|  |
| --- |
| Delphi Spring Framework Team |
| Delphi Spring Framework |
| Style Guide (Draft) |

|  |
| --- |
| Zuo Baoquan (Paul)  12/3/2009 |

Contents

1. **Preample**

为保证Delphi Spring Framework项目代码的**可读性**和**质量**，特建立本编程规范。本规范适用于Delphi Spring Framework所有项目成员。

1. **Project Management**
2. **Files Organization**
3. **Version Management**

The following types of files should never be committed to the project svn server.

* \*.bak
* \*.dcu
* \*.exe
* \*.~\*
* \*.ddp
* \*.dsk
* \*.local
* \*.identcache
* \*.tvsconfig
* \_\_history

1. **Language Style Guide**
2. **License**

This open source project is licensed in [**Apache License Version 2.0**](http://www.apache.org/licenses/LICENSE-2.0.html). The following copyright declaration must be included in all project files and applications.

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

{ }

{ Delphi Spring Framework }

{ }

{ Copyright (C) 2009-2010 Delphi Spring Framework }

{ }

{ http://delphi-spring-framework.googlecode.com }

{ }

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

{ }

{ Licensed under the Apache License, Version 2.0 (the "License"); }

{ you may not use this file except in compliance with the License. }

{ You may obtain a copy of the License at }

{ }

{ http://www.apache.org/licenses/LICENSE-2.0 }

{ }

{ Unless required by applicable law or agreed to in writing, software }

{ distributed under the License is distributed on an "AS IS" BASIS, }

{ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. }

{ See the License for the specific language governing permissions and }

{ limitations under the License. }

{ }

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

1. **Naming Conventions**

* 单元文件（即namespace），均采用Spring.\*.pas的命名方式，如：Spring.System.pas；
* 类、记录、枚举、集合以及自定义类型以**T**开始，如TSimpleClass、TDriveType；
* 接口类型一律以**I**作为前缀，如IEnumerable，IList<T>；
* 异常类以**E**开头，如EArgumentException，EFileNotFoundException；
* 资源字符串（resourcestring）以**S**作为前缀，如SFileNotFound;
* **类字段成员、局部变量和参数应采用Camel风格，如fOperatingSystem, i, value, targetStream；**
* 类的属性和方法一律采用Pascal风格，**Boolean型属性应使用Is前缀**，如Name，IsReadOnly，IsValid，IsEmpty，GetNextID；
* **抽象基类尽可能使用Base作为后缀，如TCollectionBase，TStreamBase；**
* **自定义属性类名无需前缀T（抽象属性基类则应保留），且以Attribute作为后缀，如:**DisplayNameAttribute = class(TCustomAttribute)

private

fName: string;

public

constructor Create(const name: string);

end;

应用属性时一般省略后缀Attribute，如：

[DisplayName(‘Paul’)]

TSomeClass = class

end;

* 所有保留字全部小写，如procedure，string，begin，end等。

1. **White Space Usage**

使用标准Pascal的缩进和换行方式，如：

for i := 0 to list.Count - 1 do

begin

if condition then

begin

DoSomething;

end

else

begin

DoSomethingElse;

end;

case driveType of

dtNetwork:

begin

//...

end;

else

begin

//...

end;

end;

end;

1. **Comments**

本项目一律使用Xml的风格注释。

/// <summary>

/// Provides version information for a physical file on disk.

/// </summary>

/// <remarks>

/// Remarks

/// </remarks>

TFileVersionInfo = record

//...

end;

**注意：为便于各地开发者阅读和使用代码，请使用英文。**

1. **Classes**
2. **Interfaces**

///

///

Property Getters & Setters

Methods

Properties

注意：泛型接口不应分配GUID。

1. **Statements**
2. **Unicode**
3. **Thread Safety**
4. **Additions**
5. **Design Guidelines**

See [Design Guidelines for Developing Class Libraries](http://msdn.microsoft.com/en-us/library/ms229042.aspx)

1. **References**

* [Design Guidelines for Developing Class Libraries](http://msdn.microsoft.com/en-us/library/ms229042.aspx) （MSDN）
* JEDI Style Guide

1. **泛型**

区分使用泛型类和泛型方法

1. **匿名方法**
2. **必须检查全局函数和所有公开的方法的参数合法性**

Spring.System.TArgument类提供了很多静态方法用来检查参数，如CheckTrue，CheckFalse，CheckRange等。（可使用代码模板减少输入）

1. **全局变量、全局函数以及公开的静态类方法尽可能是线程安全的**

如不支持线程安全必须注明。

1. **编写的代码必须支持Unicode**

**注意：**RTL中有些函数并不支持Unicode，如：

|  |  |
| --- | --- |
| SysUtils.UpperCase | 仅支持标准7位Ascii码中’a’-‘z’的大小写转换,应使用**Character.TCharacter**类的ToUpper和ToLower方法 |
| SysUtils.LowerCase |
| SysUtils.BytesOf | 应使用WideBytesOf |
| SysUtils.StringOf | 应使用WideStringOf |

1. **其他建议**

* 尽可能将参数定义成**const**、**var**或**out**
* 应将抽象基类标记为**abstract**，尽可能将真正的构造器放到protected里面，并在默认的构造器内抛出EAbstractClassException异常
* 应优先考虑使用record来设计工具类（Utility Class）
* 优先考虑使用record来设计值对象（生命周期自动管理和支持运算符重载）
* 尽可能保证每个过程职责清晰，一般不超过20行代码
* 尽可能保证代码是**可测试的**，并使用**DUnit**编写单元测试用例
* 尽可能保持版本兼容性，使用**deprecated**, **experimental**等关键字提醒开发者
* 谨慎使用**class constructor**和**class destructor**代替单元的初始化（initialization）和终止化（finalization）
* 抛异常时优先使用Exception.CreateRes（@SResourceString）等方法