C++ Coding Guidelines for COPASI

Stefan Hoops
Virginia Bioinformatics Institute
1880 Pratt Dr.
Blacksburg, VA 24060
USA
shoops@vbi.vt.edu

March 1, 2005

Contents

Chapter 1

Naming Conventions

The intent of naming conventions is to allow programmers, which are not familiar with the code to easily grasp the meaning and scope of symbols in the source code. Each programmer of COPASI should adhere for its own benefit and to the advantage of the project to the following conventions.

1.1 Class Names

Class names must all start with a capital letter `C`. This is followed by a descriptive name. This name might be composed by different words. These words must all start with capital letters and are concatenated without underscores. Good examples for class names include: `CCopasiXMLParser` , `CExpatTemplate`, and `CMathModel`.

1.2 Variable Name

In general a variable name should be descriptive. This name might be composed by different words. These words must all start with capital letters and are concatenated without underscores. In addition the following standards should be followed:

- **Counters** might be used such as `i`, `k`, and `l`, which may be used in loops.
- **Iterators** might be used such as `it` and `end`, which may be used in loops.

- **Method Parameters** must start with a lower case letter.
- **Class Member Variables** are prepended with a lower case letter `m`.
- **Pointers** are prepended with a lower case letter `p`.
- **Class Member Variables** which are **Pointers** must have the prefix `mp`.

In addition the following standards should be followed:

1.3 Method Name

Method names should have a descriptive name starting with a lower case letter. This name might be composed by different words. These words beginning with the second must start with capital letters and are concatenated without underscores. Good examples for method names include: `createMetabolite`, `compileIfNecessary`, and `buildMoieties`

In addition the following standards should be followed:

- **Retrieval Methods** must start with `get` followed by the member variable name without the prefix.
- **Set Methods** must start with `set` followed by the member variable name without the prefix.
- **Boolean Query Functions** should start if applicable with `is`.

Chapter 2

Program Code Guidelines

2.1 Loops

Chapter 3

Code Documentation

3.1 Class Documentation

3.2 Variable Documentation

3.3 Method Documentation

3.4 Incline Code Documentation

Chapter 4

Installation Structure

This section defines the installation structure for COPASI on different platforms. Each platform will adhere to the platform specific requirements.

4.1 Unix

The installation location needs to be available to COPASI at runtime and therefore the environment variable `COPASIDIR` pointing to this location must be set by the user.

```
$COPASIDIR
 +- bin
 |   +- CopasiSE
 |   +- CopasiUI
 +- share
 |   +- copasi
 |       +- doc
 |       |   +- html
 |       |       +- figures
 |       |       |   +- DefaultPlotAdded.jpg
 |       |       |   +- ModelSettingsDialog.jpg
 |       |       |   +- ObjectBrowserSelection.jpg
 |       |       |   +- ObjectBrowserTree.jpg
 |       |       |   +- PlotDefinition.jpg
 |       |       |   +- PlotWindow.jpg
 |       |       |   +- ReactionDialog.jpg
 |       |       |   +- ReactionOverview.jpg
 |       |       |   +- ReactionOverviewEmpty.jpg
 |       |       |   +- ReportDefinitionDialog.jpg
 |       |       |   +- TimeCourseDialog.jpg
 |       |       |   +- ...
 |       |       +- TutWiz-Step1.html
 |       |       +- TutWiz-Step2.html
 |       |       +- TutWiz-Step3.html
 |       |       +- TutWiz-Step4.html
 |       |       +- TutWiz-Step5.html
```

```
|       |       +- TutWiz-Step6.html
|       |       +- ...
|       +- examples
|           +- CircadianClock.cps
|           +- Metabolism-2000Poo.xml
|           +- YeastGlycolysis.gps
|           +- brusselator.cps
|           +- ...
+- README
+- ChangeLog
```

4.2 MacOS X

The installation location must be available to COPASI at runtime. However it is possible to determine the location through MacOS X.

```
$COPASIDIR
 +- CopasiSE.app
 |   +- Contents
 |   |   +- MacOS
 |   |       +- CopasiSE
 +- CopasiUI.app
 |   +- Contents
 |   |   +- MacOS
 |   |   |   +- CopasiUI
 |   |   +- Resources
 |   |   |   +- doc
 |   |   |       +- html
 |   |   |           +- figures
 |   |   |           |   +- DefaultPlotAdded.jpg
 |   |   |           |   +- ModelSettingsDialog.jpg
 |   |   |           |   +- ObjectBrowserSelection.jpg
 |   |   |           |   +- ObjectBrowserTree.jpg
 |   |   |           |   +- PlotDefinition.jpg
 |   |   |           |   +- PlotWindow.jpg
 |   |   |           |   +- ReactionDialog.jpg
 |   |   |           |   +- ReactionOverview.jpg
 |   |   |           |   +- ReactionOverviewEmpty.jpg
 |   |   |           |   +- ReportDefinitionDialog.jpg
 |   |   |           |   +- TimeCourseDialog.jpg
 |   |   |           |   +- ...
 |   |   |           +- TutWiz-Step1.html
 |   |   |           +- TutWiz-Step2.html
 |   |   |           +- TutWiz-Step3.html
 |   |   |           +- TutWiz-Step4.html
 |   |   |           +- TutWiz-Step5.html
 |   |   |           +- TutWiz-Step6.html
 |   |   |           +- ...
 |   |   +- Info.plist
 +- examples
 |   +- CircadianClock.cps
 |   +- Metabolism-2000Poo.xml
 |   +- YeastGlycolysis.gps
 |   +- brusselator.cps
```

```
|   +- ...
+- COPASI-README.rtf
```

4.3 Windows

The installation location must be available to COPASI at runtime. However it is possible to determine the location through Windows specific means.

```
$COPASIDIR
 +- bin
 |   +- CopasiSE
 |   +- CopasiUI
 +- share
 |   +- copasi
 |       +- doc
 |       |   +- html
 |       |       +- figures
 |       |       |   +- DefaultPlotAdded.jpg
 |       |       |   +- ModelSettingsDialog.jpg
 |       |       |   +- ObjectBrowserSelection.jpg
 |       |       |   +- ObjectBrowserTree.jpg
 |       |       |   +- PlotDefinition.jpg
 |       |       |   +- PlotWindow.jpg
 |       |       |   +- ReactionDialog.jpg
 |       |       |   +- ReactionOverview.jpg
 |       |       |   +- ReactionOverviewEmpty.jpg
 |       |       |   +- ReportDefinitionDialog.jpg
 |       |       |   +- TimeCourseDialog.jpg
 |       |       |   +- ...
 |       |       +- TutWiz-Step1.html
 |       |       +- TutWiz-Step2.html
 |       |       +- TutWiz-Step3.html
 |       |       +- TutWiz-Step4.html
 |       |       +- TutWiz-Step5.html
 |       |       +- TutWiz-Step6.html
 |       |       +- ...
 |       +- examples
 |           +- CircadianClock.cps
 |           +- Metabolism-2000Poo.xml
 |           +- YeastGlycolysis.gps
 |           +- brusselator.cps
 |           +- ...
 +- README
 +- ChangeLog
```

4.4 Handling Installation Differences

The handling of differences in the installation structure must be dealt with in one place within the COPASI code. The place for this is the class `COptions`. In this class the method:

```
template< class CType > static void getValue(const std::string & name,
                                             CType & value)
```

provides access to common options within COPASI. The following values will deal with installation dependent settings: `CopasiDir`, `TempDir`, `ExampleDir`, and `WizardDir`. The following code shows howe to retrieve the location of the examples files for COPASI:

```
std::string ExampleDir;
COptions::getValue(``ExampleDir'', ExampleDir);
```

To assure that the values are correctly set any main program must call:

```
COptions::init(argc, argv);
```