Getting Started With Log4Delphi

Table of contents

1 About This Guide....2
2 Configuring the Log4Delphi Package....2
3 Configuring the Logger....2
4 Logging Messages....3
5 Freeing the Logger Instance....3

1. About This Guide

The Getting Started Guide is designed to explain the steps necessary to quickly setup and use Log4Delphi in an application. As such, it does not go into much detail about the various options and uses.

2. Configuring the Log4Delphi Package

The first thing that needs to be done it to configure the Log4Delphi package. This is a simple matter of adding one line of code to your Application's code.

You use the Configurator Unit to do a basic configuration, which will then initialize and configure the package. The steps are:

1. Launch Delphi and open your application.
2. Select `Project > View Source` from the main menu.
3. Add `TConfiguratorUnit.doBasicConfiguration;` just after the `Application.Initialize;` line. You should have code that is similar to:

   ```
   Application.Initialize;
   TConfiguratorUnit.doBasicConfiguration;
   Application.CreateForm(TForm1, Form1);
   Application.Run;
   ```

4. Make sure that `TConfiguratorUnit` is added to the uses clause.

3. Configuring the Logger

You can configure the logger to perform logging to a file and to log specific event messages. The steps are:

1. Retrieve the Logger instance using the `getInstance` class method: `logger := TLogger.getInstance;`.
2. You can then set the level of the logger using: `logger.setLevel(TLevelUnit.INFO);` .This is not required since the default level is DEBUG.
3. You would typically add an appender to the logger to log events with, we'll use a file appender: `logger.addAppender(TFileAppender.Create('C:\test.log'));`
4. That's all there is to configuring the logger. Just remember to include the appropriate units to your uses clause. As in the example below.

```
...
uses TLoggerUnit, TLevelUnit, TFileAppenderUnit;
```

```
...
var
   logger : TLogger;
begin
   logger := TLogger.getInstance;
   logger.setLevel(TLevelUnit.INFO);
   logger.addAppender(TFileAppender.Create('C:\test.log'));
   ...
```

4. Logging Messages

Logging messages is simple once you have set up the logger. There are five types of messages that can be logged, FATAL, ERROR, WARN, INFO and DEBUG. This is done as follows:

```
   logger.debug('Debug message');
   logger.info('Info message');
   logger.warn('Warn message');
   logger.error('Error message');
   logger.fatal('Fatal message');
```

5. Freeing the Logger Instance

The last thing you'll need to do is free the logger instance when the application terminates. This will take care of freeing all resources used by the logging package. It is a good idea to put this in an application event handler or something similar.

```
...
TLogger.freeInstances;
...
```