `class XFAsyncTask`
- `class XFSyncCommTask`
- `class XFAsyncCommTask`
- `class XFMessageDrivenTask`
- Introduction of new API functions/properties of task instances of both classes `RCTask` and `XTask`:
 - `isMainTask()`
 - `isCanceling`
 - `isPendingRun`
 - `Cancel()`
 - `SelfCheck()`
 - `SelfCheckSleep()`
 - `SendMessage()`
 - `BroadcastMessage()`
 - `GetTaskOwnedData()`
 - `SetTaskOwnedData()`
- Provide new API functions to enable access to currently running task instance:
 - `GetCurTask()` -> `IRCTask` | `IRCCommTask`
 - `GetCurXTask()` -> `IXTask`
- Refactoring of class `XProcess`:
 - to enable rapid construction of child processes with no obligation to have to deal with framework-specific data types,
 - to provide the new feature of *exception tracking* of child processes when executed,
 - to provide a new API function to terminate a child process.
- New API functions to wait for termination (of a subset) of running tasks or child processes:
 - `fwapi.JoinTasks()`
 - `fwapi.JoinProcesses()`
- New API function to terminate (a subset) of running child processes:
 - `fwapi.TerminateProcesses()`
- Default start options of the framework properly arranged in a way commonly expected for an application deployed in release mode.
- Two new framework start options are available now:
 - `--fw-log-level`
 - `--disable-log-callstack`
 - (`--enable-log-callstack` has been removed).

TOC

Release Highlights v2.1

- Tasks can send messages even if in teardown phase.

- Support of positional and/or keyword arguments passed to tasks when started.
- All user or fatal errors submitted by the framework are given a pre-defined, unique error code.
- Synchronous tasks can be configured to have a cyclic run phase, too.
- Two new framework start options are available now:
 - `--enable-log-callstack`
 - `--suppress-start-preamble`

TOC

Release Notes

Release Notes v3.2 - 24.10.2025

- XPY-369 – Update doc: source code, readme and release notes
- XPY-368 – Enable official support of free-threaded Python

TOC

Release Notes v3.1 - 30.09.2025

- XPY-366 – Update readme and release notes
- XPY-359 – Introduce public api function `GetLcFailure()`
- XPY-358 – Deprecated: Task state property `isAborting` is deprecated now
- XPY-357 – Allow task/process alias name to be a printable. non-empty string literal without spaces
- XPY-350 – Introduce RTE policy for redirection of logging output to TCP sink
- XPY-349 – Introduce RTE policy for redirection of logging output to file sink
- XPY-347 – Introduce RTE policy for disabling console output
- XPY-345 – Bugfix: Sporadically exception raised while determining current running task
- XPY-341 – Bugfix: `isFailed` returns wrong value upon failure (especially for async. tasks)
- XPY-328 – Bugfix: Creating a child process before start of the framework ends up with an exception
- XPY-316 – Bugfix: Sporadically wrong exit code of child process

TOC

Release Notes v3.0 - 12.06.2025

- XPY-327 – Update readme and release notes
- XPY-314 – Rename public API module `curxtask` and its API
- XPY-311 – Introduce public API property `IXTask.isMainTask`
- XPY-310 – Introduce public API property `ITask.isPendingRun`
- XPY-308 – Re-organize subpackage `fwapi.rtecfg`
- XPY-298 – Provide framework start option `--disable-log-callstack`

- XPY-299 – Remove framework start option `--enable-log-callstack`
- XPY-297 – Introduce public API subpackage `fwapi.fwctrl`
- XPY-293 – Bugfix: Send failure when a task is going to be stopped
- XPY-290 – Bugfix: Failure regarding list of joinable child processes
- XPY-285 – Provide API functions `SendMessage()` and `BroadcastMessage()` for task instances
- XPY-280 – Refactoring: Implementation of class `XTask`
- XPY-272 – Provide public API function `TerminateProcesses()`
- XPY-269 – Rework default settings of start options
- XPY-266 – Provide basic examples for RC tasks
- XPY-264 – Introduce RTE policy for disabling exception tracking of child processes
- XPY-263 – Refactoring: Implementation of class `XTaskException`
- XPY-258 – Provide public API `XProcess.processException`
- XPY-256 – Provide public API `XProcess.Terminate()`
- XPY-255 – Close host process resources if terminated
- XPY-253 – Refactoring: Implementation of class `XTaskProfile`
- XPY-251 – Provide property `XProcess.aliasName`
- XPY-249 – Introduce public API `XTask.taskCompoundUID`
- XPY-248 – Refactoring: Construction of instances of class `XProcess`
- XPY-246 – Missing return value of `XProcess.Join()`
- XPY-244 – Provide timeout implementation of `XProcess.Join()`
- XPY-242 – Provide public API function `SelfCheck()` of tasks
- XPY-235 – Introduce public API function `JoinProcesses()`
- XPY-234 – Refactoring: Top-level design and layout of subpackage `xcofdk.fwapi`
- XPY-230 – Refactoring: Implementation of `ToString()`
- XPY-229 – Introduce RTE policy items for disabling specific subsystems
- XPY-225 – Refactoring: Implementation of dependency injection
- XPY-223 – Refactoring: Implementation of class `XProcess`
- XPY-218 – Re-organize existing basic examples for SC tasks
- XPY-217 – Allow message posting even for tasks with no external queue support
- XPY-216 – Introduce public API function `JoinTasks()`
- XPY-215 – Provide public API `GetCurXTask()`
- XPY-212 – Refactoring: Implementation of default logging config
- XPY-210 – Refactoring: Implementation of subsystem `fwerrh`
- XPY-208 – Arrange public availability of framework start option `--fw-log-level`
- XPY-202 – Provide public API `GetCurTask()`
- XPY-195 – Introduce new API function `Cancel()` of tasks
- XPY-189 – Design and implementation of class `RCTask`
- XPY-185 – Introduce read/write API for task-owned, user-specific data of tasks
- XPY-178 – Introduce new public API class `RteConfig`
- XPY-103 – Introduce RTE policy items for both auto-stop and terminal mode
- Total of 64 additional planned PM issues mostly related to stability, ease of use, re-organization and clean code.

Release Notes v2.1 - 04.02.2025

- XPY-172 – Bugfix: Resolve typo by wrong getter to access message header.
- XPY-171 – Bugfix: Resolve typo related to cmdline option ' - - fibonacci-input ' of example applications.
- XPY-162 – Bugfix: Increment of messaging error counter of multithreaded userapp examples.
- XPY-154 – Support of positional and/or keyword arguments passed to tasks when started.
- XPY-150 – Refactoring: detailed design and implementation of (limited) RTE modes.
- XPY-148 – Enable message sending in teardown-phase of tasks.
- XPY-134 – Rework output format of LC failures.
- XPY-125 – Provide framework start option ' - - suppress-start-preamble '.
- XPY-124 – Provide framework start option ' - - enable-log-callstack '.
- XPY-120 – Bugfix: Error logging not working after start of synchronous tasks.
- XPY-119 – Bugfix: Framework gets stuck if application missed to stop it before joining.
- XPY-115 – Make synchronous tasks may have cyclic run-phase, too.
- XPY-110 – Provide pre-defined error codes for errors reported by the framework.
- Total of 26 additional known and/or planned PM issues mostly related to stability.

TOC

Release Notes v2.0.1 - 19.11.2024

- Add command line option of userapp basic examples to print out a usage hint.
- Bugfix regarding scan of command line arguments of userapp basic examples.

TOC

Release Notes v2.0 - 11.11.2024

- Created release v2.0.

TOC
