

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Enterprise Library

如何创建应用程序块

patterns & practices

微软（中国）开发合作部

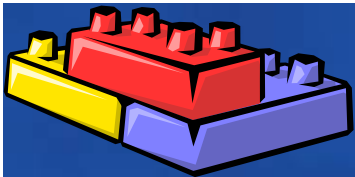
吴延安

.NET 首席顾问

yananwu@microsoft.com



MSDN Webcasts

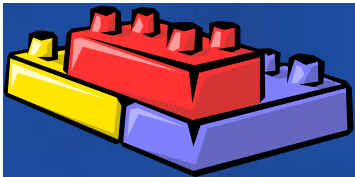


日程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 企业库及应用程序块
- 应用程序块的核心构成元素
- 创建应用程序块的步骤
- 问题



patterns & practices

面向企业级的架构指南

验证的
权威性
精准性
实用性
相关性

Based on field experience

Offer the best advice available

Technically validated and tested

Provide the steps to success

Address real-world problems based on customer scenarios

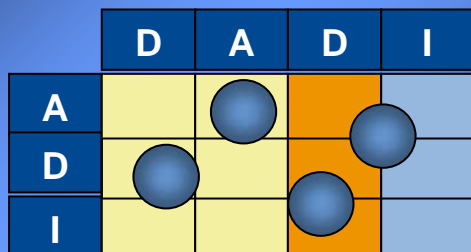
Available online: <http://www.microsoft.com/practices>

Books available: <http://www.amazon.com/practices>



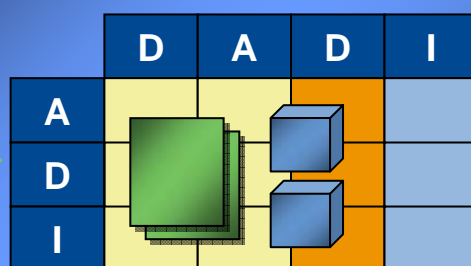
Patterns

对反复出现的问题提供原子级的解决方案



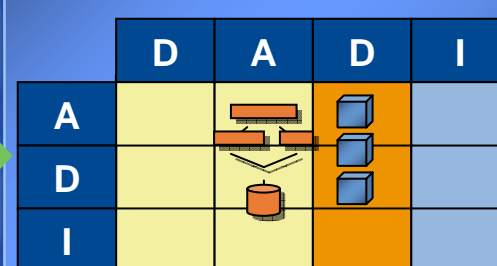
Application Blocks

面向公共服务的子系统级应用指南



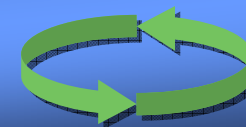
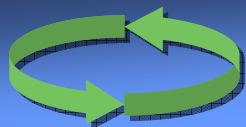
Reference Architectures

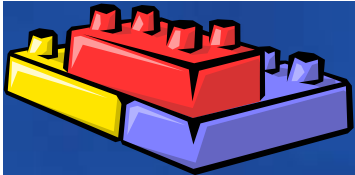
针对客户向通场景的系统级架构指南



指导方针

提供有关架构设计原则以及诸如安全与性能等综合问题的指导



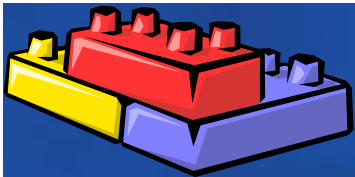


Sound familiar?

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 复制、粘贴进行代码的共享
- 通过组件模型实现重用...为了满足特殊的需求不断的改写这些组件
- 希望这些组件能更好的工作在一起...
- 想更好地解决这些组件在设计及使用上的不一致性
- 渴望有一个自己或企业的应用支撑软件库

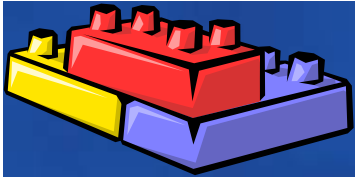


应用程序块新的设计理念

您的动力，我们的动力

Microsoft
微软(中国)有限公司

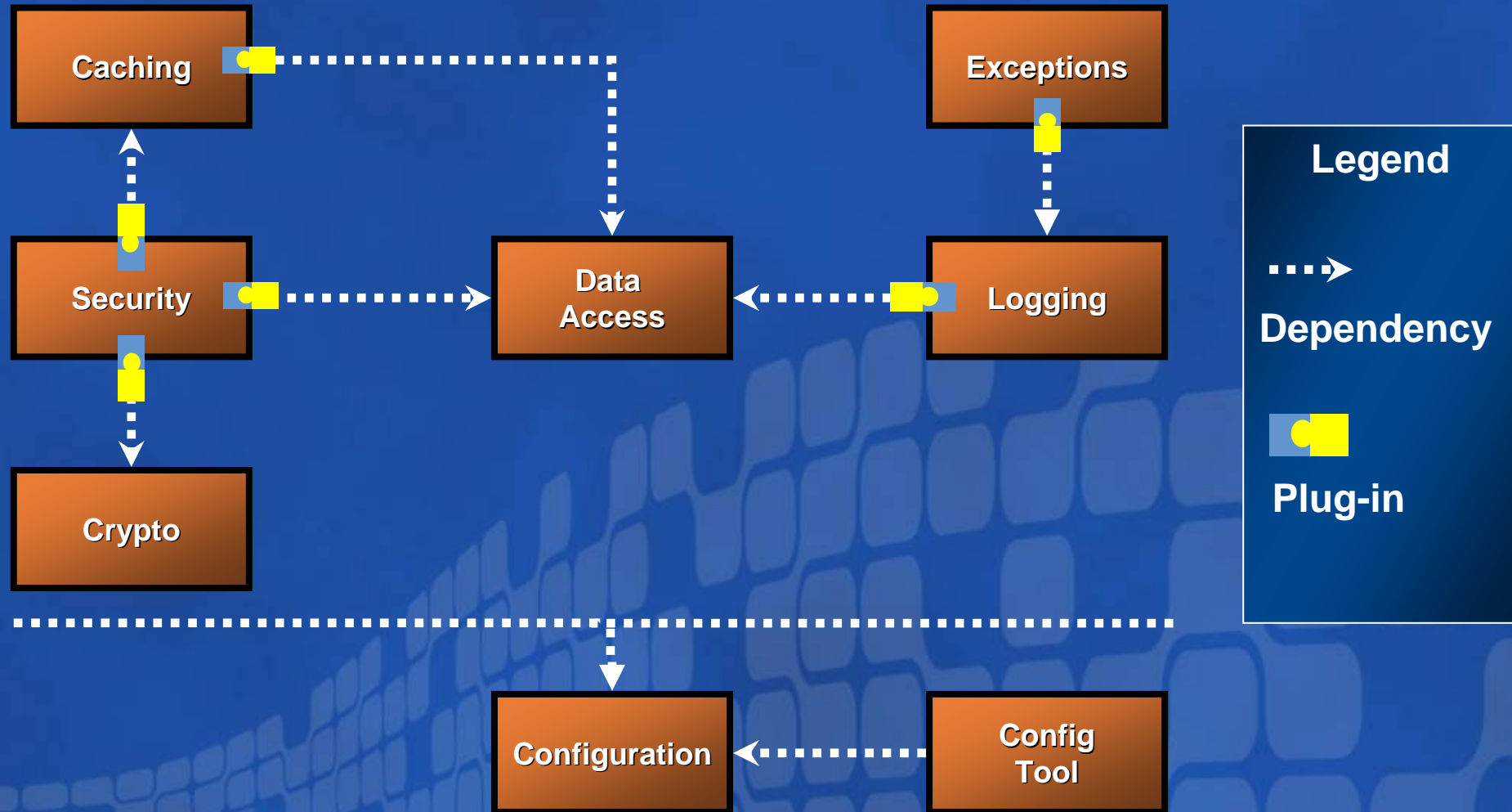
- 一致性
 - 采用了一致的设计模式和实施方法
- 扩展性
 - 不仅允许通过修改代码的方法、还提供了插接件的方式允许开发人员编写客户化代码外挂到应用程序块上进行扩展
- 易用性
 - 充分利用图形化Configuration工具
 - 提供了简明、完整的文档及快速启动案例
 - 更加简单的安装手段
- 集成性
 - 各程序块设计时要能相互工作并经测试确认，并且各程序块还要求能独自工作。

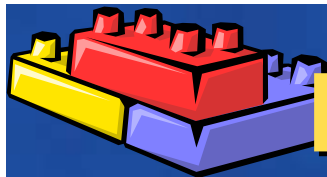


Enterprise Library v1

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

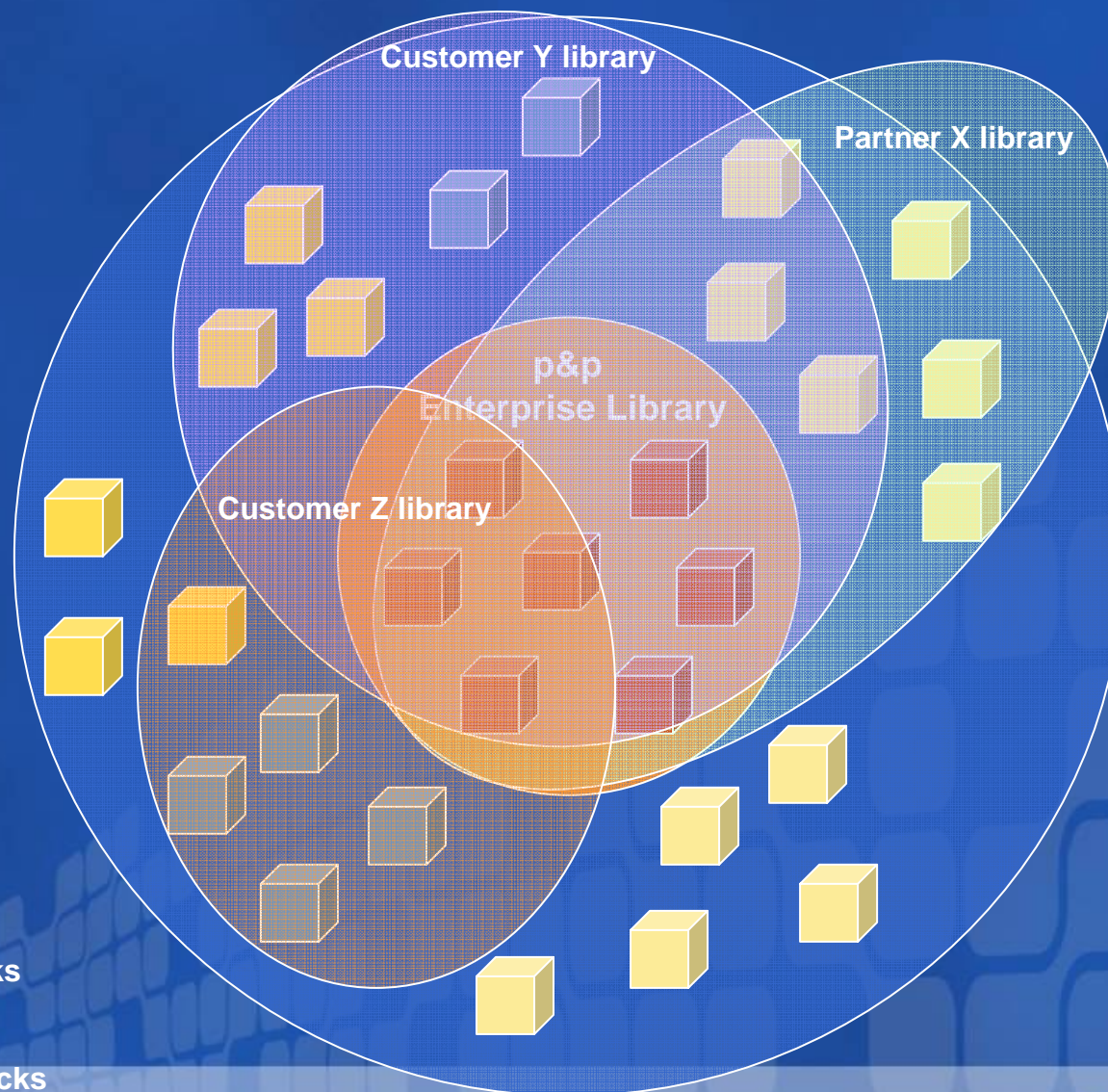








Enterprise Library 远景

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



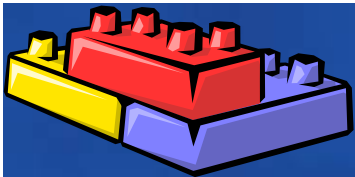
-  p&p blocks
-  Partner blocks
-  Customer blocks
-  Community blocks



Block Specification

MSDN Webcasts





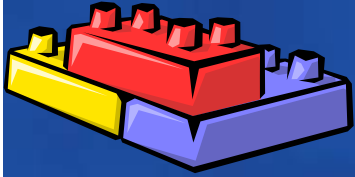
应用程序块的构成元素

您的潜力, 我们的动力

Microsoft®

微软(中国)有限公司

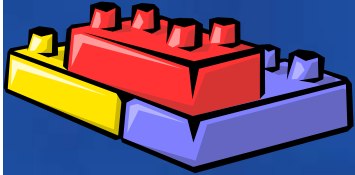
- Core functionality (核心功能)
- Pluggable providers (外部可插接的扩展器)
- Factories (工厂模式Factory的实现, 创建Provider)
- Configuration (提供灵活性及可管理性的配置)
- Unit tests (单体测试)
- Quick Starts (快速启动案例)
- Guidance and reference documentation (参考文档)



Core Functionality (核心功能)

您的程序，优化的动力
Microsoft
微软(中国)有限公司

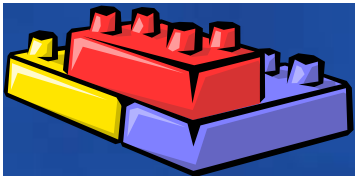
- 程序块需提供的核心功能、是操作的中心
- 可以受configuration设置的影响，但基本行为不变，除非对源代码进行更改
- 核心操作功能是通过类/接口/外观 (facade) 与应用程序打交道
- 核心操作功能- 并非都要设计成可配置或可替换的



Core Functionality Samples

您的潜力 我们的动力
Microsoft
微软(中国)有限公司

- Caching (缓存)
 - Scavenging algorithm – not exposed
 - Cache implementation (hashtable) – not exposed
 - Cache manager – exposed as a class
- Exception Handling (例外处理)
 - Exception policy – exposed as a class with single public static method (HandleException)
- Logging and Instrumentation (日志及仪表盘)
 - Log filters – not exposed



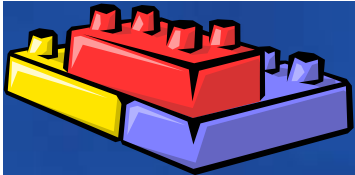
Pluggable Providers

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

(可插接的扩展器)

- 一个Provider对应“Provider type”的一个特殊实现，“Provider type”定义了应用程序块必须正确执行的功能接口
 - Caching: Backingstore
 - Data Access: Database
- 每一个应用程序块对其任一“Provider type”至少应提供一个或多个实现.
 - Caching: Database and isolated storage backingstores
 - Data Access: SQL, DB2 and Oracle databases
 - Security – Authorization: AzMan,Rule,etc.
 - Security – Roles: Active directory, role based db..



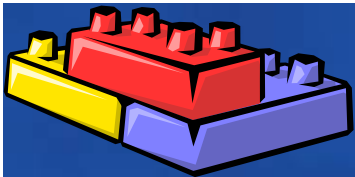
Pluggable Providers

(可插接的扩展器)

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 降低具体实现与核心功能块间的耦合性, 达到以下目的:
 - 多样性
 - 扩展性
 - 封装性
 - 不同环境间的可移植性
 - 各应用程序块间的最小耦合性
- 在配置文件中通过类型信息及名称进行指定

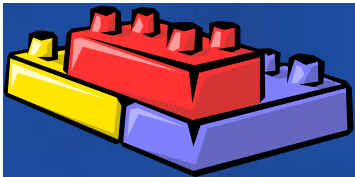


Provider 的实现与配置

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- Provider的实现
继承ConfigurationProvider及该应用程序块的provider接口(如INotificationProvider)
- 抽象基类ConfigurationProvider
定义了如何通过“configuration信息”对一个provider进行初始化的规则
- Providers通常允许对每个特殊的实现进行配置
例如: 可以通过在配置文件中指定一个新的exception type类型, 达到对原有的exception handler进行替换的目的

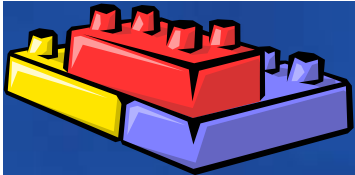


Provider Factories

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 使用Martin Fowler大师的Plugin pattern创建Provider
- Factory属于创建型设计模式, 强调了减少模块间耦合程度的设计原则
- 每一个Provider type都存在一个创建Provider对象的Factory
 - 调用者通过Factory得到需要的Provider对象
 - Factory决定如何创建对象
 - 调用者不需关心对象是如何生成的
- Factory类均有抽象基类ProviderFactory派生
- 抽象基类ProviderFactory存在于Configuration Application Block中, 提供了创建Providers实例的必要代码, 创建的具体信息由配置文件的设置决定



Provider Factories

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Block也使用静态方法封装provider实体的创建
- 允许创建默认的provider (默认provider将在configuration 中指定)
- 也允许创建用字符串名称表示的provider

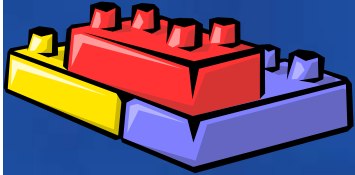
' Create the default database instance

```
Dim db As Database = DatabaseFactory.CreateDatabase()
```

' Use a named instance to map to configuration

```
Dim salesDb As Database =  
DatabaseFactory.CreateDatabase("Sales")
```

- 允许重新配置而不需进行重编译

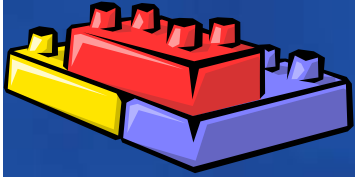


Configuration

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 允许方便地对程序块进行配置已实现灵活/扩展性的要求
 - 人们可以根据企业应用的不同使用场景, 通过配置的变更达到自适应的目的
 - 随着应用复杂性的增强, 应用程序块可以灵活扩展以适应这种变化的需求
- 统一的配置管理工具-- Configuration Console 用于对创建及编辑配置项
- 配置项—可以用来定义核心功能每一个provider的类型

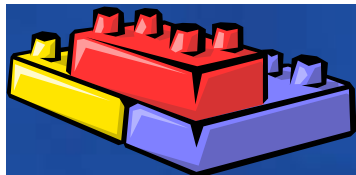


Configuration

您的潜力, 我们的动力

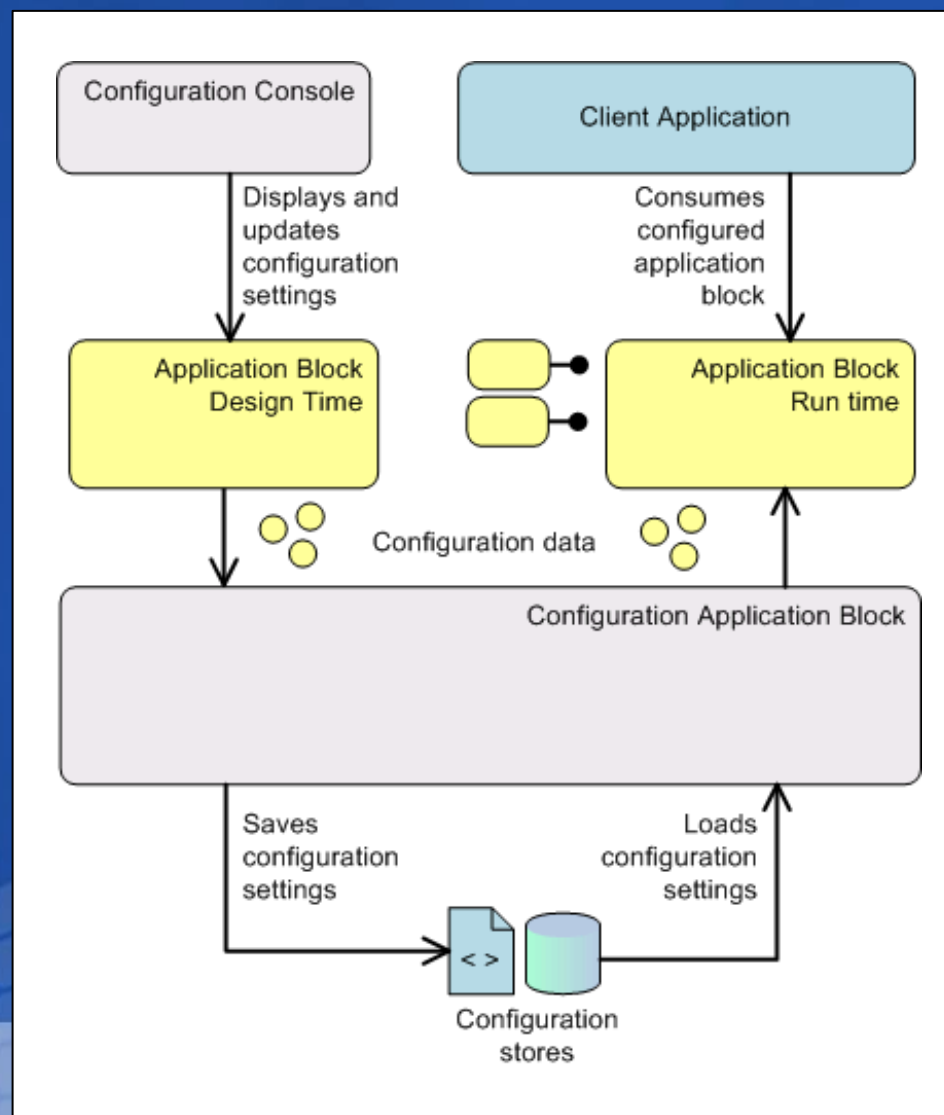
Microsoft
微软(中国)有限公司

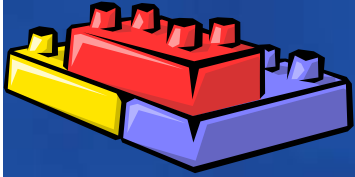
- 运行时配置 (Runtime configuration)
 - 通过configuration对象对配置信息进行访问
 - Configuration Application Block从storage读取配置信息、并向应用返回一个配置实体
 - XML File Storage Provider and Transformer 需要对对象能做序列化处理
- 设计时配置 (Design-time configuration)
 - 通过Configuration Console 展示、变更及验证配置信息



Design-time and Runtime Relationship

你的潜力，我们的动力
Microsoft
微软(中国)有限公司

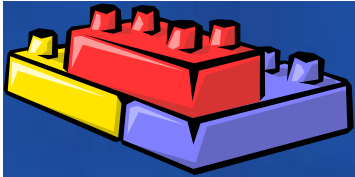




设计时配置 (configuration)

您的活力 我们的动力
Microsoft
微软(中国)有限公司

- 基于 System.ComponentModel 模型
- configuration tree 上的每一个 node 对应着一个相关的类
- 每一个 node 决定着该 node 被显示的属性
- 任何一个 node 可以包含验证规则

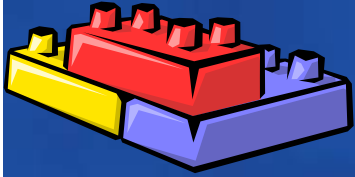


Configuration Nodes

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

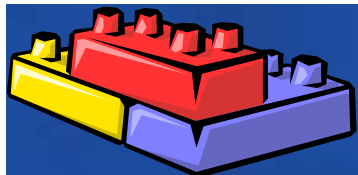
- 决定Configuration Console如何显示configuration nodes的层次化结构
- 决定哪些属性以及如何被暴露于可视化的图形接口上
- 允许Configuration Console对用户为程序块设置的配置属性进行有效性校验
- 能够包含一个程序块对另一个程序块进行依赖是的参照点 (如基于数据库的授权将指向一个数据库的实例)
- 包含指向运行时数据类的参照



Configuration Design Manager

你的潜力，我们动力
Microsoft
微软(中国)有限公司

- 实现 IConfigurationDesignManager 接口
- 负责装载及保存配置数据
- 允许应用程序块被添加到配置层次结构中
- 创建设计时的交互菜单及命令，用于用户使用 Configuration Console 进行应用程序块的配置
- Assembly 必须与 Configuration Console 置于同一运行目录下



Configuration Console

您的潜力, 我们的动力

Microsoft®

微软(中国)有限公司

Configuration Nodes

Properties

The screenshot shows the 'Enterprise Library Configuration*' window. The left pane displays a tree view of configuration nodes. The right pane shows the properties for the selected 'Inventory' node.

Configuration Nodes Tree:

- Enterprise Library Configuration
 - Application1
 - Caching Application Block
 - Cache Managers
 - Cache Manager
 - Configuration Application Block
 - Data Access Application Block
 - Connection Strings
 - Sales Production
 - Database Instances
 - Inventory
 - Sales
 - Database Types

Properties Pane (General):

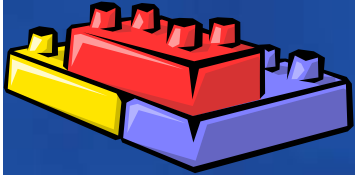
Property	Value
ConnectionStringNode	Sales Production
DatabaseTypeNode	Sql Server
Name	Sales

Configuration Errors:

Name	Property	Description	Path
Inventory	DatabaseTypeNode	The value for DatabaseTypeNode can not be null.	Application1/Data Access Application Block/D...

Ready

Validation Error

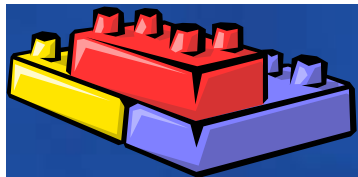


单元测试 (Unit Tests)

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

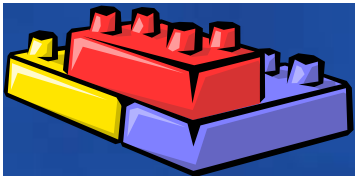
- 验证application block的功能性
- 验证对 application block进行的修改没有导致其他的质量问题
- Enterprise Library使用了测试驱动的开发方法 (Test Driven Development)



快速启动案例 (Quick Starts)

Microsoft
微软(中国)有限公司

- 用于演示应用程序块中通用场景的使用方法

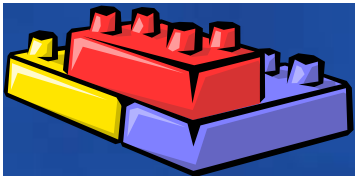


指导文档

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

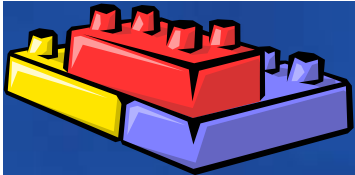
- 与Visual Studio紧密集成
- 提供指导
 - Introduction and goals
 - Developing applications using the application block
 - Design of the application block
 - Extending the application block
 - Deploying the application block
 - Quick start walkthroughs
- API参考
 - Public and protected
 - Summaries, parameters and return values



应用程序块的创建步骤

您的活力, 我们的动力
Microsoft
微软(中国)有限公司

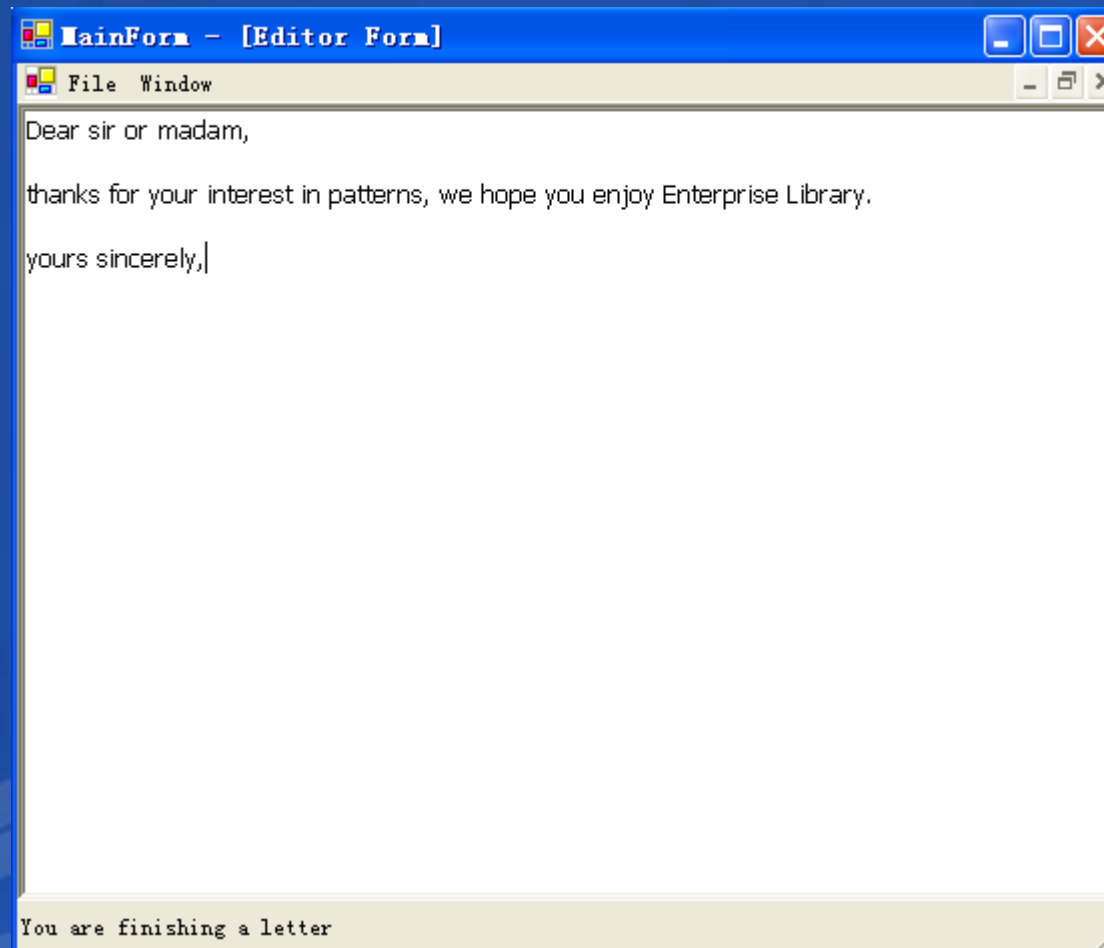
- 抽取可重用部分的功能块
- 创建应用程序块
 - 搭建应用程序块框架, 建议使用Enterprise Template
 - 定义core functionality, 即provider interface
 - 创建configuration data类
 - Create the configuration data for the block
 - 创建configuration运行时支持
 - Create the configuration view
 - Create the notification provider factory
 - Create a static Factory class for one-line creation of providers
 - 实现一个或多个特定的 provider
 - 创建configuration设计时支持
- 发布“Configuration设计时支持”的Assembly

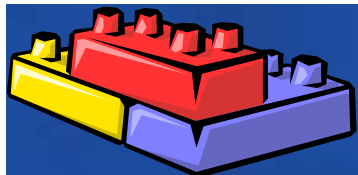


您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

抽取可重用部分的功能块 – Notification



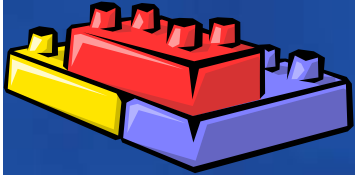


A Simple Example: Notification Application Block

您的潜力, 我们的动力

Microsoft®
微软(中国)有限公司

- 目标: 应用程序将使用notification application block对将编辑过程中检测到的信息通知到桌面
- 信息的通知机制将被封装在 *notification provider* 中
- 应用程序可以使用不同的*notification providers*, 每一个provider 均有唯一的名字确定
- 如果应用程序不通过名称指定provider时, 将使用 Configuration Console设置的默认的 notification provider,
- 每一个notification provider都有信息显示时间的设置, 该时间将由Configuration Console进行设置

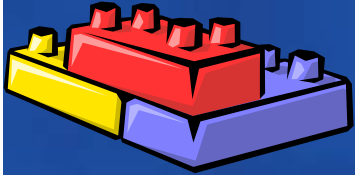


Notification Application Block

Core Functionality

您的潜力 我们的动力
Microsoft
微软(中国)有限公司

- 包含了接口的定义 `INotificationProvider`, 所有将要扩展的provider均要对该接口进行实现
- 包含一个叫 `NotificationConfigurationView` 的类
 - 继承了 `ConfigurationView` 类
 - 提供了运行时provider的配置信息获取接口
 - 将作为provider创建时的初始化参数使用

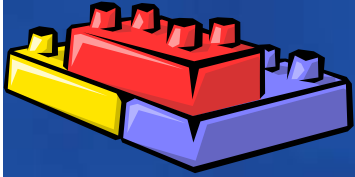


Notification Application Block Configuration View

您的潜力 我们的动力
Microsoft
微软(中国)有限公司

- Initialized with ConfigurationContext

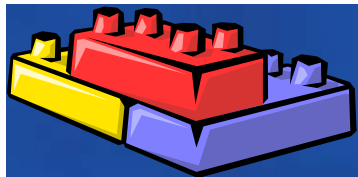
```
public virtual NotificationProviderData GetNotificationProviderData(string
providerName)
{
    // HelloWorldSettings 是运行时的配置对象
    NotificationSettings settings = GetNotificationSettings();
    NotificationProviderData sinkData = settings.NotificationProviders[providerName];
    ....
    return sinkData;
}
```



Notification Application Block

您的潜力 我们的动力
Microsoft
微软(中国)有限公司

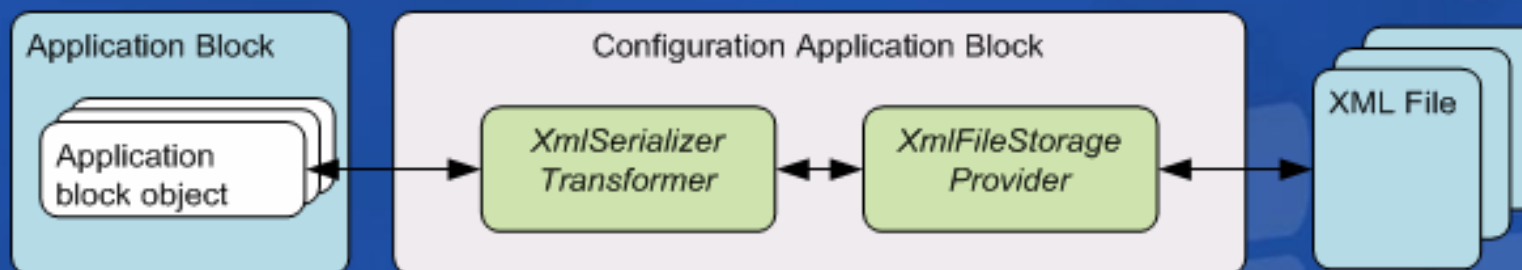
- Custom Notification provider
 - 如果外部有对INotificationProvider 进行了实现的类，允许设计时指定该类作为provider
 - 配置是通过 name-value pairs（没有设计时支持）的方式进行指定

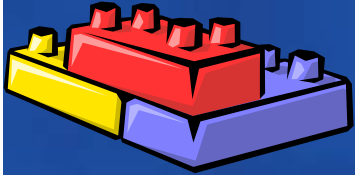


Notification Application Block Runtime Configuration

您的潜力 我们的动力
Microsoft
微软(中国)有限公司

- 配置信息由经过序列化的classes表示





Runtime Configuration Class Serialization

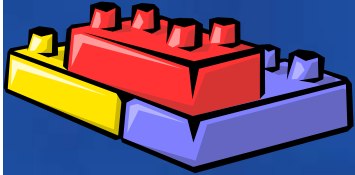
我们的动力
Microsoft
微软(中国)有限公司

- Root object serialization

```
[XmlRoot("notificationSettings",  
Namespace=NotificationSettings.ConfigurationNamespace)]  
public class NotificationSettings  
{  
...  
}
```

- Collection serialization

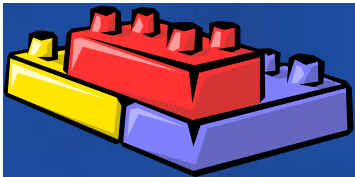
```
[XmlArray(ElementName="notificationProviders")]  
[XmlArrayItem(ElementName="notificationProvider",  
Type=typeof(NotificationProviderData))]  
public NotificationProviderDataCollection NotificationProviders  
{  
    get { return _notificationProviders; }  
}
```



Runtime Configuration Data

您的动力，我们的动力
Microsoft
微软(中国)有限公司

```
<?xml version="1.0" encoding="utf-8"?>
<notificationConfiguration>
  <xmlSerializerSection
    type="FourthCoffee.Framework.Notification.Configuration.NotificationSettings,
    FourthCoffee.Framework.Notification, Version=1.0.2026.31777, Culture=neutral,
    PublicKeyToken=null">
    <notificationSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      defaultProviderName="Custom Notification Provider"
      xmlns="http://www.fourthcoffee.com/framework/03-01-2005/notification">
      <notificationProviders>
        <notificationProvider xsi:type="CustomNotificationData" name="Custom Notification
        Provider" type="FourthCoffee.Framework.Notification.StatusBarNotificationProvider,
        FourthCoffee.Framework.Notification, Version=1.0.1913.41492, Culture=neutral,
        PublicKeyToken=null">
          <Attributes name="DisplayTimeout" value="2000" />
        </notificationProvider>
      </notificationProviders>
    </notificationSettings>
  </xmlSerializerSection>
</notificationConfiguration>
```

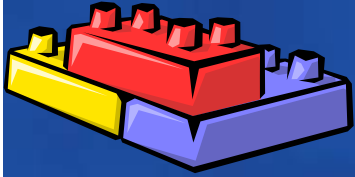


Runtime Configuration Classes

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

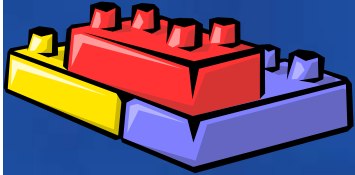
- NotificationSettings
 - 包含configuration section name, providers集合, 以及默认的Notification provider (如前页配置)
 - NotificationProviderData(base class),
 - 一抽象基类, 所有的Provider都将继承该类
 - 目的:
 - 可以包括任何公共的配置信息
 - 是进行序列化时指定XmlInclude attributes的场所
 - NotificationProviderDataCollection
 - NotificationProvider的集合,用于维护Provider列表
 - CustomNotificationProviderData
 - 负责持有Provider type信息
- 提供了弱类型设计时配置的支持



Design-Time Configuration Classes

你的潜力 我们的动力
Microsoft
微软(中国)有限公司

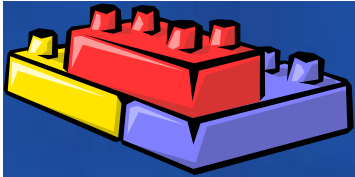
- NotificationSettingsNode
 - 对应Notification根节点
- NotificationProviderNode
 - NotificationProviderData在设计时的表现形式
 - 每一个Provider Type均有一个ProviderNode用于对配置项进行操作
- CustomNotificationProviderNode
 - 用于操作CustomNotificationData对象及属性
- NotificationConfigurationDesignManager
 - 向Configuration Console中装载或保存配置信息
 - 在相应的节点上创建Menu及Command



Notification Design-Time Configuration发布

你的潜力, 我们的动力
Microsoft
微软(中国)有限公司

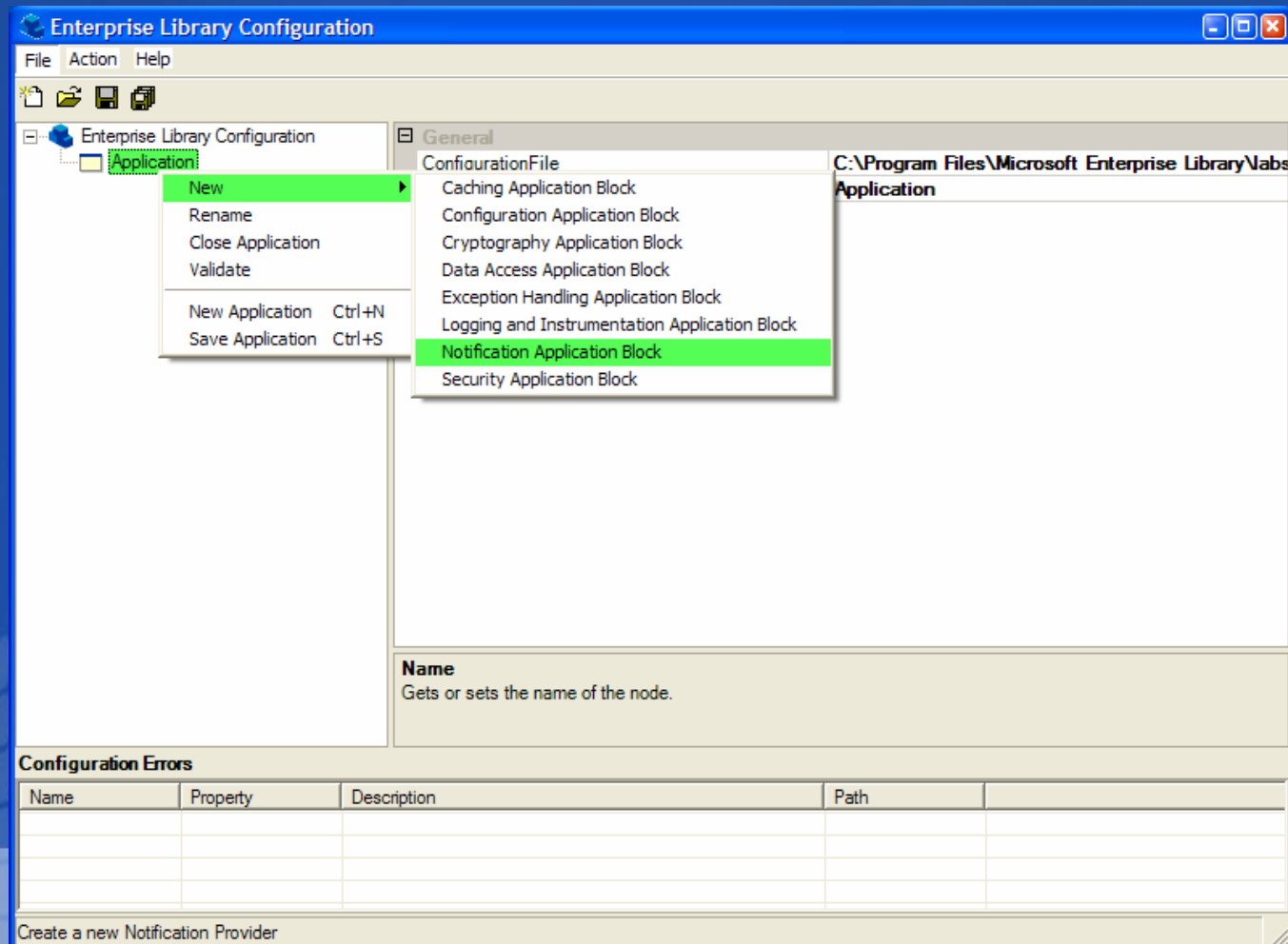
- 为了让configuration console能使用Notification application block设计时的支持
 - 使用assembly属性声明DesignManager类型
在assembly.cs中作如下声明
[assembly :
ConfigurationDesignManager(typeof(NotificationCon
figurationDesignManager))]
将编译好的design time support assembly拷贝到
configuration console的运行目录下

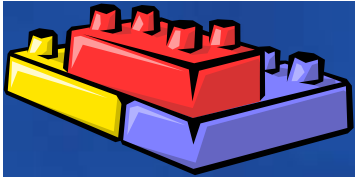


Notification Design-Time Configuration

你的潜力, 我们的动力
Microsoft
微软(中国)有限公司

允许设计时对Notification application block进行设置

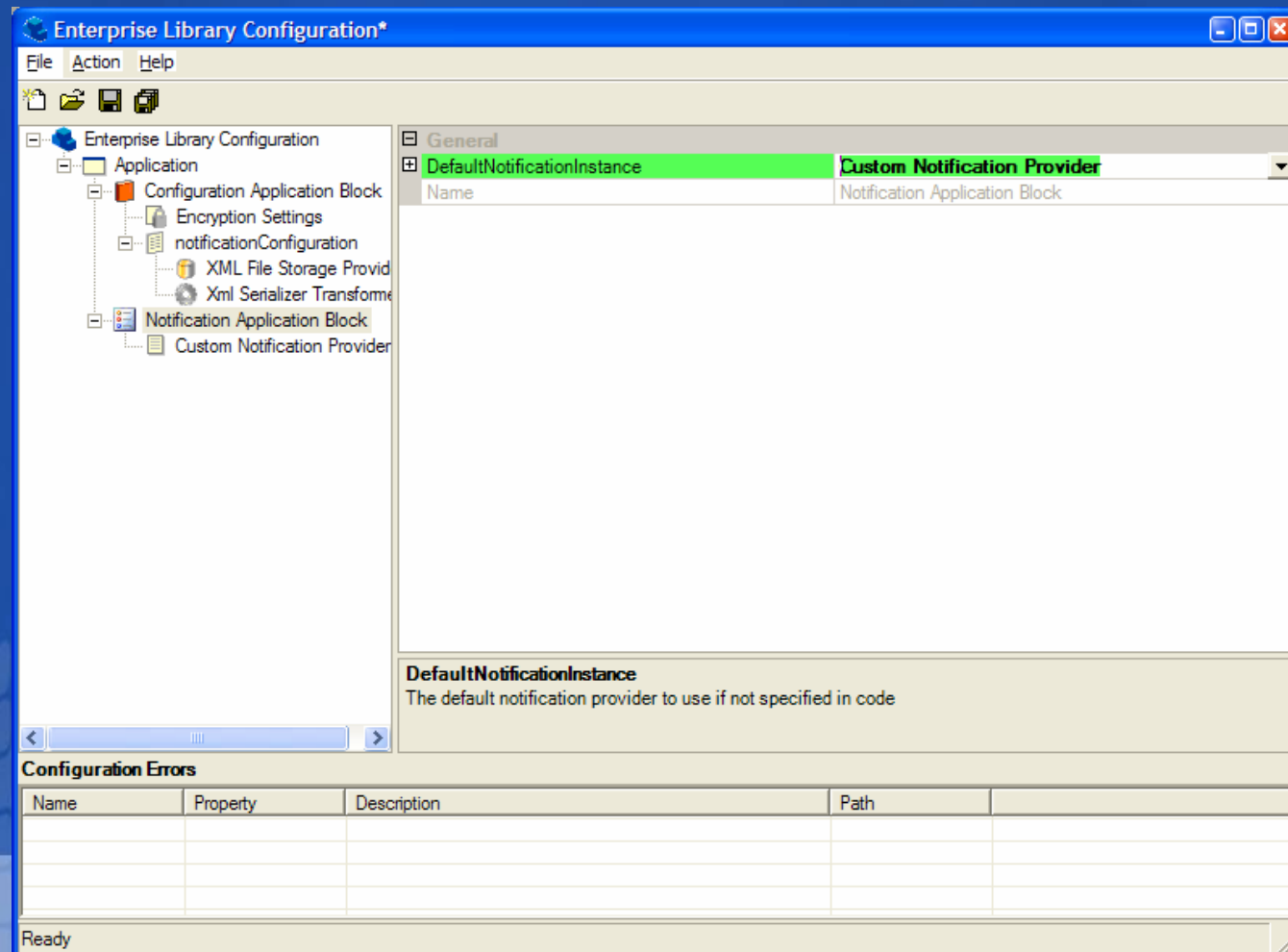


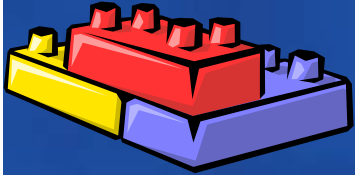


Notification Design-Time Configuration

你的潜力，我们的动力
Microsoft®
微软(中国)有限公司

- 允许设计时对Notification application block进行设置





Notification Application block的使用

你的潜力, 我们的动力
Microsoft
微软(中国)有限公司

- Create provider

```
// Create default greeting provider (determined by configuration)
```

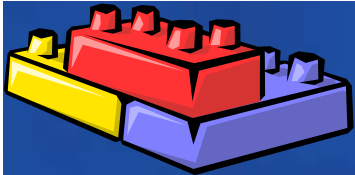
```
INotificationProvider provider = NotificationFactory.GetNotificationProvider();
```

```
// Create named instance
```

```
INotificationProvider provider =  
NotificationFactory.GetNotificationProvider("Custom Notification  
Provider");
```

- Call provider methods

```
provider.DisplayMessage(this, description);
```



Microsoft
patterns & practices
proven practices for predictable results

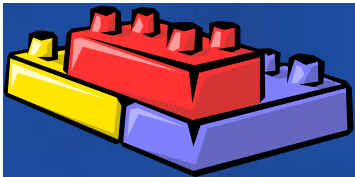
Download it
Today!

Announcing: Enterprise Library 1.0

<http://www.microsoft.com/practices>

<http://www.microsoft.com/downloads/details.aspx?familyid=0325b97a-9534-4349-8038-d56b38ec394c&displaylang=en> (EntLib download)

<http://workspaces.gotdotnet.com/entlib>

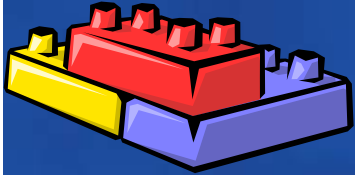


Webcasts

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Webcasts China
- 2005/5/20 – patterns & practices Live:企业库系列讲座(1): Entlib概述
- 2005/5/27 – patterns & practices Live:企业库系列讲座(2): 配置管理应用程序块
- 2005/6/3 – patterns & practices Live:企业库系列讲座(3): 数据访问应用程序块
- 2005/6/10 – patterns & practices Live:企业库系列讲座(4): 安全应用程序库
- 2005/6/17 – patterns & practices Live:企业库系列讲座(5): 日志和仪表盘管理应用程序块
- 2005/6/24 – patterns & practices Live:企业库系列讲座(6): 缓存应用程序块



Additional Resources

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Improving Web Application Security

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/ThreatCounter.asp>

- Improving .NET Application Performance and Scalability

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/scalenet.asp>

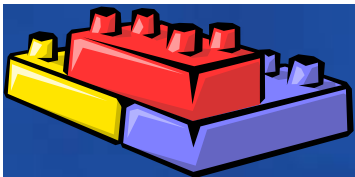
- Application Architecture for .NET

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/distapp.asp>

- PatternShare.org

- Enterprise Library Community

<http://go.microsoft.com/fwlink/?linkid=39209&clcid=0x09>



您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

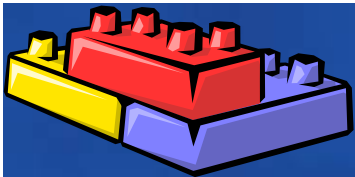
Microsoft®

Your potential. Our passion.™



msdn


MSDN Webcasts




Q&A


您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)