Celoxica
pure acceleration

Generic eXecution Accelerator GXA Lite

Product Overview

About This Document

This document describes the GXA Lite product overview.

Copyright Information

Sales	sales@celoxica.com
Customer Support	support@celoxica.com
Website	http://www.celoxica.com

UK Head Office
Celoxica Limited
34 Porchester Road
London
W2 6ES, UK
Phone: +44 (0) 20 7313 3180

US Head Office
Celoxica Inc.
275 Madison Avenue, Suite 404
New York, NY
10016, USA
Phone: +1 (0) 212 880 2075

US Chicago Office
Celoxica Inc.
141 W Jackson Blvd, Suite 2350
Chicago, IL
60604, USA
Phone: +1 (0) 312 893 1204

Content

Table of Figures

Revisions

Revision	Date	Description of Changes
R2013-1.0	28 DEC 2012	Initial version

1. Functional Description

1.1 Principles

Celoxica's Generic eXecution Accelerator GXA Lite is an FPGA-based solution providing ultra-low-latency order entry to exchanges and high-performance communication with a client software application.

User applications integrated with GXA Lite are able to leverage its functionality, which hides the complexities of working with multiple market access formats and facilitates order management.

GXA Lite ships as a set of dynamic libraries, which the user code can interface to via the single and simple user API it exposes.

GXA Lite enables multi-threading integration into the user applications code.

GXA Lite communicates with the exchanges using the execution venue native protocol.

GXA Lite can be configured to target several markets; it can also be configured to send commands e.g. order entry, order replace or order cancel via multiple shared memory input queues and receive market reports via multiple shared memory output queues. The user can assign priorities to the queues: for instance such assignments allow the user to give priority to new orders over amendments or cancellations.

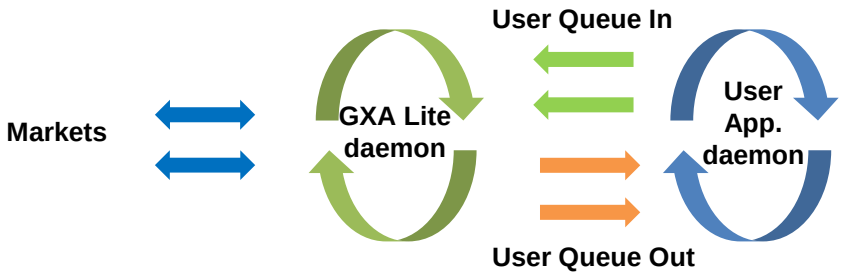

Figure 1 – Multiple user queues

See the GXA Lite user guide for more details.

1.2 Configuration

GXA Lite is configured using a GXA Lite configuration file which implements a configuration tree upon GXA Lite initialization.

The configuration tree allows GXA Lite users to set two configuration parameters types:

- **Global settings**

  Input and output queues settings, TOE TCP Offload Engine settings, order recovery settings …

- **Market settings**

  Specific market parameters, credentials for logon, lists of specific execution venue tags …

1.3 Order Management

1.3.1 Supported Functions

GXA handles the following functions through the API it exposes:

1. **Order entry**: an order entry is sent
2. **Order replace**: an order is replaced
3. **Order cancel**: an order is cancelled

The list of supported order types is as follows:

Order type
Market
Limit
Limit on close
Market on close
Inside limit

Order type
Add liquidity only
Stop loss
Stop limit
Pegged (market, primary, mid)
Post no preference
Inter-market Sweep Order

The list of supported time in force is as follows:

Time in Force
IOC
GTC
GTX
DAY
FOK
On open

The list of supported side is as follows:

Side
Buy
Sell
Sell short
Sell short exempt

The supported functions depend on the targeted execution venue.

GXA Lite does not maintain the order state: GXA Lite only makes available to the user the order state as provided by the exchange if the exchange provides order states.

1.3.2 Client Order Identifier Handling

Exchanges often require the client order identifier to be unique per trading session. GXA Lite is able to generate and maintain client order identifiers on behalf of the user application.

1.3.3 Order Recovery

GXA Lite provides an order recovery facility for the following use cases:

1. The user application is restarted intra-day.

   The user can retrieve all active orders from GXA Lite thanks to an API call.

2. GXA Lite is restarted intra-day.

   GXA Lite recovers all active orders thanks to the GXA Lite order persistence feature. A configuration tag can enable or disable this functionality per market. When enabled, GXA Lite loads all active orders thanks to previously submitted commands and received exchange-events automatically stored in internal files.

   The user can then retrieve the active orders from GXA Lite thanks to an API call.

1.4 Shared Memory API

1.4.1 Standard Usage

Firstly the user has to initialize one or more GXA contexts. A GXA context mainly consists in associating one input queue with one output queue.

Then the user can start sending commands (order creation, order amendment and order cancellation) to the markets via the shared memory input queues using a GXA context previously initialized. Sending commands is done as follows:

1. Firstly the user has to reserve a command
2. Secondly the user sends the command to the market

A GXA ID is automatically assigned to the command upon its successful push to the market. A new GXA ID is assigned to each successful command push even if the commands pertain to the same order.

The market feedbacks are received on the shared memory output queues specified by the GXA context. Note that the user is not notified upon receipt of market reports. The user is responsible for reading the output queue. Reading output queues is done as follows:

1. Firstly the user gets the information received from the market.

   If the market event is related to a command previously sent, the user will get the GXA ID of the command at the origin of the event for reconciliation purpose.

2. Secondly the user releases the received information.

1.4.2 Order Recovery

At any time the user can recover all active orders.

1. Firstly, the user requires GXA Lite to provide all active orders.

   This request can be used whether the user application or GXA Lite is restarted intra-day.

2. Secondly, the user gets the active order information.

   It is then strongly recommended to cancel the recovered orders.

1.5 Use Case

The GXA Lite daemon can be configured to target several markets, send commands via multiple shared memory input queues and receive market reports via multiple shared memory output queues.

The following figure illustrates a two-input queue and single output queue GXA Lite daemon configuration with two markets being targeted.

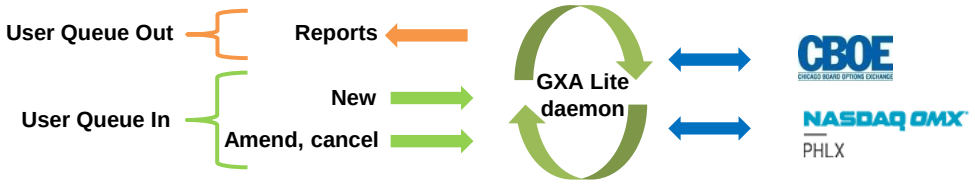

Figure 2 – Use case

2. Technical Architecture

2.1 Principles

The Celoxica application daemons, especially the GXA Lite daemon, are monitored by the Celoxica Manager.

The Celoxica Manager is a Python script which allows the user to monitor and parameterize the following processes:

1. **Master Daemon**

   The Master Daemon or the Manager Daemon is responsible for monitoring all the other Celoxica processes and switching to the backup node in case of failure on the master node.

2. **Logger Daemon**

   This process is responsible for recording application logs asynchronously.

3. **Celoxica Application Daemons**

   The following application daemons are monitored by the Manager:

   a) UltraBook (UB)

   b) FIX Market Access Gateway (MAG)

   c) Generic eXecution Accelerator (GXA Lite)

The Manager provides commands that allow the user to monitor all of these processes.

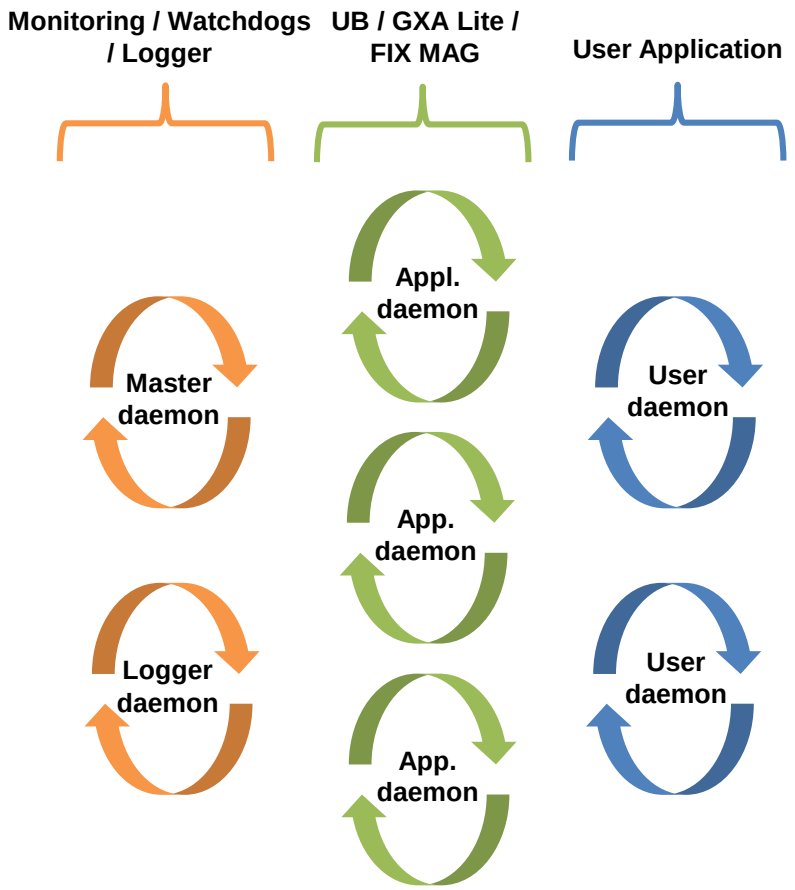

Figure 3 – Celoxica Manager daemon

See the Celoxica Manager user guide for more details.

2.2 Master Daemon

The Master Daemon can start and stop processes, acts as a watchdog and sends alerts in case of issues (e.g. process down or restarted).

It is responsible for ensuring and maintaining system availability, and is also able to manage failover to a backup node – note that this feature is not available yet.

The list of processes to watch and actions are fully configurable, for instance:

- Automatic process restart

  When a process is stopped, the Master Daemon tries to restart the process. The number of restart attempts and the timespan during which the Master Daemon tries to restart the process are configurable.

- Stop watching a process after n restart attempts

  If the Master Daemon fails to restart a process, the process will cease to be watched.

2.3 Logger Daemon

The processes which use the Logger Daemon write their logging information in shared memory. The Logger Daemon reads the shared memory, collects and formats the logs, and writes the formatted log messages to files. Log messages are coloured according to the log level. The log size is set at startup to avoid disk space issues during runtime. The number of log files is configurable and the logs can be written to specific files according to filtering expressions.

The Logger Daemon therefore provides a simple and homogeneous logging approach across applications.

If the Logger Daemon is down for any reason, the processes will write their logs into default logging files.

The Logger Daemon is automatically started when the Master Daemon starts.

2.4 GXA Lite Daemon Monitoring

GXA Lite is monitored by the Master Daemon.

GXA Lite uses the Logger Daemon.

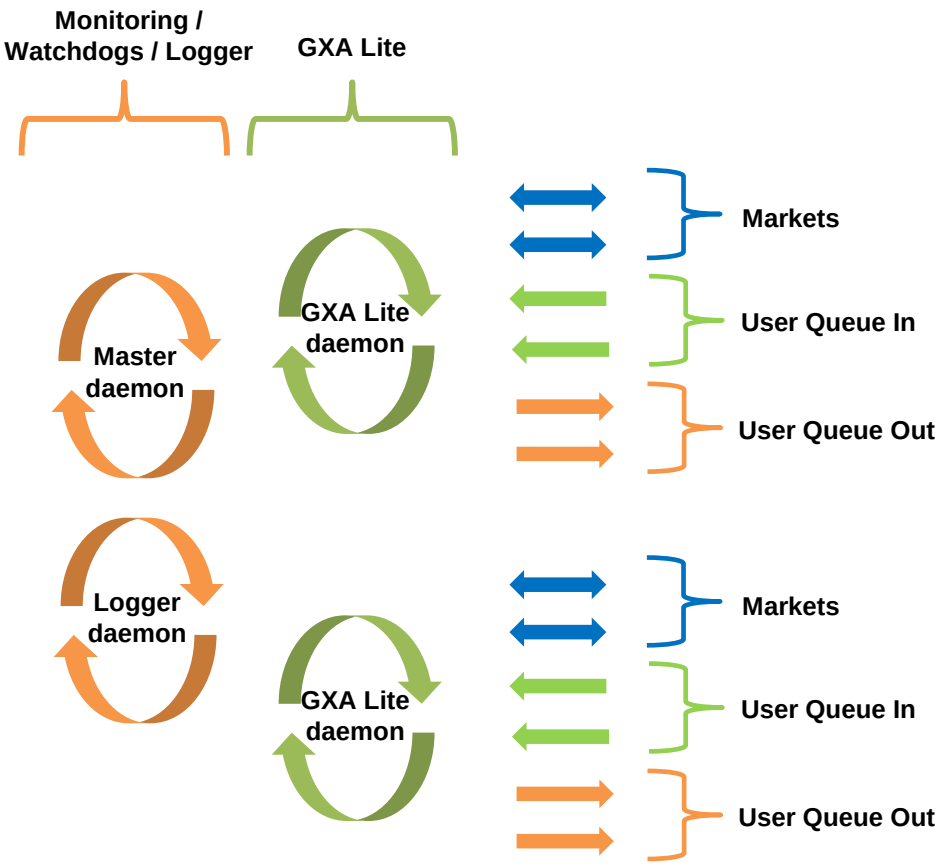

Figure 4 – GXA Lite daemon