



QuantumConcepts

Quick Start Guide

CodeGenerator v1.5.0



Quick Start Guide

CodeGenerator v1.5.0

Contents

Revision History	2
Summary	3
How It Works	4
Database Schema.....	4
Customization	4
APIs.....	4
Annotations.....	4
Attributes	5
Transformation & Output	5
Creating a Project.....	6
General Tab.....	6
Database Tab	6
Data Types Tab.....	6
Attributes Tab	6
Working With Your New Project.....	7
Creating a New Template	7
Single File	7
Multiple Files.....	7
Generating	8
Next Steps	9
Support & Issues	9

Revision History

3/26/2011 Initial version.



Quick Start Guide

CodeGenerator v1.5.0

Summary

Quantum Concepts CodeGenerator is a very powerful tool which will have a great effect on your software projects. Using CodeGenerator you will be able to streamline the development of the most crucial portions of your application. The end result is a better, more stable, more robust product in a much shorter period of time.

Using CodeGenerator, you will achieve the following:

- **Write Once** – You write your code once, in the form of an XSL Template. This is the backbone of the CodeGenerator system.
- **Fix Once** – Since numerous classes will be built from your XSL Templates, when an issue is found you simply fix the template and the fix is applied to all files which are generated from that template.
- **Standardize** – No more hand-writing the same classes over and over. CodeGenerator greatly simplifies coding standard enforcement because instead of writing numerous similar classes, CodeGenerator generates them all from a common XSL Template.
- **Work Faster** – Once you embrace the power of XSL Templates, you will quickly think of other ways you can apply a template to save you an amazing amount of coding time.
- **Stabilize** – Code stability will increase as you tweak your XSL Templates to work the way you want them to. You can quickly build in things like error checking or permission checking – and you do it in a very intuitive way.

Once you get the idea, you'll use CodeGenerator for all of your projects because it will save you time which is good for you and your clients!

This Quick Start Guide will layout the basics of using CodeGenerator. After reading this guide you should be able to get started quickly by using pre-created templates (called Packages), or by starting from scratch and writing your own.



Quick Start Guide

CodeGenerator v1.5.0

How It Works

Think of CodeGenerator as translator which talks to your database and responds with massive amounts of generated code. This process is illustrated below (Figure 1).

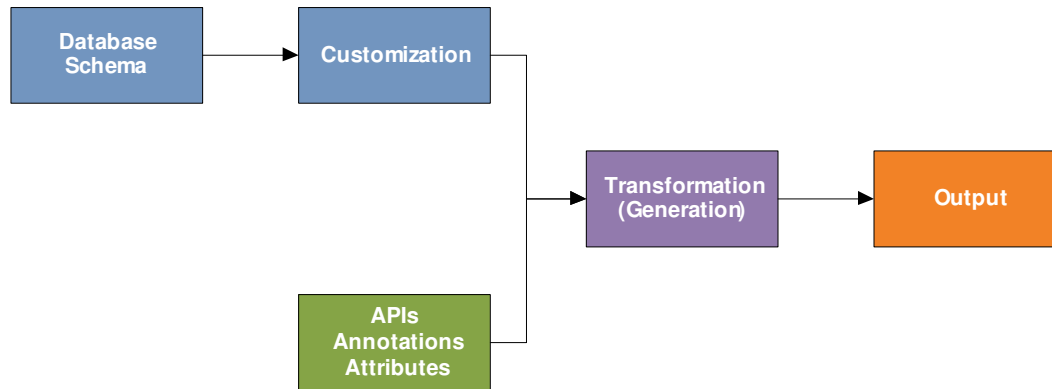


Figure 1: A high-level overview of how CodeGenerator works.

Database Schema

CodeGenerator takes your database schema as input and adds a customization layer on top of it. This allows you to completely customize class names, field names, parent and child relationships, enumerations, etc. You can then add additional meta data to your project by creating APIs, Annotations (code comments), and Attributes. All of this data is sent through your XLS Templates to generate the output – code, SQL scripts, documentation, etc.

Customization

Once your database schema has been imported, you can then begin to customize your output. The **Project Explorer** lists all of the tables, views, columns, foreign keys, and unique indices which exist in your database schema. Selecting one of these items will populate the **Properties** panel. You can use the **Properties** panel to override class names, data types and column names, define APIs and add annotations and attributes.

*Hint: use the **Project > Add to All** menu to quickly add annotations and attributes to all of your classes, properties, foreign keys, unique indices, and APIs.*

APIs

Defining an API on a Table allows you to stub out common functionality such as method signatures, permissions checking, and error handling. This way all of your common functionality will work the same which will increase the reliability of your code.

Annotations

You can custom-write annotations for your classes, column, enumerations, enumeration values, foreign keys, etc. This will allow you to auto-generate documentation for each of these items.



Quick Start Guide

CodeGenerator v1.5.0

Attributes

Attributes can be used to store anything you may need to reference from your XSL Templates. This allows you to treat certain items in specific ways. For instance, adding an attribute with a Key of “Cache” could signify that this object should be kept in memory and all queries should be fed through the cache instead of to the database. A good example of this would be to cache a States or Countries table.

Transformation & Output

This is where the magic happens. Using packaged or custom-written templates, the entire project is pushed through your XSL Templates and the output is generated. Templates are extremely powerful and allow you to do just about anything. You can even write .NET code inside of your XSL Templates to perform tasks which are not supported using standard XSL. Each template performs a specific duty. Here are some templates that you are likely to use or create:

- **DataObjects.xslt** – This template generates a base Data Object for each of your non-excluded database tables.
- **DataAccess.xslt** – This template handles the interface between the Data Objects and your database.
- **Cache.xslt** – This template is responsible for generating cache classes for each of the tables which you marked with the “Cache” attribute.
- **Logic.xslt** – This template handles all of your API logic. These classes interact with the Data Access layer and perform specific functions as defined on the APIs tab of your tables.
- ...and that’s just the beginning!

Other types of templates include web services (WCF, SOAP, REST, etc.), code documentation, database schema documentation, use cases, test cases, etc. The options are limitless.



Quick Start Guide

CodeGenerator v1.5.0

Creating a Project

Open CodeGenerator and click the **New** button in the toolbar. You can also choose **File > New** from the menu. You will be presented with the **Project Properties** window. Project setup is fairly straightforward.

General Tab

On the **General** tab, currently all you have to enter is the **Root Namespace** of your project. For instance, “QuantumConcepts.CodeGenerator.Sample” is used for the sample project.

Database Tab

On the **Database** tab, enter the connection details for the server and database of your project. CodeGenerator will automatically read your database schema and extract information about your tables, views, columns, foreign keys, unique indices, etc.

Data Types Tab

The **Data Types** tab allows you to map your database data types to your application data types. A set of pre-generated mappings exists for some combinations. Click the down arrow on the **Reset to Default** button and choose the appropriate language and the mappings will be automatically populated for you.

Attributes Tab

You can add attributes at almost every level within CodeGenerator. In this case, this tab allows you to add project-level attributes which you can then reference in your XSL Templates. This will be discussed in detail later on.



Quick Start Guide

CodeGenerator v1.5.0

Working With Your New Project

Once you've finished configuring your project for the first time, CodeGenerator will read your database schema and populate the **Project Explorer**. At this point you should look through the items here and customize them to fit your needs. Once you're satisfied, you need to link up your templates.

Creating a New Template

To create a new templates simply right-click on the **Templates** node and choose **New Template**. First you need to point to the XSL Templates you would like to use. Once that is set, you will have to choose an output mode.

Single File

This option indicates that the XSL Template you have selected will be generated to one file. This is the most common case but it depends on a few factors. If you're using a language that requires each class to be in its own file – such as Java – then you cannot use this option. This option is recommended if your language or framework is compatible with multiple-class files. Simply enter or choose the output file where you would like this template to store the resulting output.

Multiple Files

CodeGenerator offers a very flexible option of allowing you to generate more than one file from a single XSL Template. This is useful if you are using Java or if you are generating web pages as in the sample project. If you wish to use this output method, click the **Add** menu button to create a new Template Output Definition.

From the **Element Type** drop down box, choose which type of element you wish to target for this Template Output Definition. A single Template Output Definition can only target one output type but you can add more than one Template Output Definition to your template if necessary.

In the **Filter XPath** textbox, you may optionally enter an XPath query which will be used to filter the elements which are transformed during the generation process.

The **Root Path** textbox should contain the directory within which the files will be generated.

The **Filename XPath** textbox allows you to customize the name of each generated file. This powerful field allows you to enter multiple XPath queries in order to build a directory structure or a complex filename.



Quick Start Guide

CodeGenerator v1.5.0

Generating

Once you are satisfied with the customizations you have made and the templates you have configured, you are ready to generate! Simply click the **Generate** button on the toolbar and you will see the progress displayed in the panel at the bottom of the CodeGenerator main window.



Quick Start Guide

CodeGenerator v1.5.0

Next Steps

Continue to experiment and try using new templates to generate more and more code, documentation, etc. Once you understand how CodeGenerator works, you will constantly think of new ways to make your job easier. When you do, let us know! We would love to hear about how you are innovating using this tool!

Support & Issues

Should you have trouble using CodeGenerator, you should visit <http://www.quantumconceptscorp.com> and contact us. If you would like to report an issue or request a feature, you may do so at <http://bugs.quantumconceptscorp.com>.