

This way to .NET

NetQuarry Studio

Application Developers build applications by doing one of three primary tasks. First, a developer develops or analyzes the data that is to be used in the application. Second, the developer uses NetQuarry Studio to manipulate meta-data that describes to the NetQuarry runtime how the end application will look and manage the application's data. Third, any custom business logic is written using any .NET language and IDE (e.g. Visual Studio .NET) to produce focused application extensions.

To describe an application to the NetQuarry runtime the developer interacts with the NetQuarry Studio. The NetQuarry Studio is a Windows application that manipulates the meta-data objects and guides the developer through the creation and management of these objects.

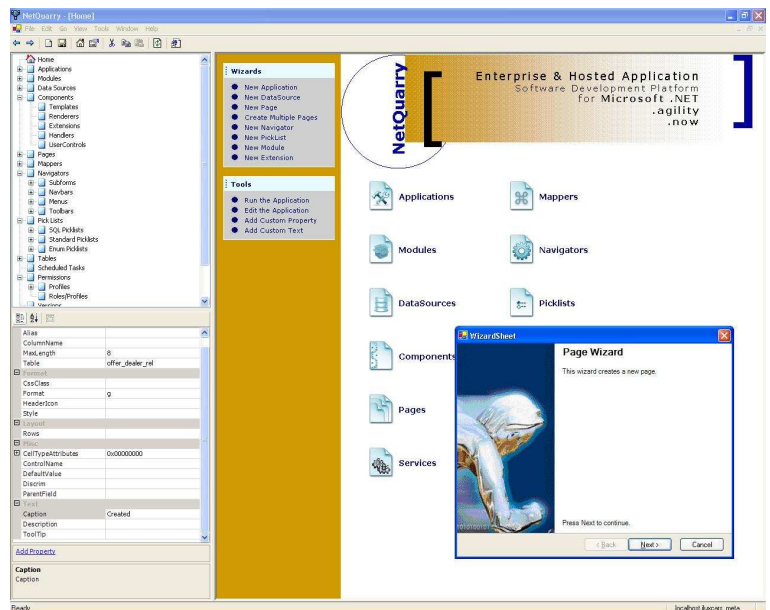
The Studio is built using both Microsoft .NET and the NetQuarry platform. It provides a Visual Studio developer with a familiar user interface further enhancing the productivity of developers.

NetQuarry Studio is not a replacement for Visual Studio. Rather, it is the main tool interface used to describe the application. Developers still use Visual Studio to write business logic using the .NET language of their choice.

Task Wizards

Most of the major tasks related to metadata configuration have associated task wizards to guide the application developer through the process of creating objects. Wizards exist for tasks that are as simple as adding a tab to a sub-form and as complex as creating an entire application.

In addition, there are a set of wizards that are accessible via Visual Studio that simplify the task of creating extensions, wizard pages, and custom form layouts.



Studio Features

The NetQuarry Studio features a simple to use object/property paradigm allowing the developer to work on metadata objects exactly as they would controls on a form in Visual Studio.

The Studio manages every aspect of the metadata development. The developer has a rich MSDN style help file available that describes the objects, their properties, and targeted code examples.

Metadata and Configuration Management

An application created with the NetQuarry platform uses metadata as the definition or description of the runtime application. At both design and run-time this metadata is stored in a relational database (e.g. SQL Server, Oracle, MySQL). The NetQuarry Studio reads and writes data to this operational store.

However, for several reasons, mainly the fact that a relational database is not a convenient format for source code control, the permanent format of the metadata is not a relational database. Instead, for configuration purposes, the "truth" of the metadata is stored in human readable text format.

When editing the metadata, the tools allow the developer to segment sections of the metadata into a grouping called a module. The module can then be saved to the file system and checked into a configuration management system. Creating a current snapshot of the metadata is accomplished quickly and simply by loading the application module files into an empty metadata schema via a command line database initialization tool.

The segmentation of the metadata allows multiple developers to effectively work on a single application, ensuring that metadata changes can be handled by the configuration management system at the same level as source code.

While editing, the studio tracks changes to each module and allows the developer to export only those modules that have been changed.

The NetQuarry Studio also allows for a user to add their own custom properties to an object. This allows even custom objects to be configured dynamically at runtime. For example, if you write an extension that is shared across multiple Mapper objects, you can add properties that apply to that extension to change the behavior of the extension for different objects.

In addition, all object permissions are controlled by the Studio. NetQuarry supports application security using a powerful, replaceable set of authentication and permission patterns. Depending on the needs of your application and the capabilities of your enterprise infrastructure, NetQuarry can support several different methods of user authentication.

Most objects use the NetQuarry.Security.ObjectPermissions enumeration to specify security for that object. The NetQuarry.Security.ObjectPermissions enumeration is a superset of all possible system permissions (and is extensible). The NetQuarry Studio is the management interface to all static permissions.

