

# Log4Delphi Coding Standards

## Table of contents

1 Coding Standards Used By Log4Delphi Developers.....	2
1.1 Introduction.....	2
1.2 General Source Conventions.....	2
1.3 Object Pascal Conventions.....	3
1.4 File Conventions.....	4
1.5 .....	5
2 Other Guidelines.....	5

## 1. Coding Standards Used By Log4Delphi Developers

### 1.1. Introduction

This document describes a list of coding conventions that are required for code submissions to the project. By default, the coding conventions for most Open Source Projects should follow the existing coding conventions in the code that you are working on. For example, if the begin is on the same line as the if statement, then you should write all your code to have that convention.

**If you commit code that does not follow these conventions and you are caught, you are responsible for fixing that code.**

### 1.2. General Source Conventions

#### 1.2.1. Indentation

Indenting is three spaces per level. **NO TAB** characters!

The reserved words `unit`, `uses`, `type`, `interface`, `implementation`, `initialization` and `finalization` are always flush with the left margin.

#### 1.2.2. Margins

Margins are set at 80 characters. In general, no source code should exceed past this margin. Statements that are longer than 80 characters should be wrapped at an operator, ie: parenthesis, commas, etc. If a statement is wrapped it should be indented to match the block it is in.

#### 1.2.3. Comments

Comments should be made using `{` and `}`. Single line comments embedded within the code are fine if they are done with `//`. The `{ * * }` comments should be avoided. Code should be commented using the [DelphiCodeToDoc](#) / Javadoc format so that the API can be produced from it.

#### 1.2.4. Conditional Defines

Conditional defines are with `{ }` braces and in uppercase. Each define is named in its closing block to aid readability and they are indented as blocks.

```
if (true) then
begin
    {$IFDEF SOMEVAL}
        // statements here
    {$ELSE}
        // statements here
    {$ENDIF SOMEVAL}
end;
```

### 1.2.5. Begin and End

Begin and end occur on separate lines with an exception being made for an else clause.

```
if (x = 2) then
begin
    y := 0;
end else begin
    y := x;
end;
```

## 1.3. Object Pascal Conventions

### 1.3.1. Parenthesis

White space is permitted but not required or enforced. Generally there is no space between an opening parenthesis and the next character as well as between a character and a closing parenthesis.

```
if (x = 2) then // PREFERRED way of doing it
    // ...

if ( x = 2 ) then // ACCEPTABLE way of doing it
    // ...
```

### 1.3.2. Reserved Words

Reserved words and key words are always in lower case.

### 1.3.3. Case Statements

Case statements follow a strict indenting pattern:

```
case x of
    0 : begin
```

```

    // statements
end;
// ...
n : begin
    // statements
end;
else {CASE}
    // ...
end;

```

### 1.3.4. Classes

All classes are named with a capital T followed by the class's name in sentence case. All classes should descend (subclass) from TObject, so that they may be added as objects to TString instances. The keywords `public`, `private`, `protected`, `published` are indented with the class and occur in the order: `private`, `protected`, `public` and `published`.

```

TFileAppender = class (TObject)
private
    // statements
protected
    // statements
public
    // statements
end;

```

## 1.4. File Conventions

### 1.4.1. Units

Each unit is declared in its own file with the file name the name of the unit with a `Unit.pas` suffix.

Only one class may be defined per unit, per file. The Unit name will be the class name with the `Unit` suffix.

### 1.4.2. File Header

Each source file must contain the header:

```

{
    Copyright 2005-2006 Log4Delphi Project

    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.
}
```

## 1.5.

## 2. Other Guidelines

1. Destructors of a class should only free the objects within that class and call the super class's destructor to free those objects. Abstract classes that define objects on the heap should include destructors.
2. In general, the caller is responsible for freeing memory. Objects passed as parameters to a procedure or function are not freed by that function but should be freed by that function's caller.

Copyright 2005-2006 Log4Delphi Project. All Rights Reserved.