

Delphi Spring Framework 编程规范 (草稿) V0.1

1. 前言

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。本规范仅适用于 Delphi Spring Framework 各项目成员，大家如有任何意见和建议，请联系：

姓名：左保权 (Paul)

MSN/Email: baoquan.zuo@gmail.com

2. 所有项目文件、单元文件和示例程序必须包含下列版权声明：

```
{*****}
{
    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.
}
{*****}
```

3. 命名规范

- 单元文件 (即 namespace)，均以 Spring 开头，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** 开头，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` , `GetNextID` ;
- 所有保留字全部小写，如 `procedure` , `string` , `begin` , `end` 等。

4. 缩进及换行

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
        //...
    end;
    case driveType of
        dtNetwork:
        begin
            //...
        end;
        else
        begin
            //...
        end;
    end;
end;
```

Comment [Paul1]: 此处是否需要缩进？

5. 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。

6. 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）

7. 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。

8. 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
SysUtils.LowerCase	Character.TCharacter 类的 ToUpper 和 ToLower 方法
SysUtils.BytesOf	应使用 WideBytesOf
SysUtils.StringOf	应使用 WideStringOf

9. 除非特别需求，否则永远不要把下列类型的文件上传到 svn 服务器：

- *.bak
- *.dcu
- *.exe
- *.~*
- *.ddp
- *.dsk
- *.local
- *.identcache
- *.tvsconfig
- __history (文件夹)

10. 其他建议

- 尽可能将参数定义成 **const**、**var** 或 **out**
- 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
- 尽可能使用 **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