

OPTIPRICEPRO

Institutional Product Catalogue

SIX SPECIALIST PRICING PRODUCTS

JAX | PANDAS | C++

Python API clients and browser-based Studio workbenches for derivatives pricing, Monte Carlo analysis and risk sensitivities.

Asteroid Belt Trading Company Limited
United Kingdom · Company number 17451422

Catalogue edition 1.0 | 19 October 2026

Product descriptions derived from twelve supplied installation and user guides. This catalogue is an informational draft; commercial terms and deployment assurances are subject to written agreement.

Executive overview

OptiPricePro is a six-product family organized around three pricing-service technologies—JAX, Pandas and C++—each available through a Python-facing interface and a browser-based Studio workbench. The supplied customer guides describe Monte Carlo-based valuation across vanilla and path-dependent derivatives, with configurable exercise styles, stochastic models and numerical controls.

Commercial delivery model

The current JAX and C++ API clients, all three Studios, and the comprehensive Pandas guide describe client-side packages that transmit requests to protected pricing services over HTTPS. Customers interact through installed Python packages or a local browser workbench; proprietary hosted pricing code is not delivered. One separate Pandas cross-platform guide also describes a local-execution wheel; availability and scope of that variant require confirmation.

Who may benefit

Quantitative research and model development teams; derivatives trading and structuring desks; valuation and independent price-verification functions; risk analytics groups; and institutions evaluating programmatic access to configurable simulation models. These are intended customer segments, not claims of current deployments.

Product family

Product	Interface	Customer experience
JAX	Python client	Python scripts and integrations; hosted JAX service
JAX Studio	Browser workbench	Interactive configuration; hosted JAX service
Pandas	Python interface	Comprehensive flat price() API; hosted service in main guide
Pandas Studio	Browser workbench	Interactive configuration; hosted Pandas service
C++	Python client	Python/CLI interface; hosted C++ service
C++ Studio	Browser workbench	Catalog-driven controls; hosted C++ service

Product 01 | OptiPricePro JAX

A Python client that exposes the OptiPricePro pricing service through the OptiPricePro class. The customer-facing guide shows calls such as `opp.price("European", "Call", S0=100, K=100, ...)`, returning a dictionary containing price and standard error.

Integration

Customers install the `optipricepro` Python wheel, configure an API key and send HTTPS requests. The guide describes Windows, macOS and Linux installation, Python 3.9+, and no customer-side JAX or Docker requirement.

Illustrative workflow

Instantiate `OptiPricePro()`; specify contract, market and simulation inputs; retrieve a result dictionary; store values, aggregate portfolio exposures or populate pandas tables.

Version and boundary

The installation guide illustrates wheel version 1.0.2 but directs customers to substitute the actual supplied release. The Python usage guide is introductory and does not itself enumerate the full JAX service model catalogue. For detailed feature coverage, refer to the corresponding JAX Studio guide and validate API parity before promising it.

Source guides

`JAX_Customer_Installation_Guide.docx`; `JAX_Customer_Python_Usage_Guide.docx`

Product 02 | OptiPricePro JAX Studio

A locally launched graphical workbench for the hosted JAX pricing service, documented as Studio 1.0.2 / JAX pricing service v1 (July 2026). Users assemble valuation requests without writing Python.

Documented product scope

European, American, Bermudan and native lookback exercise; vanilla, digital, arithmetic/geometric Asian, floating-strike and rolling-average structures; barrier and lookback compositions; configurable stochastic rates, volatility, dividends, sentiment and sentiment volatility; stock jumps; simulation choices; importance sampling and analytical control variates.

Sensitivities and diagnostics

Price, Monte Carlo error, headline Greeks, complete gradient and optional Hessian; request inspection and selected diagnostic outputs. Hard payoff is the documented default, while smoothing is optional and changes the computed value.

Installation and access

Package: `optipricepro_jax_gui-1.0.2-py3-none-any.whl`. Requires Python 3.9+, browser, outbound HTTPS/443 and a separately issued Studio key. Command: `optipricepro-jax-gui`; default local interface `localhost:8501`.

Source guides

`OptiPricePro_JAX_Studio_Installation_Guide_v1.0.2(1).docx`; `OptiPricePro_JAX_Studio_User_Guide_v1.0.2(1).docx`

Product 03 | OptiPricePro Pandas

A Python-facing pricing product described in a comprehensive 105-page user guide, prepared from API version 2.1.0. Its public price() interface supports flat named parameters and a broad range of derivatives structures and numerical methods.

Documented scope

European, American, Bermudan and native lookback exercise; vanilla, digital, Asian, barrier and lookback structures; nested compositions; stochastic model families, jumps, importance sampling, control variates, finite-difference Greeks and Hessian-related workflows. The guide includes tutorials and regression/benchmark recipes.

Integration and execution

The comprehensive guide describes a private optipricepropandas wheel authenticating to a protected hosted pricing service. It notes that requests are serialized by an engine lock, which matters for enterprise concurrency planning.

Deployment clarification required

A separate cross-platform guide states that buyers may receive a local-execution wheel without an API key, a hosted API key, or both. This differs from the comprehensive guide's hosted-service description. Both statements are preserved here; product packaging and availability must be reconciled before contractual commitments.

Source guides

OptiPricePro_Pandas_Customer_Guide_with_API_Key(2).docx; OptiPricePro_Pandas_Customer_User_Guide(1).docx

Product 04 | OptiPricePro Pandas Studio

A graphical interface for the hosted Pandas pricing service, documented as Studio 1.0.1 / pricing service API 2.2 (July 2026). Customers construct valuation requests through a browser and inspect returned results.

Documented scope

European, American, Bermudan and native lookback exercise; vanilla, digital and Asian structures; barrier and lookback wrappers; stochastic rates, volatility, dividends and sentiment; jumps; quasi-random methods, variance reduction, importance sampling, control variates and finite-difference Greeks.

Simulation controls

Time intervals, paths, seed, standard/quasi-random generators, Latin hypercube and stratified sampling, and Vanilla, Antithetic, Brownian Bridge or Moment Matching Wiener-process choices. The interface displays request summaries and diagnostic outputs.

Installation and access

Package: optipricepropandas_gui-1.0.1-py3-none-any.whl. Python 3.9+, browser, outbound HTTPS/443, separate Studio API key. Launch command: optipricepro-gui.

Source guides

InstallationGuide_OptiPriceProPandasGUI.docx; UserGuide_OptiPriceProPandasGUI.docx

Product 05 | OptiPricePro C++

A Python API and command-line client for a protected C++ pricing service. Documentation identifies client version 1.0.1 and engine v2.4 (July 2026). The installed client contains no pricing engine logic.

Documented structures

Terminal-price and arithmetic/geometric Asian products, including floating-strike and rolling-average variants; Call, Put, DigitalCall and DigitalPut payoffs; European, American and Bermudan exercise; barrier and lookback compositions. The guide describes discrete and continuous barrier variants.

API and operational features

Example: `import optipricepro_cpp as opp` followed by `opp.price(product="PathIndependentOption", payoff="Call", ...)`. A service-backed `opp.catalog()` lists accepted selectors. The package also registers `check`, `catalog` and pricing CLI commands.

Installation

Package: `optipricepro_cpp-1.0.1-py3-none-any.whl`. Requires Python 3.9+, HTTPS/443 and an API key; no customer-side C++ compiler, Eigen or Boost.

Release caveat

The source filename includes “v1.0.2”, while the guide body identifies client 1.0.1 / engine 2.4. Confirm the distributable wheel version before issuing final catalogue copy.

Source guides

`OptiPricePro_CPP_Installation_Guide_v1.0.2.docx`; `OptiPricePro_CPP_User_Guide_v1.0.2.docx`

Product 06 | OptiPricePro C++ Studio

A browser-based workbench for the hosted C++ pricing service, documented as Studio 1.0.3 / engine v2.4 (July 2026). Controls are populated from the service catalogue and adapt to selected contracts and stochastic factors.

Documented scope

Vanilla and Asian contracts; Call, Put and digital payoffs; European, American and Bermudan exercise; barrier/lookback structures; configurable market state, stochastic model parameters and correlation matrices. The interface provides service connectivity and licence-quota checks.

Interaction design

Only relevant controls appear: Bermudan dates for Bermudan exercise, Heston parameters for Heston volatility, and appropriately sized correlation matrices when multiple stochastic factors are enabled.

Installation and access

Package: optipricepro_cpp_gui-1.0.3-py3-none-any.whl. Python 3.9+, browser, HTTPS/443 and a separately metered Studio licence. Launch: optipricepro-cpp-gui.

Source guides

OptiPricePro_CPP_Studio_Installation_Guide_v1.0.3.docx; OptiPricePro_CPP_Studio_User_Guide_v1.0.3.docx

Institutional evaluation process

1. Scope the use case

Agree on contract families, exercise rules, required stochastic factors, desired Greeks, expected throughput, acceptable numerical tolerances and deployment constraints.

2. Provision controlled access

Supply the correct product wheel, its paired guides, a time-limited evaluation key and the approved endpoint. Never include real keys in sales decks or shared source files.

3. Reproduce reference valuations

Run agreed vanilla and path-dependent examples with fixed seeds and explicit parameters. Record analytical references where available, Monte Carlo errors, numerical convergence and any material model assumptions.

4. Review production requirements

Assess authentication, usage quotas, logging, availability, data residency, security, incident handling, concurrency and support. These are diligence topics, not presently verified certifications or SLAs.

5. Commercial decision

Prepare a written order form defining licensed product(s), named legal entity, evaluation or production use, usage limits, pricing, term, renewal, support and permitted integration. No fees or guarantees are asserted in this draft.

Commercial identity

Seller: Asteroid Belt Trading Company Limited

UK company number: 17451422

Product brand: OptiPricePro

Publication and technical diligence notes

The following items are intentionally disclosed for internal approval before this catalogue is sent externally.

Model semantics across engines

JAX and Pandas Studio documentation describes the two nested barrier/lookback orders as distinct contracts with different values. C++ API and Studio guides describe them as equivalent under the C++ engine's contract semantics. They must not be represented as universally interchangeable across all products.

Version control

Confirm deployed engine versions, current wheel filenames, JAX client release, C++ client filename/body mismatch and whether Pandas API 2.1 versus Studio API 2.2 reflects supported differences.

Hosted versus local Pandas

Confirm whether a true local-compute Pandas licence is available, how it is packaged, and whether local distribution exposes proprietary implementation. The catalogue presently describes both source statements without selecting one.

Security and performance evidence

Obtain live service checks, reproducible pricing results, numerical benchmark reports, security architecture, operational support details and throughput measurements before making claims of production readiness, superiority, uptime or performance.

Intellectual property and commercial terms

Verify chain of title and software IP assignment, employment-related restrictions, EULA, privacy/data processing terms, export or regulatory requirements where applicable, and final subscription/pricing policy.

Document provenance

Prepared solely from the twelve customer guides supplied on 19 October 2026. No binaries, services, licence servers, benchmark datasets or corporate register entries were independently inspected.

© 2026 Asteroid Belt Trading Company Limited. Draft catalogue for commercial review.