Tech Notes

Embarcadero® RAD Studio 2010:

UML and Audits and Metrics: Not the Old UML!

Michael Rozlog – Sr. Director of Delphi Solutions, Embarcadero Technologies

September 2009

Corporate Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

WHAT IS INCLUDED IN RAD STUDIO 2010?

As most developers know, Embarcadero RAD Studio, comes with both C++Builder® and Delphi®. The UML and static analysis features have been added to all levels of the product, so if you are using Delphi 2010 or C++Builder 2010 Professional, Enterprise, or Architect you will have access to some if not all the UML features in the product. For Delphi 2010 users, audits and metrics have been added to each level of the product, so even users of our Professional level products get some of the advanced features.

Below is a simplified list of features included in RAD Studio 2010:

	Personality:
	D=Delphi
	C=C++Builder
	P=Delphi Prism
		Architect	Enterprise	Pro

UML MODELING				
UML Code Visualization—at any time, get a UML model view of your source code	C D	>>	>>	>>
Diagrams have an updated look and feel	C D	>>	>>	>>
Sequence diagrams	C D	>>	>>	
Collaboration diagrams	C D	>>	>>	
State charts	C D	>>	>>	
Deployment diagrams	C D	>>	>>	
Use case diagrams	C D	>>	>>	
Activity diagrams	C D	>>	>>	
Component diagrams	C D	>>	>>	
Documentation generation	C D	>>	>>	
80+ Code Metrics that allow for deep understanding and static analysis of the health of the object domain	D	>>	>>	
Introduced for Delphi Professional in 2010! 10+ introductory level code Metrics for understanding the health of the object diagram	D			>>

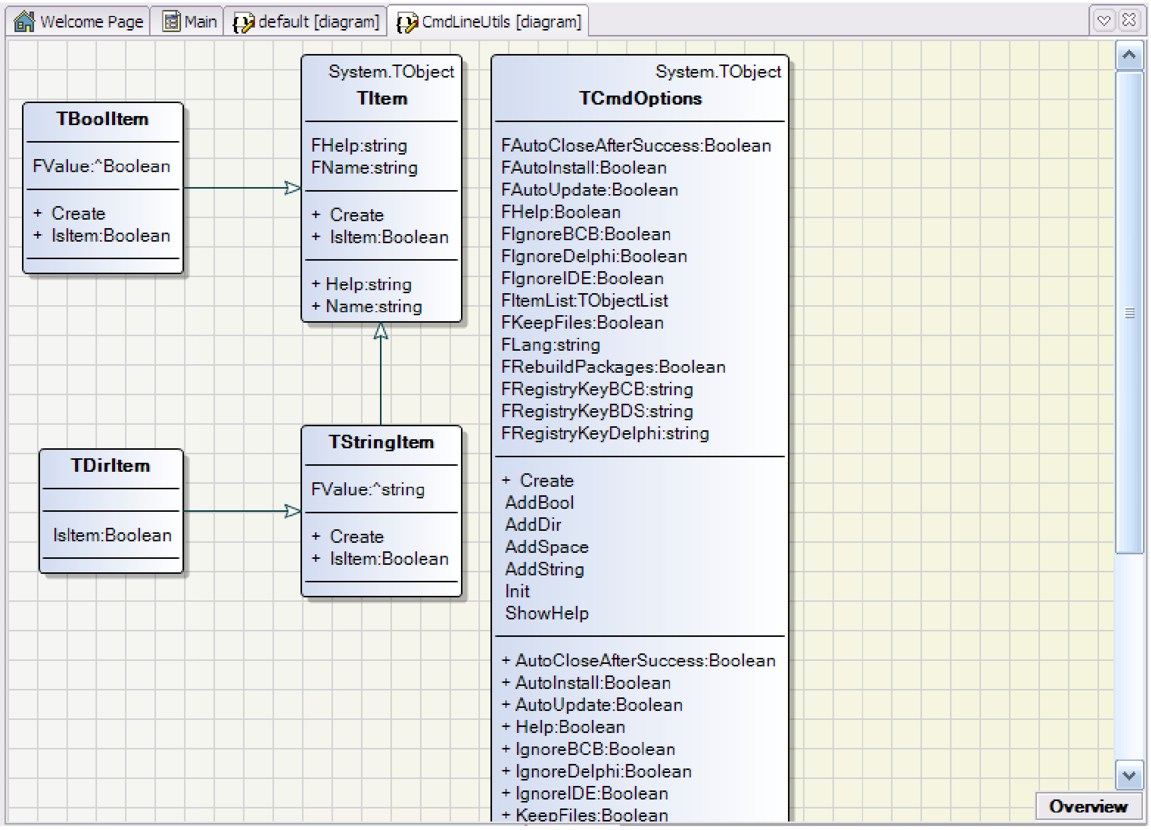
	Personality:
	D=Delphi
	C=C++Builder
	P=Delphi Prism
		Architect	Enterprise	Pro

200+ Code Audits for deep code analysis to better understand the coding style and approach	D	>>	>>	
Introduced for Delphi Professional in 2010! 10 introductory level code audits for better code understanding	D			>>
Import model from Rational® Rose (mdl import)	C D	>>		
Language neutral UML 1.5 and 2.0 modeling	C D	>>		
Transformation from language neutral UML 1.5 and UML 2.0 projects to source code projects	C D	>>		
Design Patterns support, including GOF patterns	C D	>>	>>	
Custom design patterns support	C D	>>	>>	
Diagram printing	C D	>>	>>	>>
XMI 1.1 Import/Export	C D	>>	>>	
OCL 2.0 support	C D	>>	>>	>>
"UML in Color" profile	C D	>>	>>	>>
Diagram hyper-linking and annotations	C D	>>	>>	>>

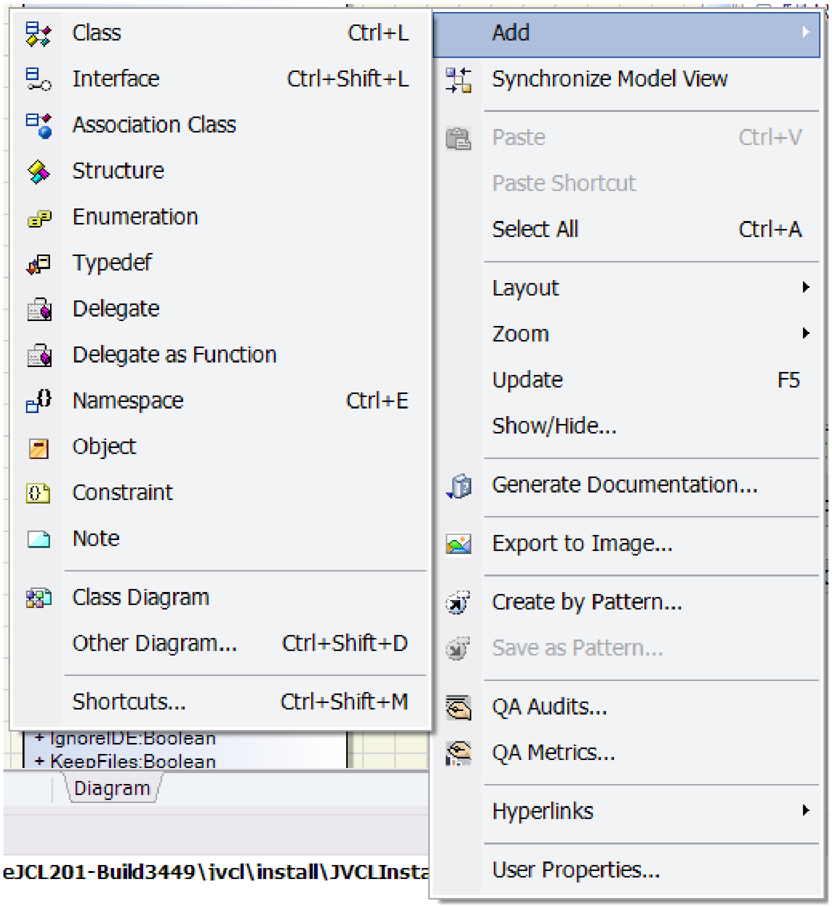
ARE THESE THE SAME UML FEATURES FROM YEARS PAST?

The short answer to that question is NO. It is still based on the same technology, but many, many, upgrades have occurred to make the product faster to load, faster to reverse and forward engineer, and presents a much more modern look. The same benefits of these tools still hold true today, that they help developers get more done in a shorter amount of time.

Today Delphi and C++Builder users can reverse engineer a class diagram at any time in all levels of the product.

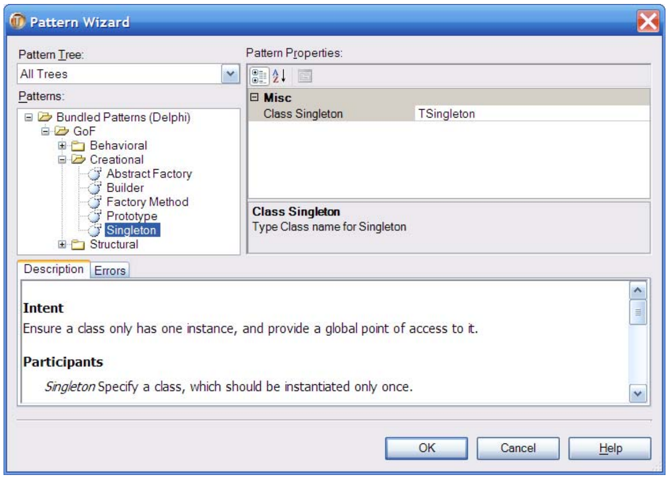


In the enterprise and above, developers can make changes to the code or model and both will be kept in synch with each other. So if the developer would right-mouse click on the above diagram and adds a Class, the code would also be generated in the unit for that class. This 2-way capability, a feature called LiveSource™, makes adding anything to the class diagram, and ultimately your code, as simple as a right-mouse click to the context menu shown below.



WHAT IS PATTERN SUPPORT?

As you can see from the above, the UML integration allows for support around Pattern, so if you want to create a GOF Singleton pattern, click the **Create by Pattern…** menu item and select it from the included patterns in the product:



Misc

Class Singleton: TSingleton

Class Singleton
Type Class name for Singleton

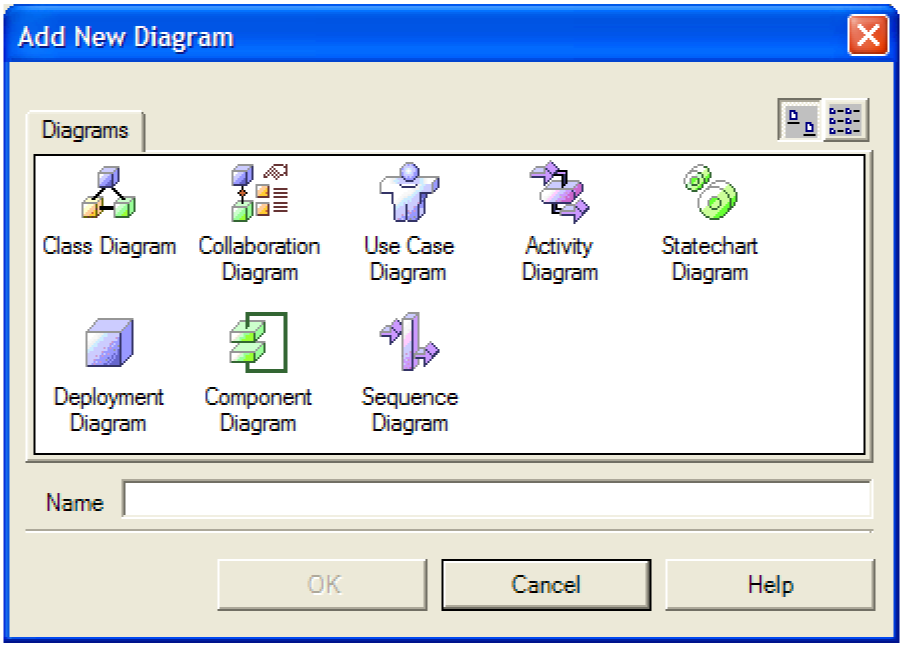
Intent
Ensure a class only has one instance, and provide a global point of access to it.

Participants
Singleton Specify a class, which should be instantiated only once.

Click the Ok button and the pattern will be added to the model and corresponding code will be generated.

A FULL UML PRODUCT…

In the Enterprise edition and above, developers have full UML integration meaning they can create any of the standard UML diagrams.



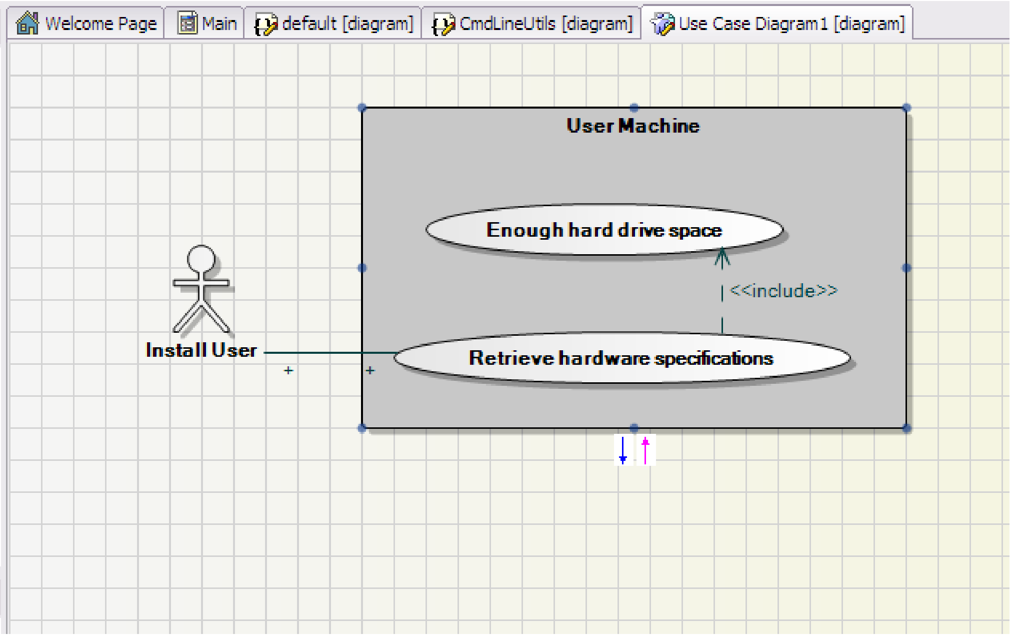
For those not familiar with what each diagram is really responsible for, the following is a list:

- UML – Activity Diagram Definition
  A diagram showing the step-by-step workflow of components in a system.
- UML – Class Diagram Definition
  A diagram showing the step-by-step workflow of components in a system.
- UML – Use Case Diagram Definition
  A high level graphical representation of functional requirements.
- UML – Component Diagram Definition
  A diagram used for showing physical components and their dependencies.
- UML – Collaboration Structure Diagram Definition
  A diagram that shows the internal structure of a class and its collaborations.
- UML – Deployment Diagram Definition
  A diagram for showing the hardware and its deployed components.
- UML – State Machine Diagram Definition
  A diagram that shows the different states/transitions an object.
- UML – Interaction (Sequence and Communication) Diagrams
  Diagrams that show Message Sequence and or interaction between objects and the sequences of messages.

Today, when it comes to support for UML, RAD Studio is making key technologies available to developers to help them communicate better, create better designs quicker, have the ability to set coding standards, and check the health of a Delphi application. Plus, it also supports automatic Documentation Generation, which all developers like.

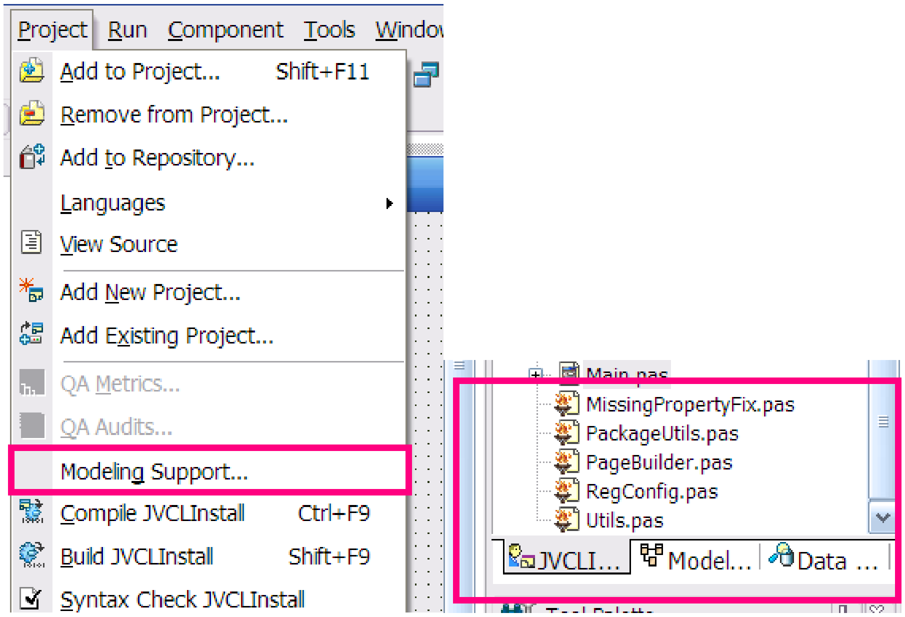
RAD Studio 2010 Architect adds support for both types of diagrams; ones based on source code and the other based on design. An example of a diagram based on source would be a Class diagram, which uses the LiveSource™ engine to automatically keep the code and model in-synch at all times. Whereas, a language neutral design is only based on the UML specification and no code is generated or needed. The tool also includes the ability to take these language neutral designs and generate source code for the designs, which gives the best of both worlds; the ability to create great object models, and if code is needed it can work with that just as easily.

This is a great time saver and adds significant piece of mind knowing that when a change is made in either the code or the model, that both areas will be up-to-date. A design diagram (with no source code dependencies) would be a Use Case; this communicates high-level functional requirements graphically.

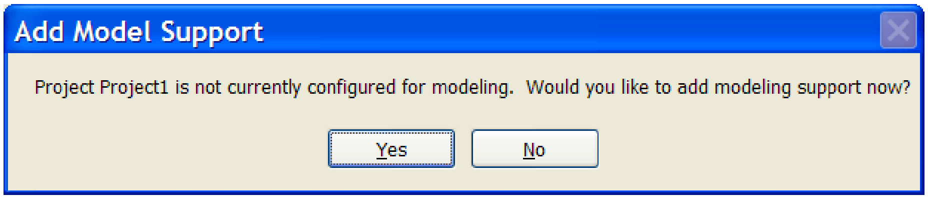


RAD STUDIO 2010 UML INTEGRATION EASY TO USE

RAD Studio 2010 makes it extremely easy to use UML and takes advantage of the advanced tools. First, either start with a new project like you would normally do, or open up an existing project. Once the project is open click on the Model-View tab under the project manager or Project|Model Support… menu item.



Clicking either of these on the open project will activate the UML integration and present a new dialog:



Project Project1 is not currently configured for modeling. Would you like to add modeling support now?

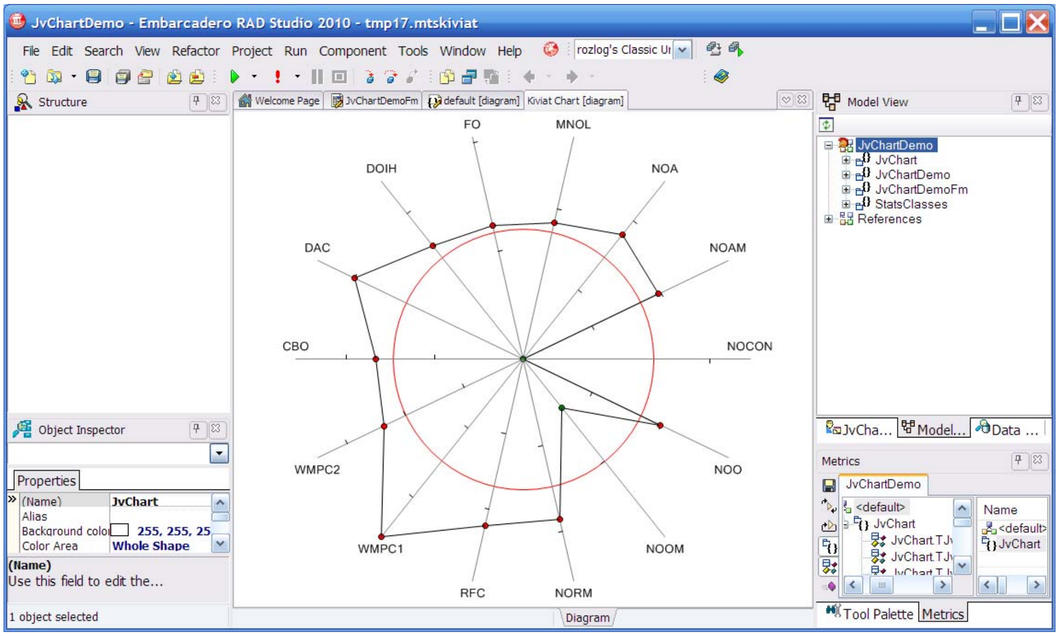
Simply click the Yes button and you're on your way. This will then open the design surface and will use the LiveSource™ technology to reverse the project and give you a Class diagram as shown above. Again, any changes made in the model will be reflected in the code, so if you change the model, like a string attribute to a new name, that name will now be in the code as well.

From this point you can add more objects, setup associations, aggregations, and all things object oriented. You can also add new models to the project and these industry standard diagrams, including use case diagrams, class diagrams, and sequence diagrams, etc., to help developers using UML to communicate.

Introduction to Metrics

The Delphi 2010 part of RAD Studio also brings more than simple modeling to the tool. You can use Delphi 2010 to analyze your existing projects, and generate metrics and audits that you can use to uncover potential problems with your applications. For example, the following figure shows a generated Kiviat chart. A Kiviat chart depicts a best practices circle. Points that are within the inner circle are considered acceptable, and those outside denote areas that may deserve your attention.

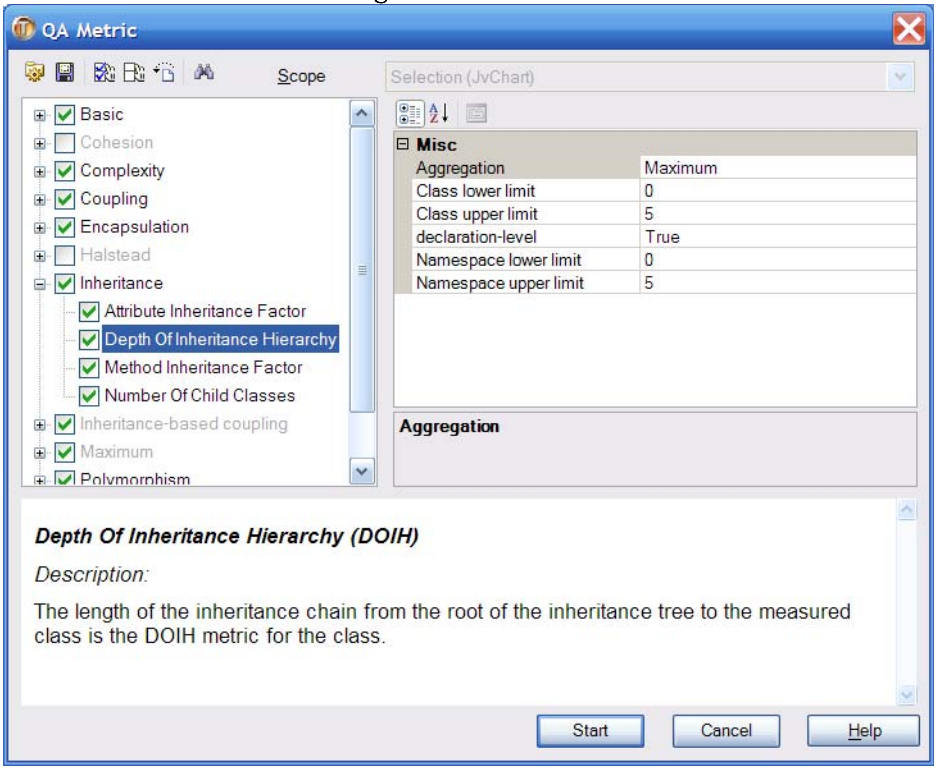
For example in the graph shown here, the depth of inheritance hierarchy (DOIH) appears outside the user defined inner circle. This metric indicates that one or more of your classes are abnormally deep, with respect to the root class (Object or TObject).



(Name)	JvChart
Alias	
Background colo	255, 255, 25
Color Area	Whole Shape

(Name)
Use this field to edit the...

You set the limits for the metrics used by Metrics. The following figure shows the QA Metric dialog box where these metrics are configured.



- ☑ Basic
- □ Cohesion
- ☑ Complexity
- ☑ Coupling
- ☑ Encapsulation
- □ Halstead
- ☑ Inheritance
  - ☑ Attribute Inheritance Factor
  - ☑ Depth Of Inheritance Hierarchy
  - ☑ Method Inheritance Factor
  - ☑ Number Of Child Classes
- ☑ Inheritance-based coupling
- ☑ Maximum
- ☑ Polymorphism

Misc	
Aggregation	Maximum
Class lower limit	0
Class upper limit	5
declaration-level	True
Namespace lower limit	0
Namespace upper limit	5

Aggregation

Depth Of Inheritance Hierarchy (DOIH)

Description:

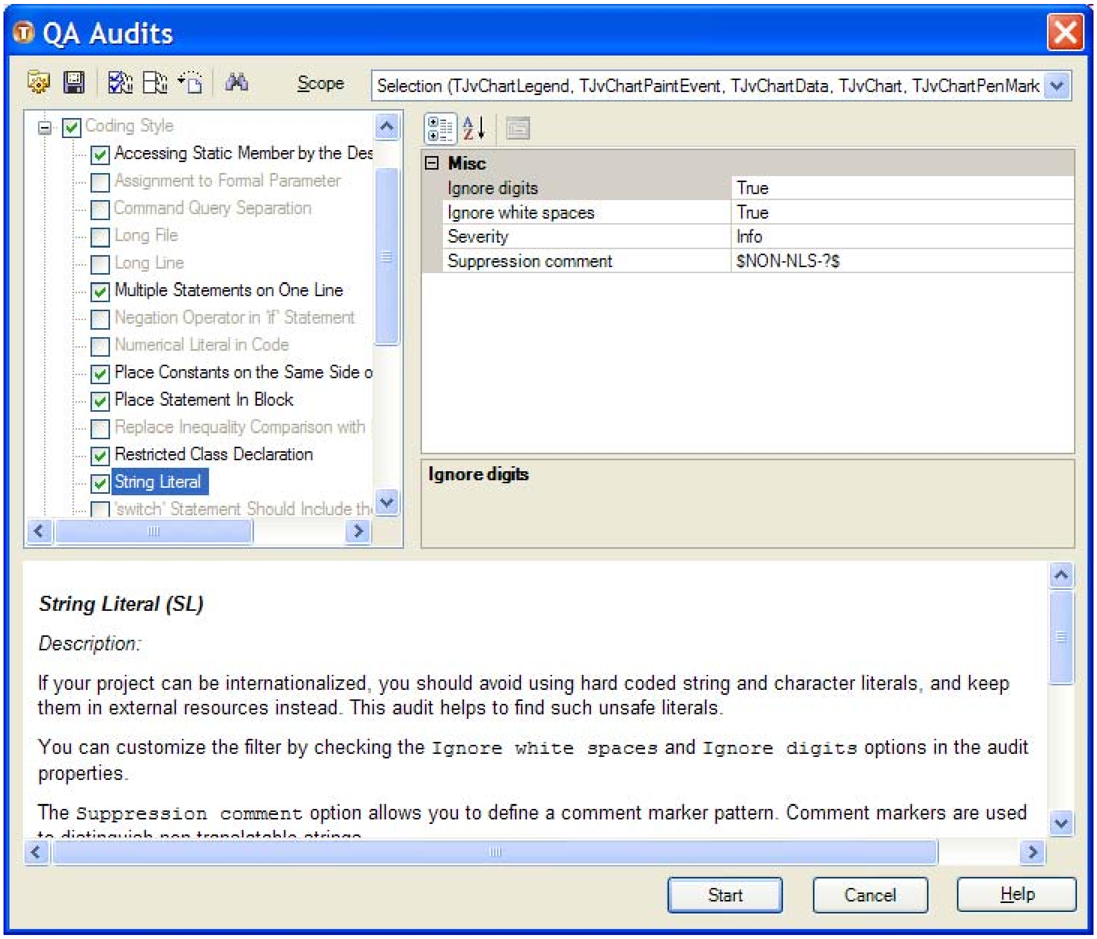
The length of the inheritance chain from the root of the inheritance tree to the measured class is the DOIH metric for the class.

Delphi 2010 covers the major areas of Metrics, with over 10+ included in professional and 80+ in the Enterprise and above. Below is a high-level table of which ones are included:

- Basic
- Cohesion
- Complexity
- Encapsulation
- Halstead
- Inheritance
- Inheritance-based coupling
- Maximum
- Polymorphism
- Ratio

INTRODUCTION TO CODE AUDITS

The use and configuration of audits is similar to metrics. The following figure shows the QA Audits dialog box, which contains the audits equivalent to the QA Metrics dialog box.



Scope: Selection (TJvChartLegend, TJvChartPaintEvent, TJvChartData, TJvChart, TJvChartPenMark

☑ Coding Style
- ☑ Accessing Static Member by the Des
- □ Assignment to Formal Parameter
- □ Command Query Separation
- □ Long File
- □ Long Line
- ☑ Multiple Statements on One Line
- □ Negation Operator in 'if' Statement
- □ Numerical Literal in Code
- ☑ Place Constants on the Same Side o
- ☑ Place Statement In Block
- □ Replace Inequality Comparison with
- ☑ Restricted Class Declaration
- ☑ String Literal
- □ 'switch' Statement Should Include th

Misc	
Ignore digits	True
Ignore white spaces	True
Severity	Info
Suppression comment	$NON-NLS-?$

Ignore digits

String Literal (SL)

Description:

If your project can be internationalized, you should avoid using hard coded string and character literals, and keep them in external resources instead. This audit helps to find such unsafe literals.

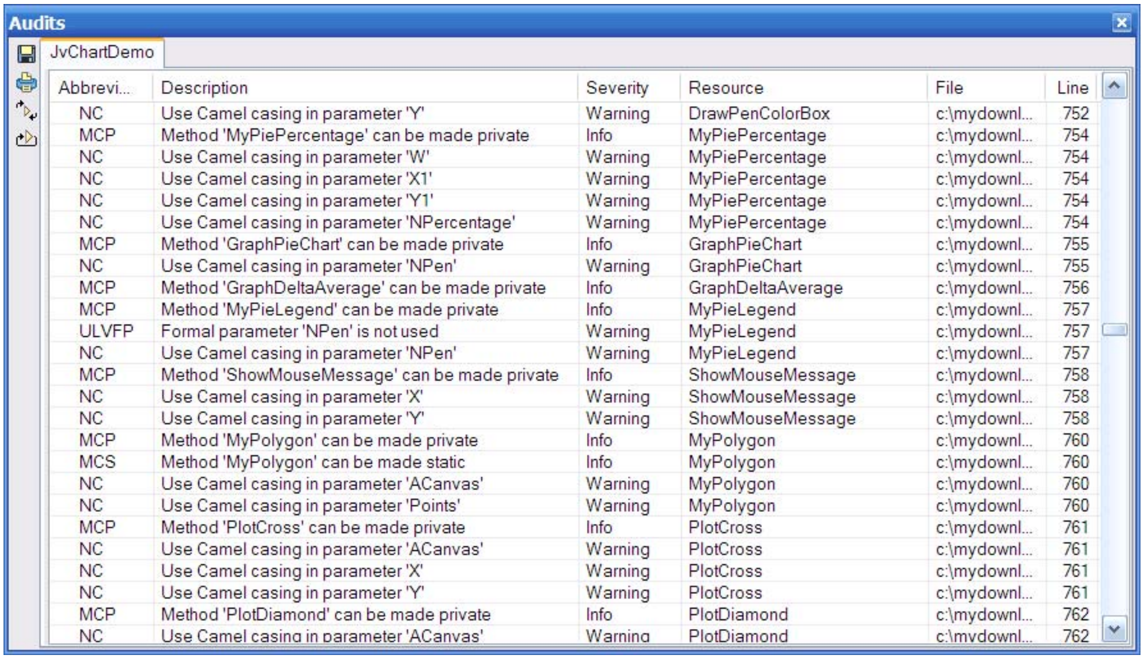
You can customize the filter by checking the `Ignore white spaces` and `Ignore digits` options in the audit properties.

The `Suppression comment` option allows you to define a comment marker pattern. Comment markers are used to distinguish non-translatable strings

Delphi 2010 covers the major areas of Audits, with over 10+ included in professional and 200+ in the Enterprise and above. Below is a high-level table of which ones are included:

- Arrays and References
- Branches and Loops
- Coding Style
- Declaration Style
- Duplicated Code
- Superfluous Content
- Design flaws
- Expressions
- Naming Style
- Performance
- Portability
- Possible Errors

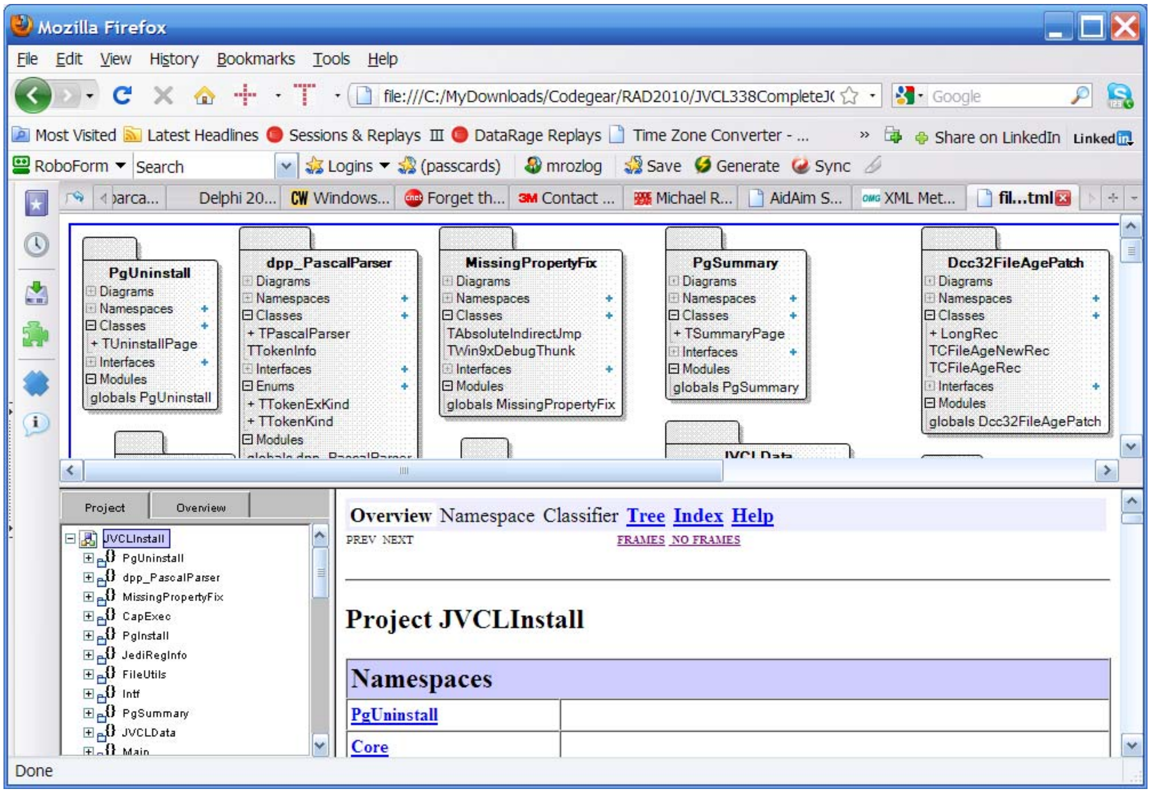
The static analysis using code Audits will look at every single line of every single method of every single class, or every single namespace, and generate a list of faults that you define. This helps with poor code and refactoring.



Abbrevi...	Description	Severity	Resource	File	Line
NC	Use Camel casing in parameter 'Y'	Warning	DrawPenColorBox	c:\mydownl...	752
MCP	Method 'MyPiePercentage' can be made private	Info	MyPiePercentage	c:\mydownl...	754
NC	Use Camel casing in parameter 'W'	Warning	MyPiePercentage	c:\mydownl...	754
NC	Use Camel casing in parameter 'X1'	Warning	MyPiePercentage	c:\mydownl...	754
NC	Use Camel casing in parameter 'Y1'	Warning	MyPiePercentage	c:\mydownl...	754
NC	Use Camel casing in parameter 'NPercentage'	Warning	MyPiePercentage	c:\mydownl...	754
MCP	Method 'GraphPieChart' can be made private	Info	GraphPieChart	c:\mydownl...	755
NC	Use Camel casing in parameter 'NPen'	Warning	GraphPieChart	c:\mydownl...	755
MCP	Method 'GraphDeltaAverage' can be made private	Info	GraphDeltaAverage	c:\mydownl...	756
MCP	Method 'MyPieLegend' can be made private	Info	MyPieLegend	c:\mydownl...	757
ULVFP	Formal parameter 'NPen' is not used	Warning	MyPieLegend	c:\mydownl...	757
NC	Use Camel casing in parameter 'NPen'	Warning	MyPieLegend	c:\mydownl...	757
MCP	Method 'ShowMouseMessage' can be made private	Info	ShowMouseMessage	c:\mydownl...	758
NC	Use Camel casing in parameter 'X'	Warning	ShowMouseMessage	c:\mydownl...	758
NC	Use Camel casing in parameter 'Y'	Warning	ShowMouseMessage	c:\mydownl...	758
MCP	Method 'MyPolygon' can be made private	Info	MyPolygon	c:\mydownl...	760
MCS	Method 'MyPolygon' can be made static	Info	MyPolygon	c:\mydownl...	760
NC	Use Camel casing in parameter 'ACanvas'	Warning	MyPolygon	c:\mydownl...	760
NC	Use Camel casing in parameter 'Points'	Warning	MyPolygon	c:\mydownl...	760
MCP	Method 'PlotCross' can be made private	Info	PlotCross	c:\mydownl...	761
NC	Use Camel casing in parameter 'ACanvas'	Warning	PlotCross	c:\mydownl...	761
NC	Use Camel casing in parameter 'X'	Warning	PlotCross	c:\mydownl...	761
NC	Use Camel casing in parameter 'Y'	Warning	PlotCross	c:\mydownl...	761
MCP	Method 'PlotDiamond' can be made private	Info	PlotDiamond	c:\mydownl...	762
NC	Use Camel casing in parameter 'ACanvas'	Warning	PlotDiamond	c:\mydownl...	762

AUTOMATIC DOCUMENTATION GENERATION

One of the other major features found inside the UML integration is automatic Documentation Generation. Notice in the below figure that complete developer documentation has been produced by a simple click of a menu item.



Project JVCLInstall

Namespaces	
PgUninstall	
Core	

These advanced tools really help all of the developers out there, from the Architects to the newbie just starting with RAD Studio 2010. The advanced tools like LiveSource™ to keep the code and model in-synch, and the ability to generate new models for better communications, is great. It is also nice that advanced tools like static Code Audits and Metrics are available, especially if large amounts of code is being inherited or a team wants to know where the code really is as far as health and standards. Finally, the ability to generate the documentation with hyperlinks, which links the various models together, is a real time saver and allows developers to keep up-to-date with a click of the button.

About the Author

Mike Rozlog is the senior director of products for Embarcadero Technologies. In this role, he is focused on ensuring the developer focused products being created by Embarcadero meet the expectations of developers around the world. Much of his time is dedicated to discussing and explaining the technical and business aspects of Embarcadero's products and services to analysts and other audiences worldwide. Mike was formerly with CodeGear™, a developer tools group that was acquired by Embarcadero in 2008. Previously, he spent more than eight years working for Borland® in a number of positions, including a primary role as Chief Technical Architect. A reputed author, Mike has been published numerous times. His latest collaboration is Mastering JBuilder from John Wiley & Sons, Inc. Mike is online at http://blogs.embarcadero.com/michaelrozlog/.

EMBARCADERO
TECHNOLOGIES.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, Embarcadero® RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.