neoxen®

Neoxen QX Framework

White Paper

Table of Contents

Multi-platform System Software Development

1   Neoxen QX Framework

1.1   Executive Summary

Neoxen QX Framework is a 32/64-bit System Software Development Kit (SDK). It is designed for distributed information management and communications systems for multiple client and server operating systems. It combines on-premises development with cloud computing. It scales from smart hand-held devices to private and public cloud environments.

Together with the included Visual Modus QX SDK, Neoxen QX Framework provides integration capabilities to any Information Management solutions supporting ODMA, WebDav or any of the standard TCP protocols. Additionally, it allows deep integration with Microsoft SharePoint 2010/2013/2016 and Bentley ProjectWise.

All current desktop and server versions of Microsoft Windows are supported – From Windows 7 and Windows 8(.1)/10 to Windows Server 2012(R2).

Neoxen QX Framework is designed for project groups, systems integrators and independent software vendors for rapid system software development. It provides a solid foundation, which drastically shortens design and implementation time and significantly reduces the development costs. Neoxen Professional Services uses it as a foundation technology in all customer projects requiring a robust, high-performance platform.

1.2   From Smart Devices to Server Clusters

The origin of Neoxen QX Framework was in the evolving need to speed up the development cycle of projects requiring distributed components. Intelligent code re-use, flexibility and robustness were the key requirements. These design principles and visions have guided the development ever since. Globalization and increasingly versatile infrastructure emphasize the need for a small, fast and highly flexible development platform even further.

When distributed multi-platform applications are produced for customers, the possibility to offer a short and efficient project cycle with flexible and robust software results can be a true winning combination.

Some of the essential features required from a distributed development platform are:

- It should be transparent to use in different platforms, from desktop operating systems to servers, from smart hand-held devices to multimedia terminals.

- It needs to seamlessly support all the Microsoft operating environments in the system level. It needs to be compliant with Microsoft Windows SDK, Microsoft Foundation Classes (MFC), Microsoft Visual Basic, scripting languages and C#. It should provide developers with an optimal path between existing and next generation systems.

- It should contain ready-made project, server and client templates, in order to shorten the time-to-market and time-to-money spans.

- It needs to be small in size and have a small memory footprint. It should be very well performing and robust. It should scale from small devices to enterprise servers. The structure should be modular, preferably object oriented and it needs to support multiple programming languages.

- It should be fully internationalized and provide easy localization procedures for all the major language areas, including the Far East.

Neoxen QX Framework answers these requirements and many others. Therefore it is the chosen development platform for any organization seeking for more competitive advantages in today's rapidly evolving global markets.

1.3 Neoxen QX Framework - Built To Meet the Business Needs

Designed as a highly flexible and customizable toolkit, Neoxen QX Framework allows exercising as much control over the architectural layers as required by the business needs of the solution. It permits developers and integrators to easily leverage the full power of this technology for their own benefit.

By design, solution development with Neoxen QX Framework allows subsequent access to all the technologies and platforms supported by Microsoft and many others. The users of the platform are not forced to stick in a single solution or technology. This business-oriented flexibility offers a true competitive advantage.

This technology specifically offers the following system development benefits:

- The modular architecture integrates easily with existing solutions and services

- It works with many different commonly used technologies from multiple manufacturers

- Multi-layered architecture is extendable and customizable with all layers accessible to developers

Using Neoxen QX Framework as the main development platform also has a positive impact on development and maintenance costs, whether it is implemented with a single application or within a network of different solutions.

2   Main Advantages

2.1   Save in Development Costs

Neoxen QX Framework is easy to learn and easy to use and the licensing is flexible and competitive. The resulting binaries are royalty-free. Neoxen QX Framework contains a consistent interface with lots of extensively tested modules. Therefore less coding is required to achieve the targets and the testing efforts can be concentrated on the business level functionality.

2.2   Build Better Solutions

As you save in development costs, you will also build better solutions utilizing the extensively tested and proven foundation. Less bugs, shorter testing cycles and shorter project completion will raise your customer satisfaction and give you a competitive advantage.

2.3   Save in Support Costs

When there is a need to upgrade or modify your solution, your developers can typically concentrate on just the business level. Neoxen QX Framework Release 10.58 is backwards compatible. If there is a need to upgrade the platform, the existing solutions remain fully functional and can even take advantage of some of the new features without re-coding.

2.4   Earn Confidence

Whether you create a solution designed to reside on an intranet or run over the Internet, customers will feel more confident using a system based on a robust and proven platform. Furthermore, developers can have the Neoxen QX Framework sources stored in an escrow service for even further confidence.

One of the additional benefits of Neoxen QX Framework is that its usage is not restricted to any particular programming language or development environment. The core technology is written in low-level C/C++ for optimal performance and inter-operability. Therefore, all the functionality is accessible from any programming language capable of calling C-style library functions (C, C++, C#, most scripting and macro languages, Java, etc.).

The whole code base of Neoxen QX Framework has been fully internationalized. Therefore the solutions created can be easily localized to any language area, including the Far East.

As a whole, Neoxen QX Framework is an answer to the development of globally as well as locally targeted business driven solutions.

3 General Availability

3.1 Shipping Release

The Neoxen QX Framework Release 10.58 is available directly from the vendor. Please visit our web site at http://www.neoxen.com or contact our sales department at sales@neoxen.com.

Note:	Nothing contained on this document should be construed as a commitment on the part of Neoxen Systems to license any particular person or entity to use the Neoxen QX Framework Software and Documentation. While Neoxen Systems welcomes all inquiries regarding the product and technology, Neoxen Systems reserves the right to refuse to license the Neoxen QX Framework Software and Documentation to anyone for any reason.

DOCUMENTATION LICENSE

This documentation, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Neoxen Systems.

Neoxen Systems assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Neoxen Systems.

Information in this document is provided in connection with the vendor products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Neoxen Systems reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

TRADEMARKS

Neoxen, the Neoxen logo, Trelox, Neoxen QX, Neoxen QX Framework, Neoxen Modus, Neoxen Visual Modus, Neoxen iModus, Neoxen EveryPlace and Neoxen NaviList are trademarks or registered trademarks of Neoxen Systems in USA and/or other countries.

Microsoft, Microsoft Office, Microsoft Windows and Microsoft Visual Studio are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat and Acrobat Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds.

All other trademarks, registered trademarks and/or product names are property of their respective owners.

COPYRIGHT

Microsoft Partner
Gold Application Development
Gold Intelligent Systems
Gold Data Analytics
Gold Data Platform
Gold Datacenter
Silver Cloud Platform