您的潜力，我们的动力

Microsoft
微软（中国）有限公司

Enterprise Library
Cryptography Application Block

曹严明

.NET架构顾问

Microsoft (China)

msdn

msdn
Microsoft Developer Network
MSDN

日程

- Enterprise Library 概述
- Crypto Application Block 概述
- 进一步的讨论
- Q & A

您的潜力，我们的动力

Microsoft
微软（中国）有限公司

Enterprise Library 概述

msdn

MSDN Webcasts

什么是 Enterprise Library

- 一组可重用的应用程序块，用于解决企业级应用开发过程中所面临的共性的问题
  - 配置管理
  - 日志管理
  - 异常处理
  - 数据访问
  - 缓存机制
  - 加密机制
  - 安全机制
- 好处
  - 重用
  - 最佳实现
  - 一致性
  - 易用性
  - 可扩展性

Enterprise Library 的构成

- 源代码
- 示例应用程序
- 文档
- 免费下载

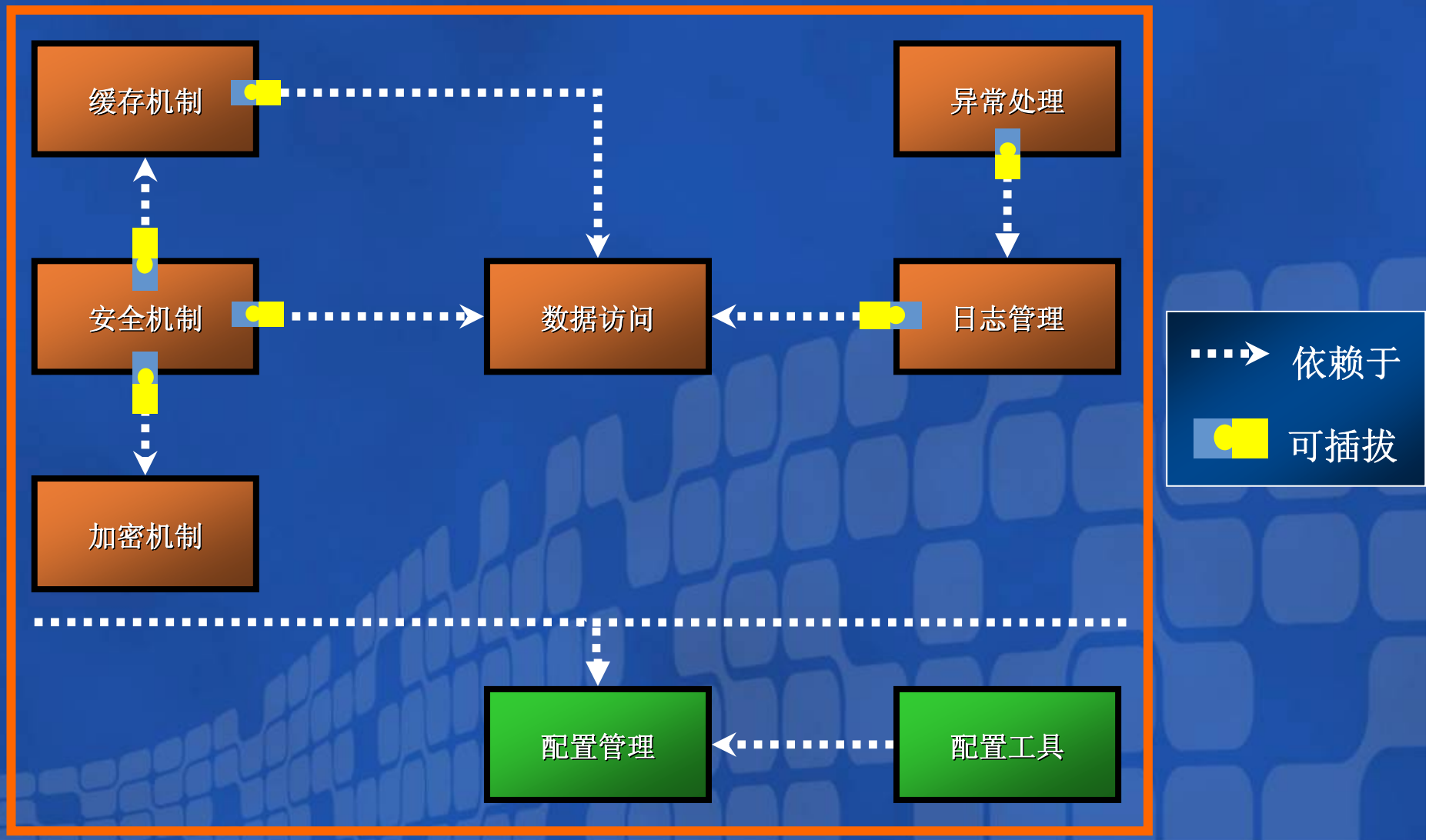
  http://www.microsoft.com/practices
- 社区支持

  http://workspaces.gotdotnet.com/entlib
- June 2005 release

  A minor update of January 2005 release
- Next release

  Target .NET 2.0 and Visual Studio 2005

Enterprise Library 1.0



您的潜力，我们的动力

Microsoft
微软（中国）有限公司

Crypto App Block 概述

msdn

MSDN Webcasts

你在开发中曾经遇到这些问题吗？

- 重复编写有关 cryptography 的代码 (streams, 初始化向量, 字符串到字节数组的转换， 等等)
- 搞不清楚该用哪个算法
- 改变算法需要重新编译代码
- 搞不清楚该如何管理 cryptography keys
- 搞不清楚如何正确使用 System.Security.Cryptography 类

Cryptography 方面的需求

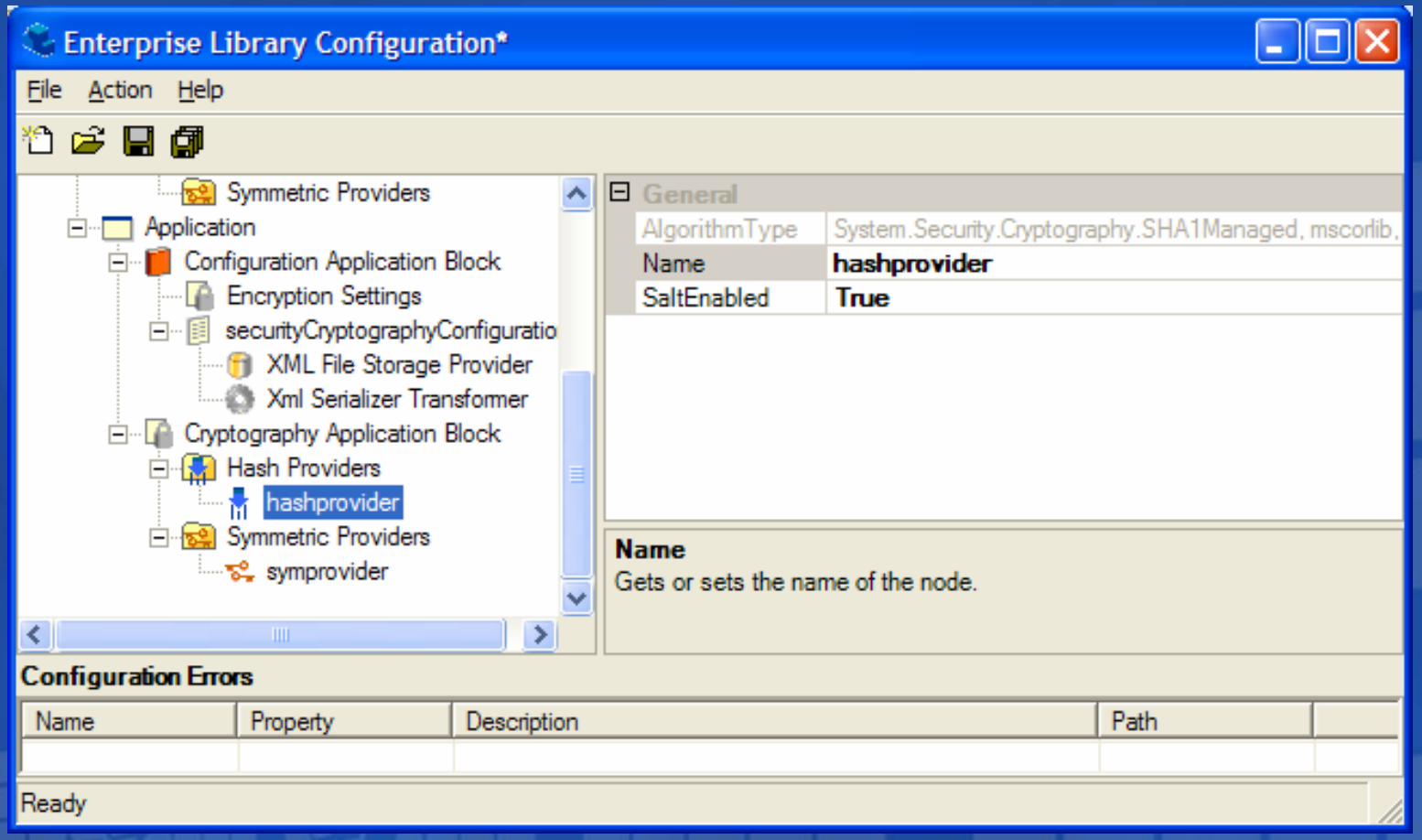
- 一个简单的方法哈希数据和比较哈希值
- 一个简单的方法加密和解密数据
- 可以在一台机器上加密信息而不必使用密钥
- 可以对同样的应用使用不同的 cryptography providers
- 可以方便的调整有关 cryptography 的配置

Cryptography Application Block

- 提供了一个简单的方法实现常见的 cryptography 场景
- 提高了应用的安全性
  - 考虑安全威胁和反措施
  - 方便使用
  - 与其它 application blocks 集成

Step 1: 配置

- 使用配置工具为 Cryptography Application Block 创建配置



Step 2: 调用相应的 Cryptography 方法

- 使用 Symmetric Provider 对数据加密

```
// Encrypt using the named provider
byte[] valueToEncrypt = Encoding.Unicode.GetBytes("password");
byte[] encryptedContents =
   Cryptographer.EncryptSymmetric("symmProvider", valueToEncrypt);

// Clear the byte array memory that holds the password
Array.Clear(valueToEncrypt, 0, valueToEncrypt.Length);

// Convert the value so that it can be displayed
string encryptedText = Convert.ToBase64String(encryptedContents);
```

- 解密数据

```
// Decrypt using the named provider
byte[] decryptedContents =
   Cryptographer.DecryptSymmetric("symmProvider", encryptedText);

// Convert the value so that it can be displayed
string plainText = Encoding.Unicode.GetString(decryptedContents);
```

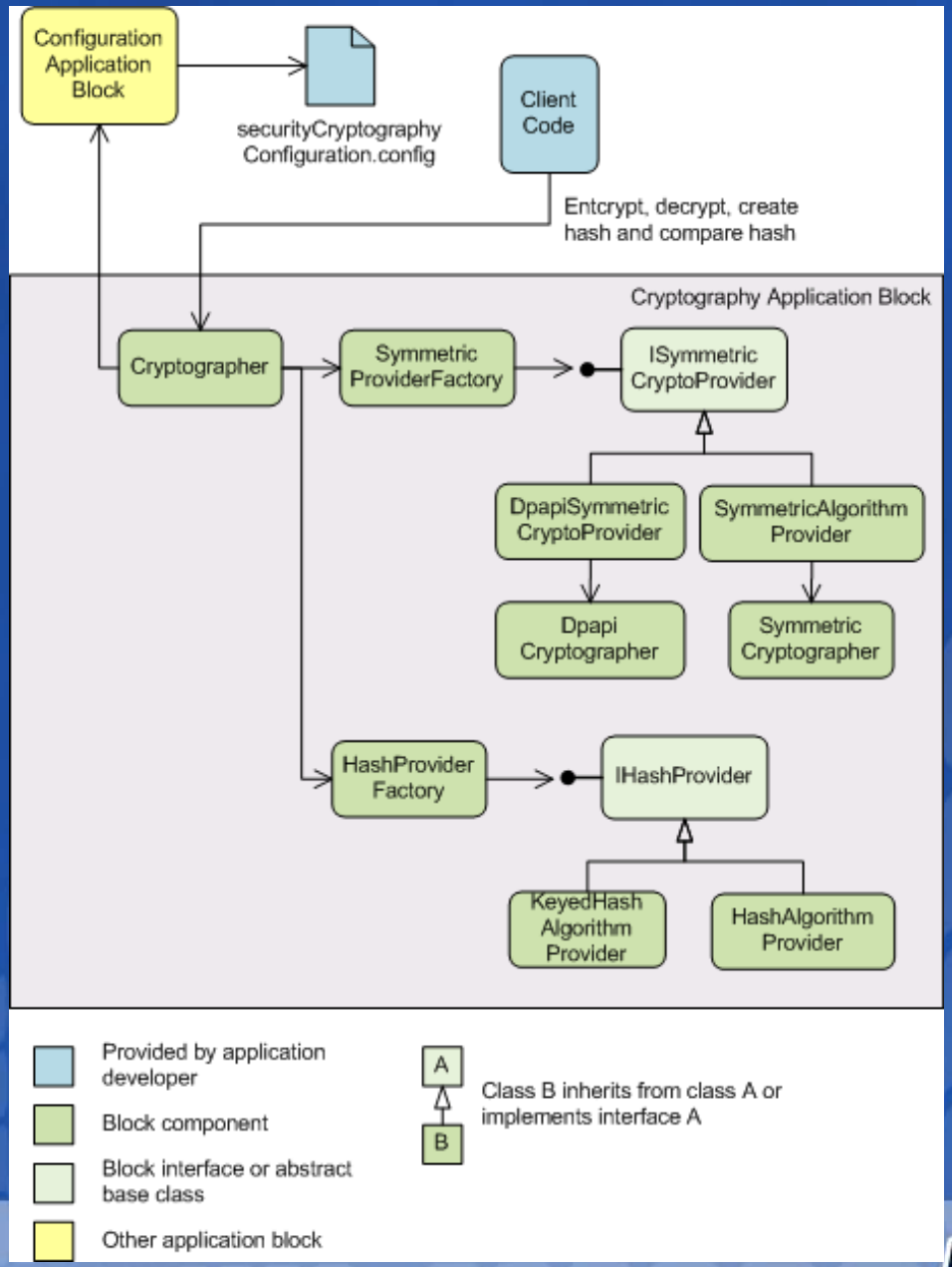
产生哈希值

```
// Generate a hash value using the named provider
byte[] valueToHash = Encoding.Unicode.GetBytes(“password”);
byte[] generatedHash =
    Cryptographer.CreateHash(“hashProvider”, valueToHash);

// Clear the byte array memory
Array.Clear(valueToHash, 0, valueToHash.Length);
```

检验哈希值是否匹配

```
// Generate a hash value using the named provider
byte[] valueToHash = Encoding.Unicode.GetBytes(yourPwd);
bool matched = Cryptographer.CompareHash(“hashProvider”, valueToHash,
    existingHashValue);

// Clear the byte array memory
Array.Clear(valueToHash, 0, valueToHash.Length);
```

Cryptography Application Block 体系结构



进一步的讨论

关于秘密信息的存储问题

- 常见的秘密信息包括:
  - SQL connection strings
  - Credentials used for SQL application roles
  - Fixed identities in Web.config
  - Process identity in Machine.config
  - Keys used to store data securely
  - SQL Server session state
  - Passwords used for Forms authentication against a database

秘密信息存储的各种方案

- 采用 Windows 平台提供的方案
  - .NET cryptography classes
  - Data Protection API (DPAPI)
  - CAPICOM
  - Crypto API
- 或者采用Cryptography Application Block
  - 更简单方便
  - 最佳实践

加密算法

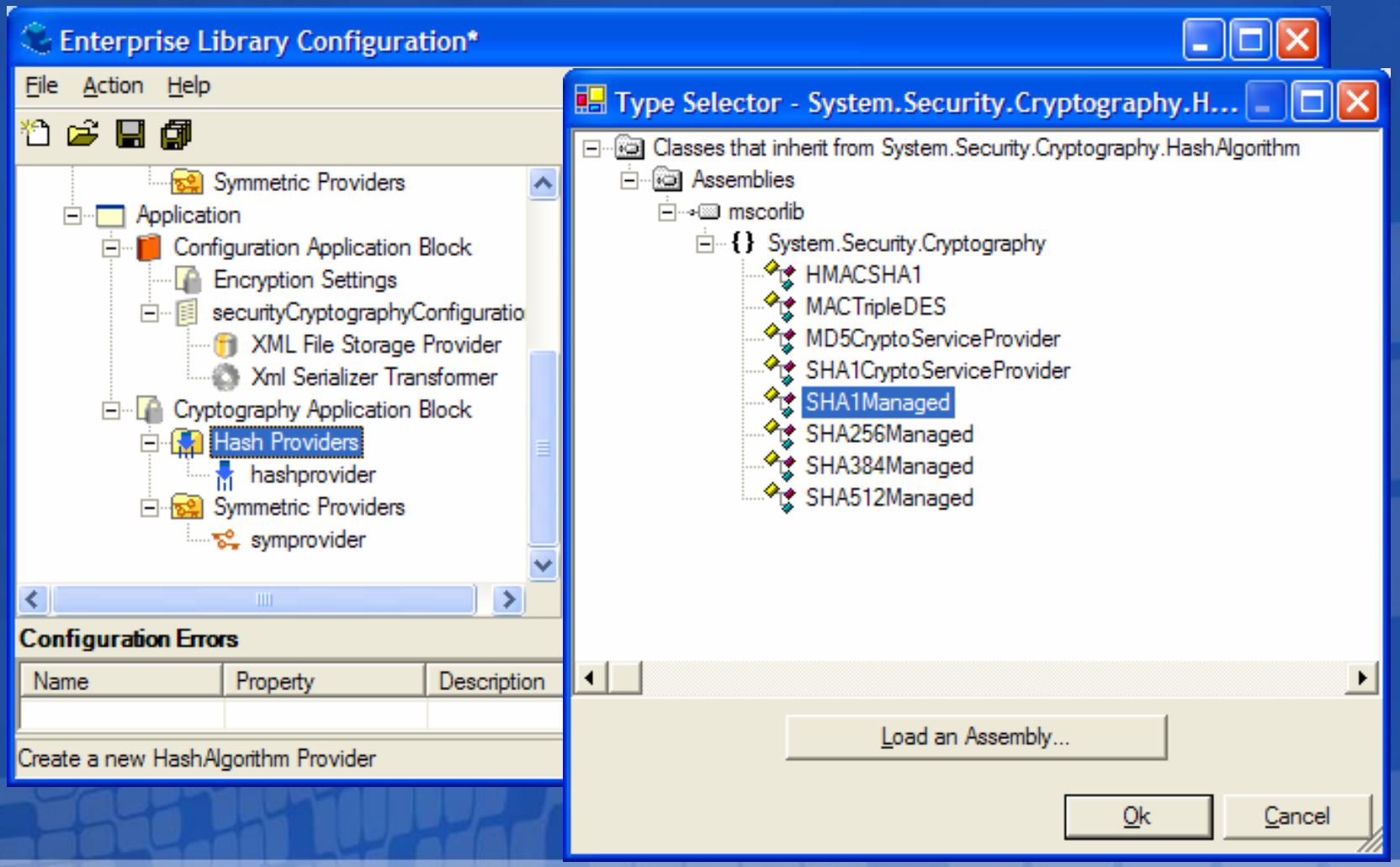
- 选择加密算法
  - 有些性能高，有些加密强
  - 一般长的密钥的安全性高
- 常犯的一个错误
  - 开发自己的加密算法

关于密码的存储问题

- 不要在数据库中存储明文的密码
- 避免保存加密的密码—考虑密钥的管理因素
- 保存密码的单向哈希值
- 哈希使用 salt

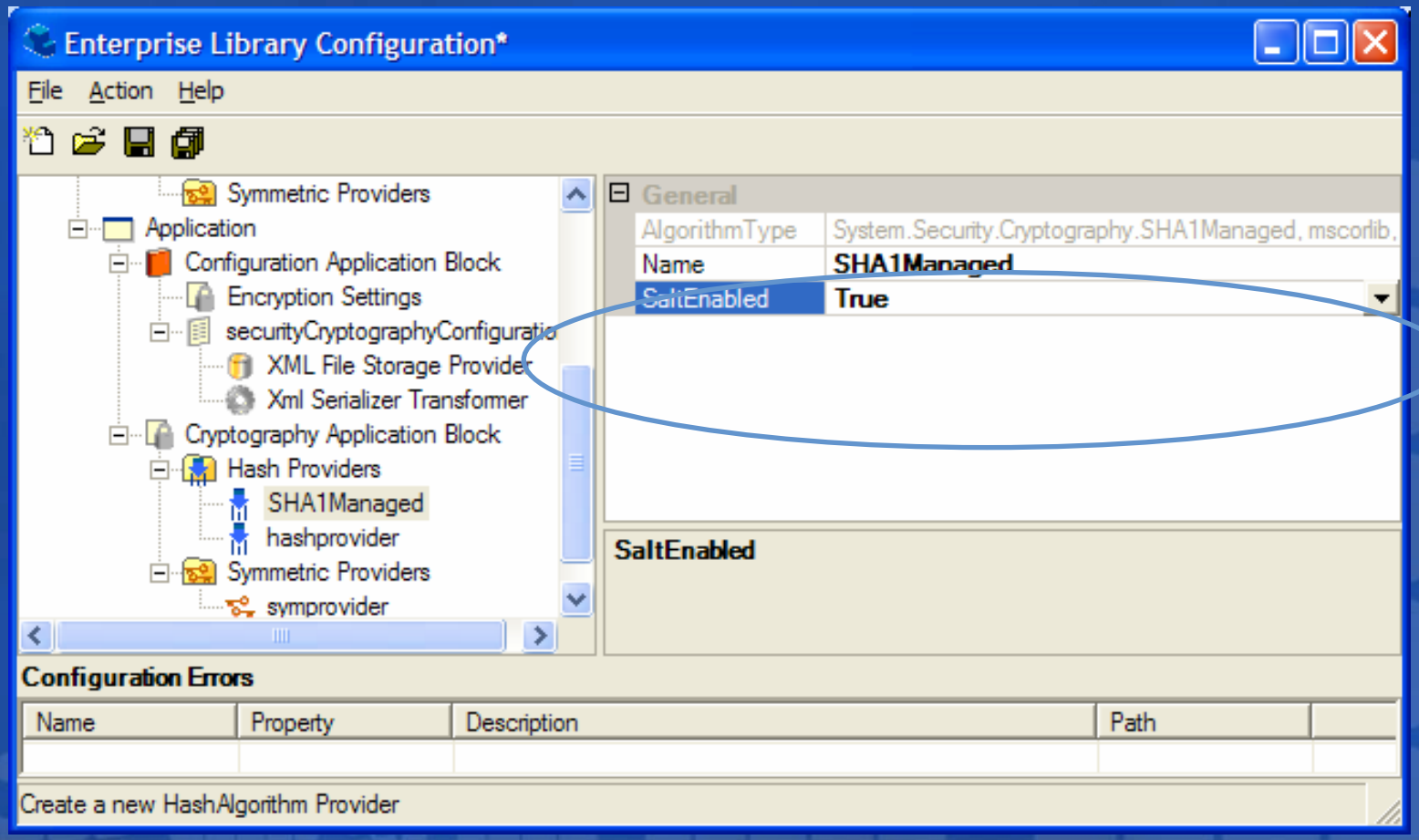
配置 Hash Provider

- 使用 Configuration Console



配置 Hash Provider 使用 Salt

- 每个 provider 都可以使用 salt
- Salt 值由 application block 生成

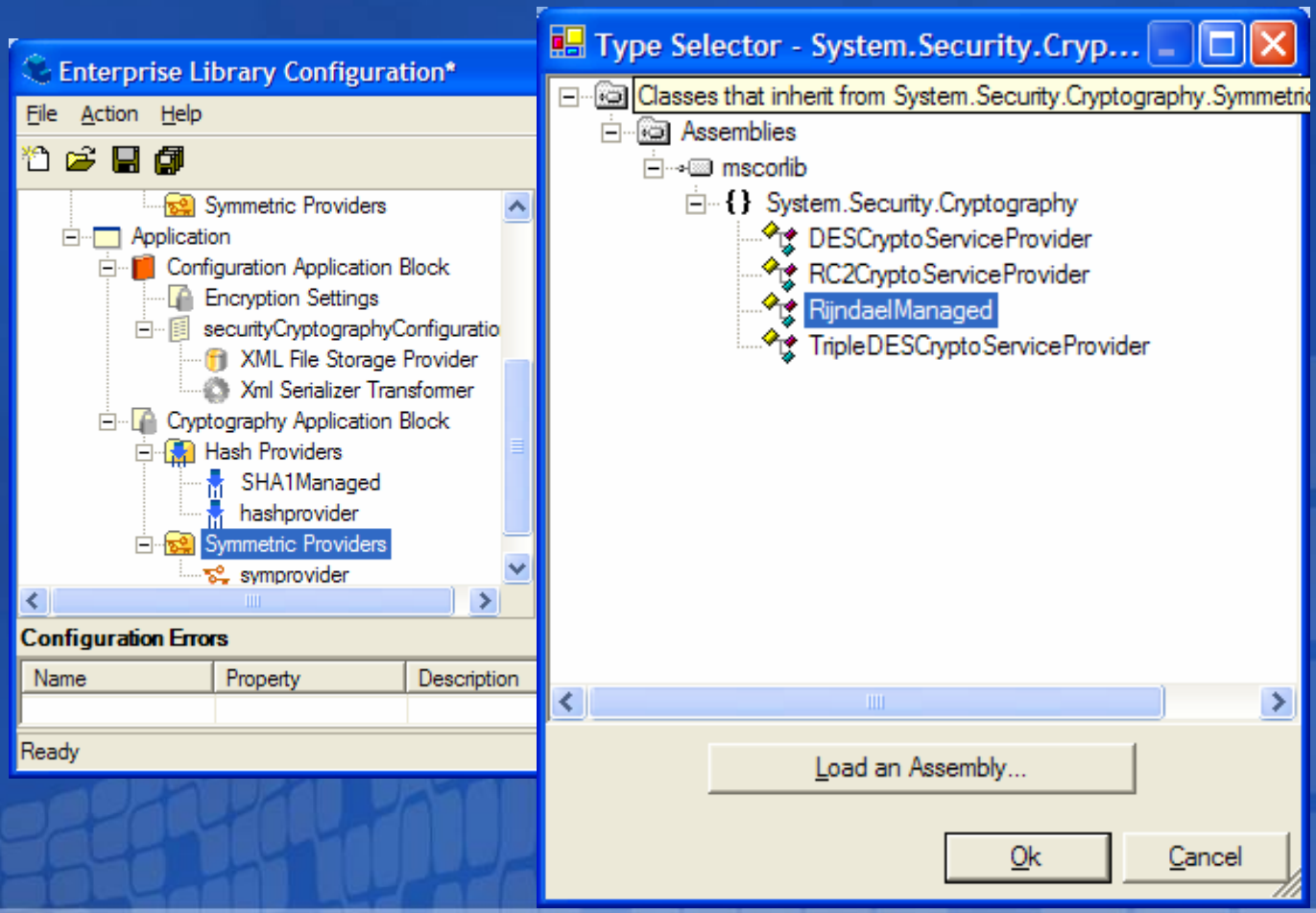


有关 Salt 的更多信息

- salt 长度缺省为 16 字节 (providers can override)
- 采用 RNGCryptoServiceProvider (not Random) 降低了重复 salt 值的可能性
- Salt 与需哈希的值组合，然后产生哈希
- CreateHash() 返回 Salt 和 hash 值
- CompareHash() 提取 salt 并用它来计算哈希值
- 别担心: application block 帮你做了所有这些!

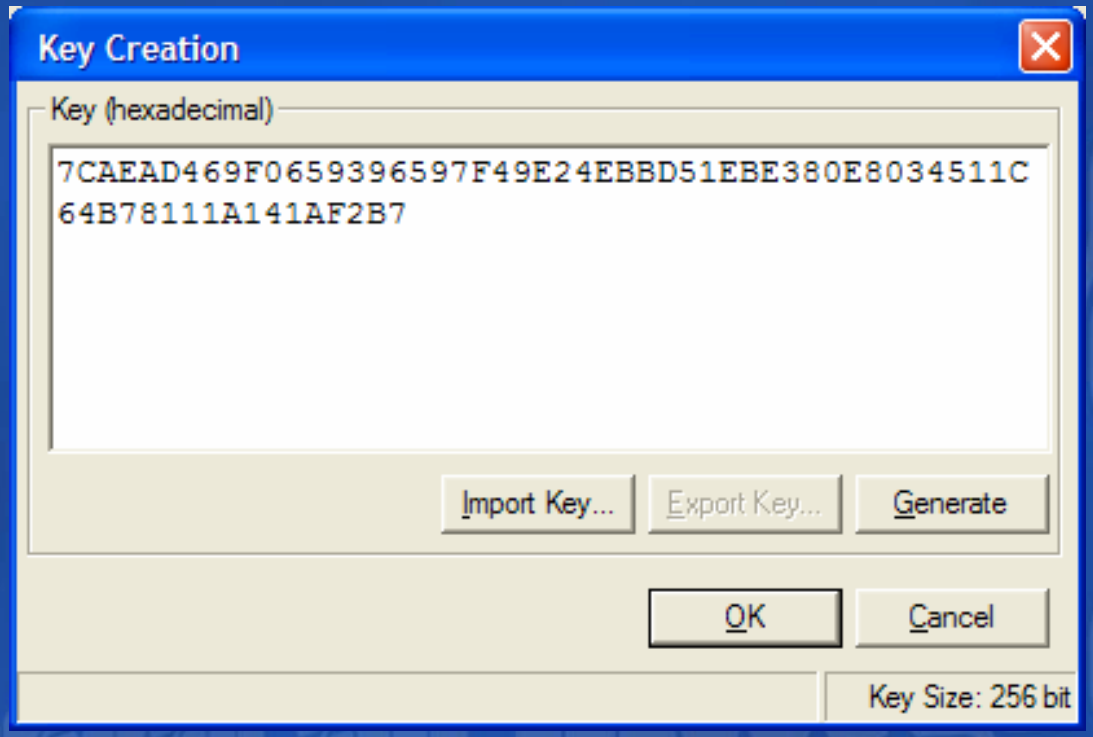
配置一个 Symmetric Encryption Provider

- 使用 Configuration Console



创建对称密钥

- 自动生成密钥
- 导入、导出密钥



密钥的保存

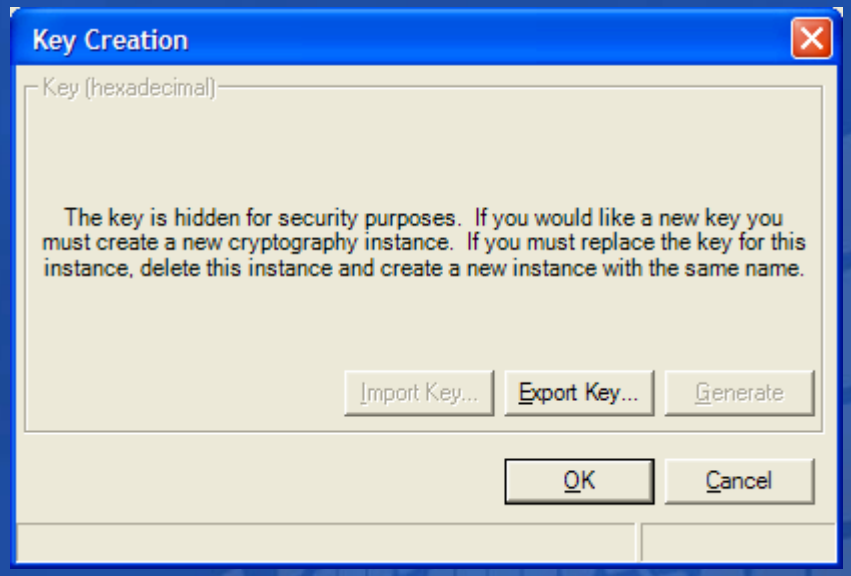
- 没有安全的保存密钥是最常见的错误之一
- 采用以下几种方法:
  - 使用 DPAPI 避免管理密钥
  - 不要在代码中保存密钥
  - 限制密钥的访问

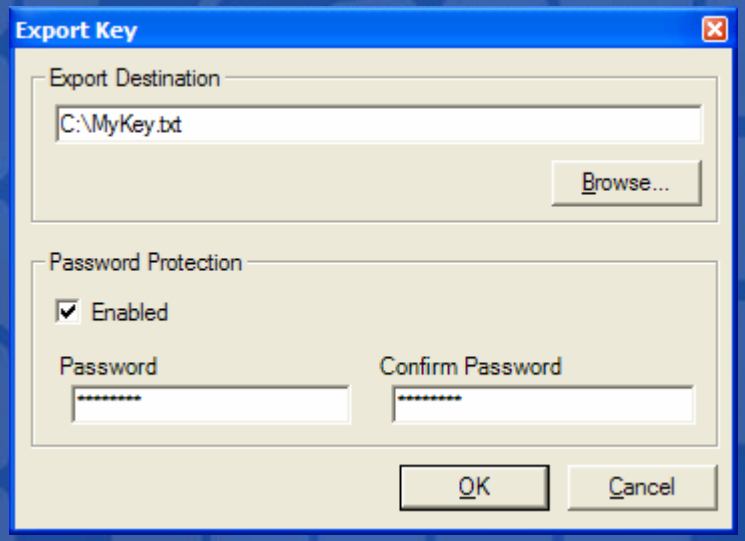
对称密钥的管理

- 密钥以 Base 64 encoded string 的形式保存在 securityCryptographyConfiguration.config 配置文件中
- 如何保护配置文件
  - 文件系统 ACL
  - 对文件系统加密 (EFS)
  - 使用 Configuration Console 对配置文件进行加密 (DPAPI)

导出对称密钥

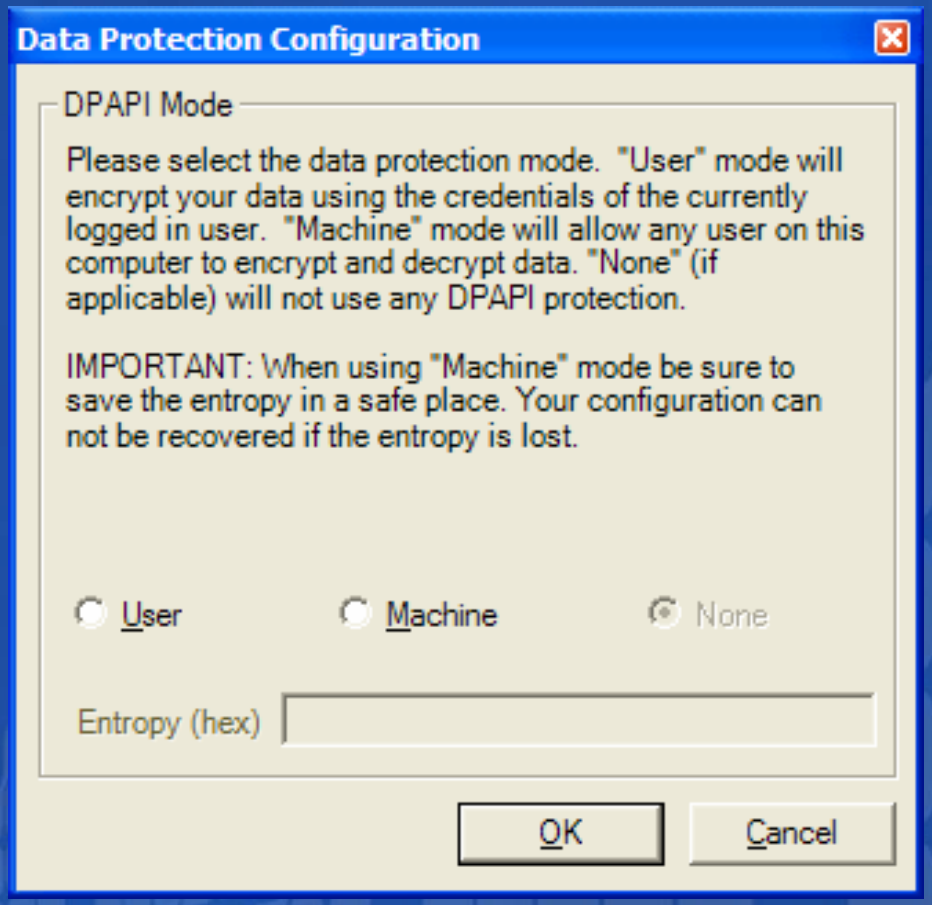
- 将密钥保存到文本文件中
- 可以用密码对导出的密钥加密
- 好好保护密钥!





使用 DPAPI Provider

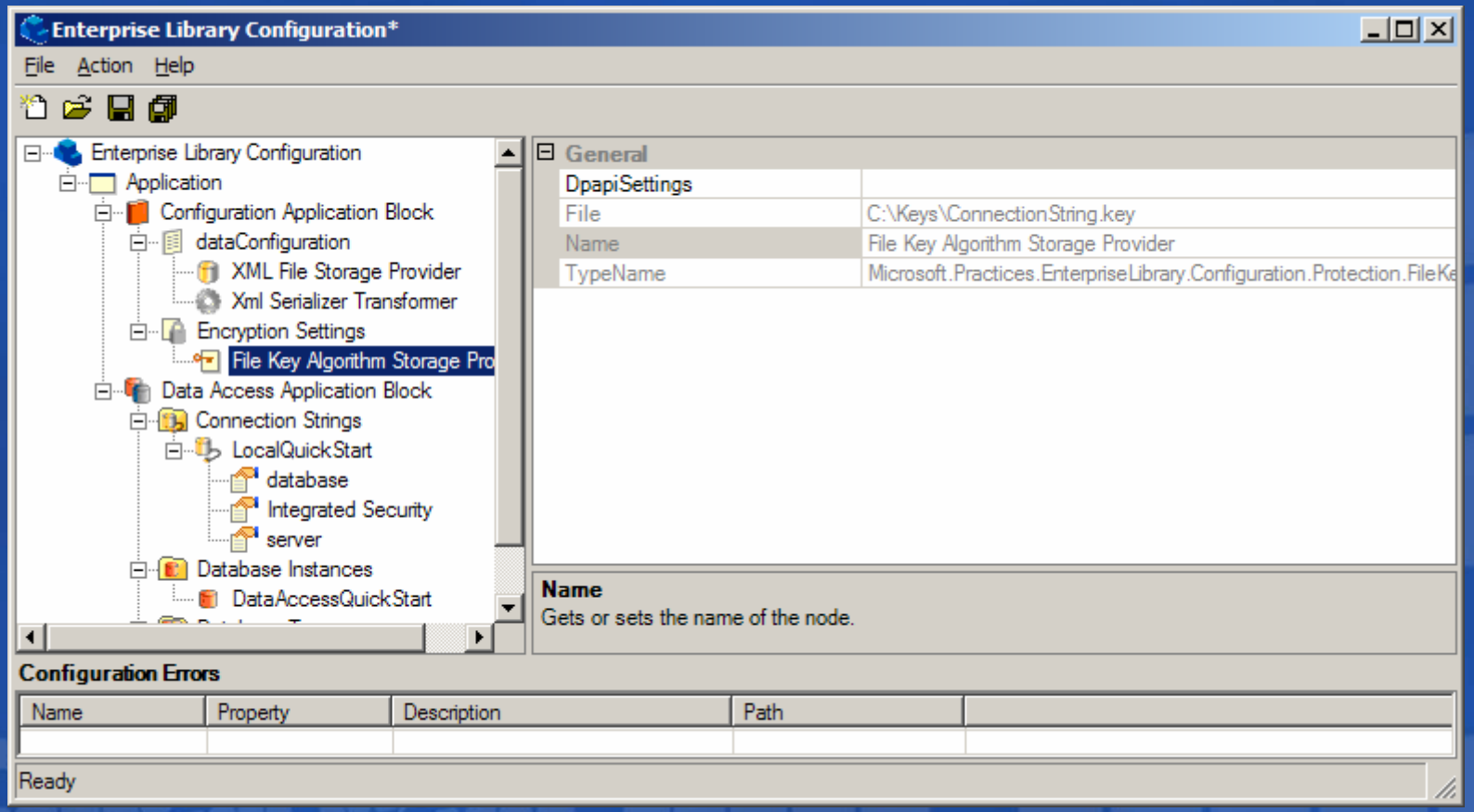
- 避免了密钥管理 (操作系统来管理)
- 有用户和机器两种模式



数据库连接字符串的安全

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

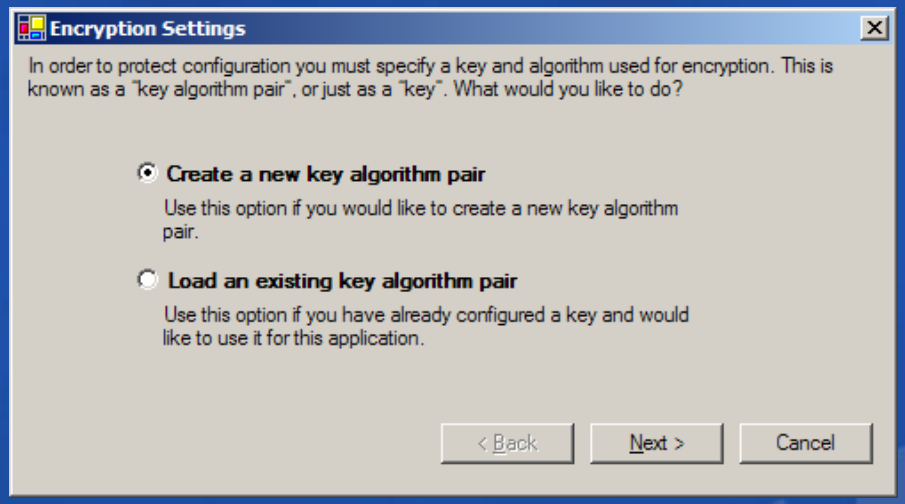
- 加密配置项决定应用程序块的配置如何被加密

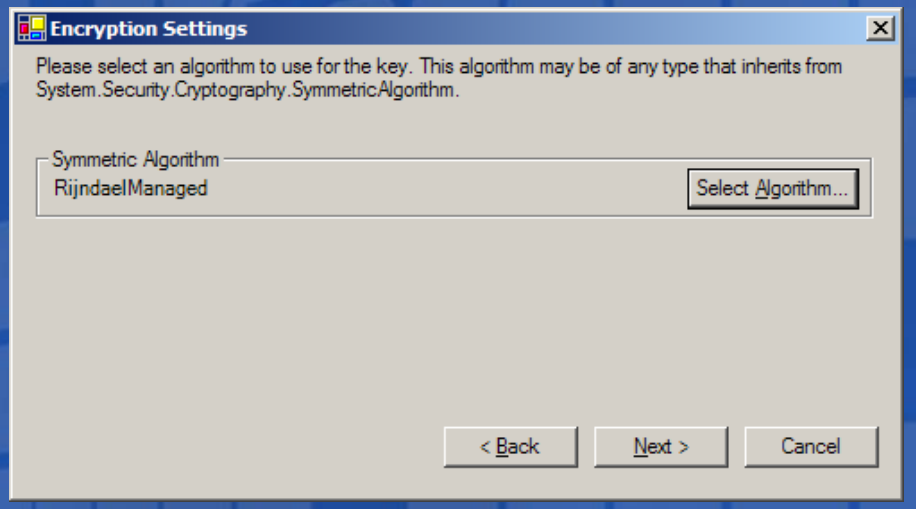


Step 1: 设置 Encryption Settings

您的潜力，我们的动力

Microsoft
微软（中国）有限公司

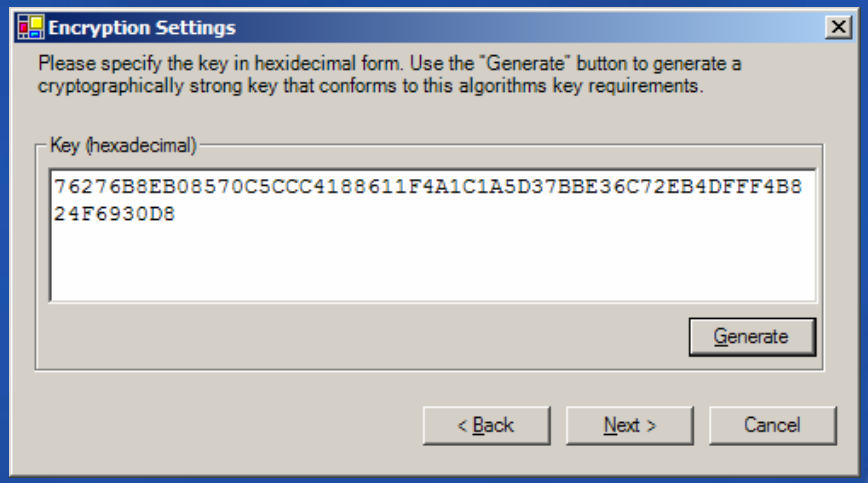


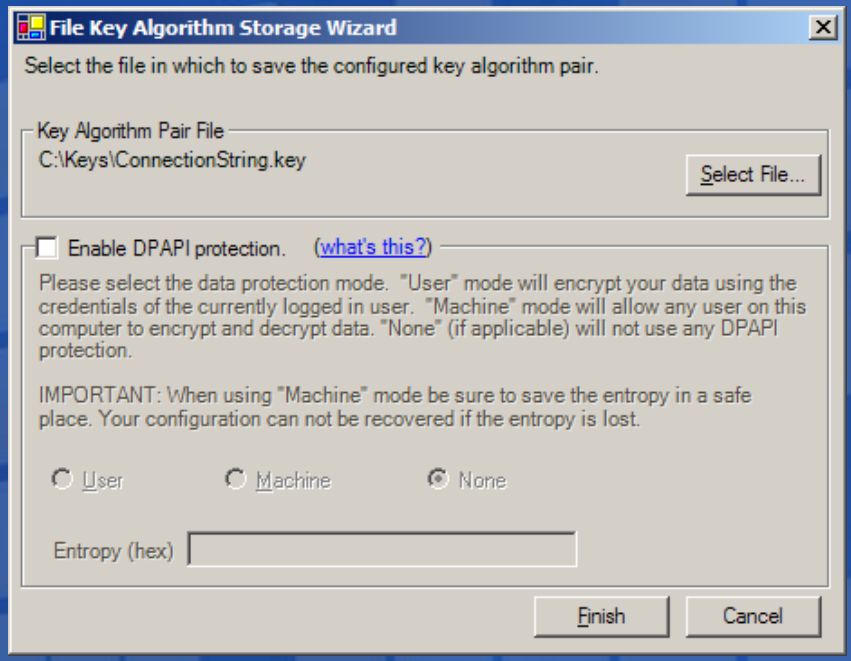


msdn

MSDN Webcasts

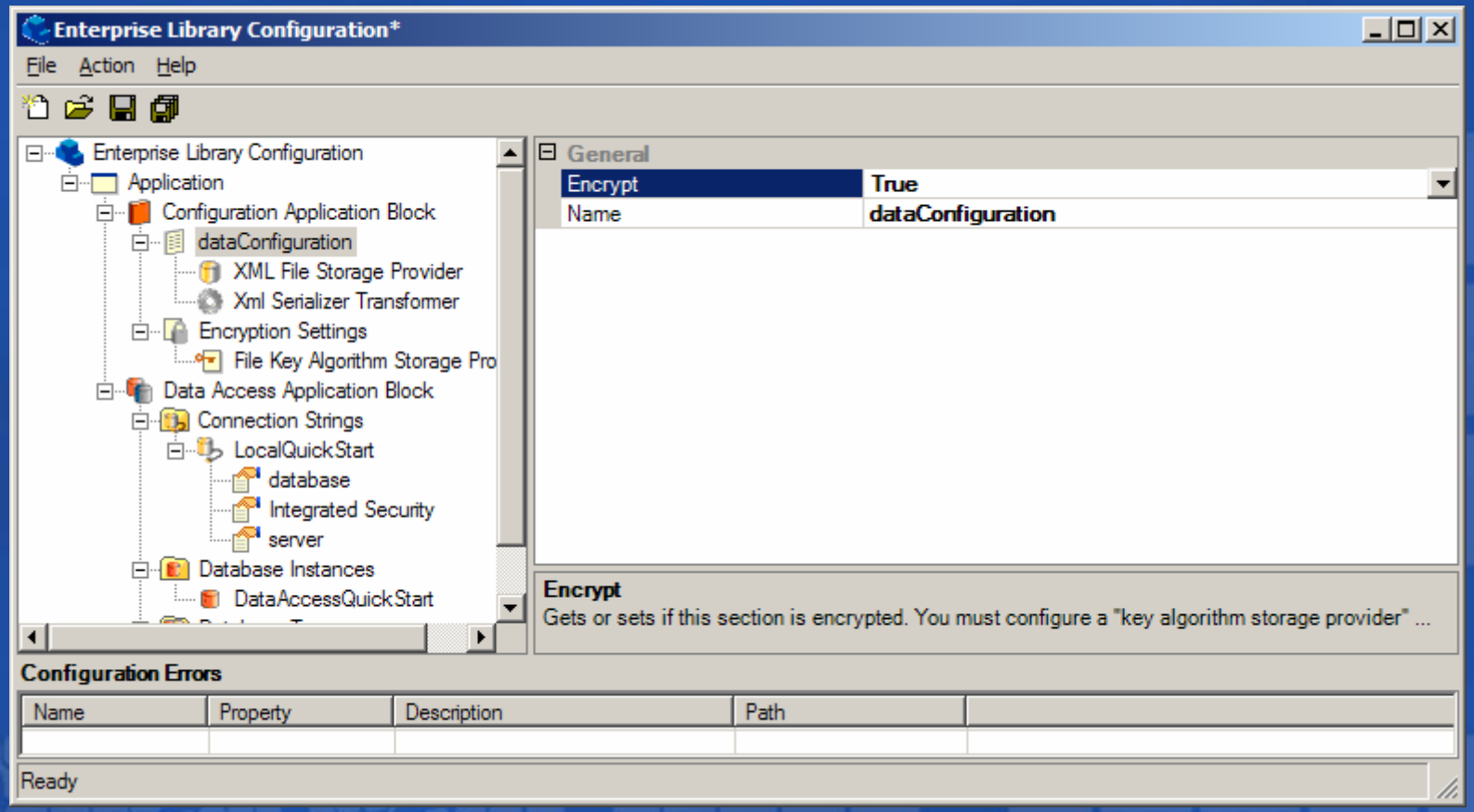
Step 1: 设置 Encryption Settings





Step 2: 将数据库配置区设为加密

- 是否加密则由每个应用程序块的配置来决定



更多资源

- Improving Web Application Security

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

- Building Secure ASP.NET Applications

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

- Enterprise Library 社区

  http://workspaces.gotdotnet.com/entlib


您的潜力，我们的动力
Microsoft
微软（中国）有限公司
Microsoft
msdn
MSDN Webcasts

Q & A

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。