Microsoft's Application Blocks

Enterprise Library 5.0

Microsoft Enterprise Library is a collection of application blocks designed to assist developers with common enterprise development challenges. Application blocks are a type of guidance, provided as source code that can be used "as is," extended, or modified by developers.

1. Caching Application Block: The Enterprise Library Caching Application Block lets developers incorporate a local cache in their applications. It supports both an in-memory cache and, optionally, a backing store that can either be the database store or isolated storage. The Caching Application Block can be used without modification; it provides all the functionality needed to retrieve, add, and remove cached data. Configurable expiration and scavenging policies are also part of the block. Parts of this block are built into .NET v4.0.
2. Cryptography Application Block: The Cryptography Application Block includes support for the following features. 1. Encryption algorithms, 2. Hashing algorithms, 3. Multiple cryptography providers, 4. Additional implementations of cryptography providers, 5. Key protection with the data protection API (DPAPI)
3. Data Access Application Block: The application block includes support for both stored procedures and inline SQL statements. Common housekeeping tasks, such as managing connections and creating and caching parameters, are encapsulated in the application block's methods. In other words, the Data Access Application Block provides access to the most often used features of ADO.NET in simple-to-use classes and provides a corresponding boost in developer productivity.
4. Exception Handling Application Block: Helps developers implement common design patterns and create a consistent strategy for processing exceptions. It is designed to support the typical code contained in **catch** statements in application components. The Block includes four exception handlers: 1. **Wrap handler**. Wraps one exception around another, 2. **Replace handler**. Replaces one exception with another. 3. **Logging handler**. Formats exception information, such as the message and the stack trace. Then the logging handler passes this information to the Enterprise Library Logging Application Block. 4. **Fault Contract exception handler**. Designed for use at Windows® Communication Foundation (WCF) service boundaries, and generates a new Fault Contract from the exception.
5. Logging Application Block: Simplifies the implementation of common logging functions. You can use the Logging Application Block to write information to a variety of locations: 1. The event log, 2. An e-mail message, 3. A database, 4. A message queue, 5. A text file, 6. A Windows® Management Instrumentation (WMI) event, 7. Custom locations using application block extension points
6. Policy Injection Application block: While the Policy Injection Application Block is still included in this release of Enterprise Library, it is (with the exception of one call handler) a set of legacy components such as the **PolicyInjection** facade that supports backwards compatibility with applications that use versions of Enterprise Library prior to version 5.0.
7. Security Application Block: The Security Application Block provides code that will help you with the following scenarios: 1. Authorization, and 2. Caching security-related credentials
8. Validation Application Block: Allow developers to implement structured and easy-to-maintain validation scenarios in their applications