Product Data Sheet

Codagen Architect 3.2

The leading Model Driven Architecture® tool for model transformation, code generation, and automation of your development process

Transform your software architecture into a reusable asset!

The adoption of model-based approaches is gaining momentum, and Codagen Architect fills the gap in the "software production line" between business and technology - whether the development is in-house, onshore, or offshore. Codagen Architect allows you to maintain a separation of the business logic from its implementation. At the same time it bridges the design-to-implementation gap by creating the rules that transform business models into source code and transform your software architecture into reusable assets. You are free to focus on the code specific to your business, allowing it to evolve independently from the technology infrastructure. Bridging the gap between business and technology, Codagen Architect is perfectly in step with the direction toward a mature, agile software development process that effectively links the models, design patterns, frameworks, and code.

Codagen Architect integrates seamlessly with your favorite UML™ modeling tool and works with your existing models to enforce your software architecture while generating up to 90% of the application source code. By implementing the best practices and design patterns related to the framework APIs, Codagen Architect eases the adoption and evolution of any in-house or commercial framework like J2EE or .NET . Your choice of programming languages includes Microsoft Visual Basic, C#, C++, and Java.

Features at a Glance

- Fully customizable model-to-code transformation
- XML-based code generation templates: no scripting language or API to learn
- Compatible with the most popular UML™ modeling tools
- Support for XMI
- Generates code for class, activity, statechart, sequence, collaboration, and use case UML diagrams
- Compliant with OMG's Model Driven Architecture® (MDA)
- Generates code in Java, Visual Basic, C#, and C++
- Generates documentation, Web pages, SQL scripts, and more!
- Includes Technology Accelerators™: ready-made templates that provide off-the-shelf code generation

UNIFIED MODELING LANGUAGE™

MODEL DRIVEN ARCHITECTURE™

Microsoft® Visual Studio .net Integration Program

software development 13th annual productivity award

"The ACORD and IBM IAA standards are the dominant business models in the insurance industry. Codagen Architect manages the architectural complexity associated with implementing these models, and does so using an MDA-compliant approach. The Codagen tool uses these platform independent business models as targets for generating the source code and supporting artifacts necessary for an agile IT portfolio of services, components, and inter-enterprise business integration."

Bill Nadal, Chief Architect,
Fireman's Fund Insurance Company

"I rely on Codagen to produce source code specifically for the Microsoft .NET platform that is consistent with our design team's principles and standards. Since the source code produced is based on a template-driven approach, our design patterns are preserved and the quality of the software solution is greatly improved. Better quality code and reduced software life cycle both lead to a reduction in the total cost of ownership."

Aaron G. Daisley-Harrison,
Director, Fujitsu Consulting

"We never saw a tool that could really answer the code generation need that we were looking for in the past. When we understood Codagen and realized it was completely different from the others in the market, it was like "Wow!" We've been waiting for this kind of tool for years. I planned a month and a half to build the J2EE templates, but in one week, everything was done. It's delivering what we were expecting."

David Bertrand,
Technical Architect, CGI

Delivering Architectural Guidance

One of the key challenges facing application development projects is the ability to capture the knowledge of the most experienced developers and architects and share that knowledge across development teams. Using transformation templates, architects can establish implementation best practices. For example, a distributed application might contain pre-defined blocks of code for three architectural layers: user interface, business logic, and data access. The flexibility of the Codagen Architect transformation engine allows for the generation of source code that is derived from the UML model elements, thus making the templates generic enough to be used by many projects sharing a common architecture, infrastructure, or platform. In the transformation templates, an application architect includes standard components as well as any other reusable assets that are appropriate for the corporate application infrastructure. It is also possible to generate code for a custom framework in the event that your organization has created one. It then becomes much easier to guide other architects, developers, and modelers in applying the framework consistently across projects.

Geared to Productivity

Codagen Architect 3.2 features Technology Accelerators™, pre-built transformation templates and UML profiles that are ready to be used in your development projects. The Codagen library of Technology Accelerators is constantly expanding to reflect the latest technologies, frameworks, and best practices. While they can be used as learning aids, the Technology Accelerators™ are so comprehensive that they can be used in real projects. In the past, our customers have achieved dramatic productivity gains when building transformation templates from scratch. Imagine the productivity gains you'll experience by leveraging the best practices of some of the leaders in application development. Furthermore, because the templates are not locked in any way, your organization can fully rework them if appropriate.

Available Technology Accelerators™:

For data access:
- Enterprise Java Beans (EJBs)
- ADO.NET object-to-relational mapping

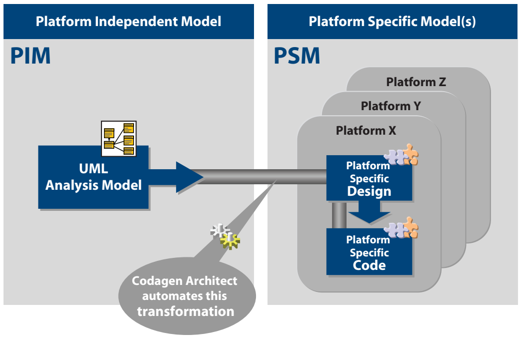
For web page navigation:
- Struts framework (Apache Jakarta project)
- ASP.NET with Model-View-Controller (MVC) pattern
- WAF (Sun Pet Store) with MVC pattern

Transforming Your Software Architecture into a Reusable Asset with MDA

The MDA initiative of the Object Management Group (OMG) seeks to create a specification that not only protects the investment in UML when the domain layer of an application is modeled but also provides adequate agility when implementation changes are required. If an enterprise's domain is captured inside a technology-independent UML model (a so-called platform independent model or PIM), then the model serves a function that is similar to that of an analysis model because it represents the logical view in which the composition and behavior of all components (but not their implementation) are fully specified.

Since the PIM, by definition, does not contain technology details, it can be mapped to one or more platform specific models (PSM). The PSM, which serves the function of a design model, represent the source code or its UML representation.

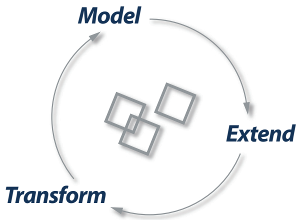
Codagen Architect is fully compliant with MDA, providing a self-documenting approach that protects the intellectual property of the architecture by allowing architects to automate the process of transforming a PIM into a (set of) PSM(s), as illustrated in this figure:



New to Codagen Architect 3.2

- Support for MagicDraw and Rational XDE
- Graphical UML adapter for any modeling tool compliant with XMI
- Ability to save transforamtion rules (Architect project) in XML
- Improved UML support: works with collaboration and sequence diagrams in addition to statechart, activity, class, and use case diagrams
- Predefined generation templates containing useful model-to-code transformations for common situations
- Improved tutorials and samples

Code Generation: As Easy as Model, Extend, Transform!



MODEL. UML modeling tools and notation bring a powerful, yet unified, approach to application development. At a higher level of abstraction, analysis models describe the domain for which an application is built. Capturing the business requirements and developing the associated analysis models provide the knowledge that is indispensable for building applications that can weather the evolution of technologies and frameworks.

EXTEND. Codagen Architect integrates seamlessly with your modeling tool and allows you to extend UML models using transformation markers. These markers encapsulate your logical architecture by defining architecture issues and related design decisions associated with the platform you have chosen, be it J2EE, .NET, or another. Codagen Architect also lets you define an integration schema that describes how the common services provided by your software infrastructure are reused by your business objects, thereby making your modeling tool "aware" of your environment and clearly leveraging your existing investment in UML. Codagen's fully customizable UML profiles allow for effective mapping between an analysis model and its associated, automatically created design model, which comprises the implementation details.

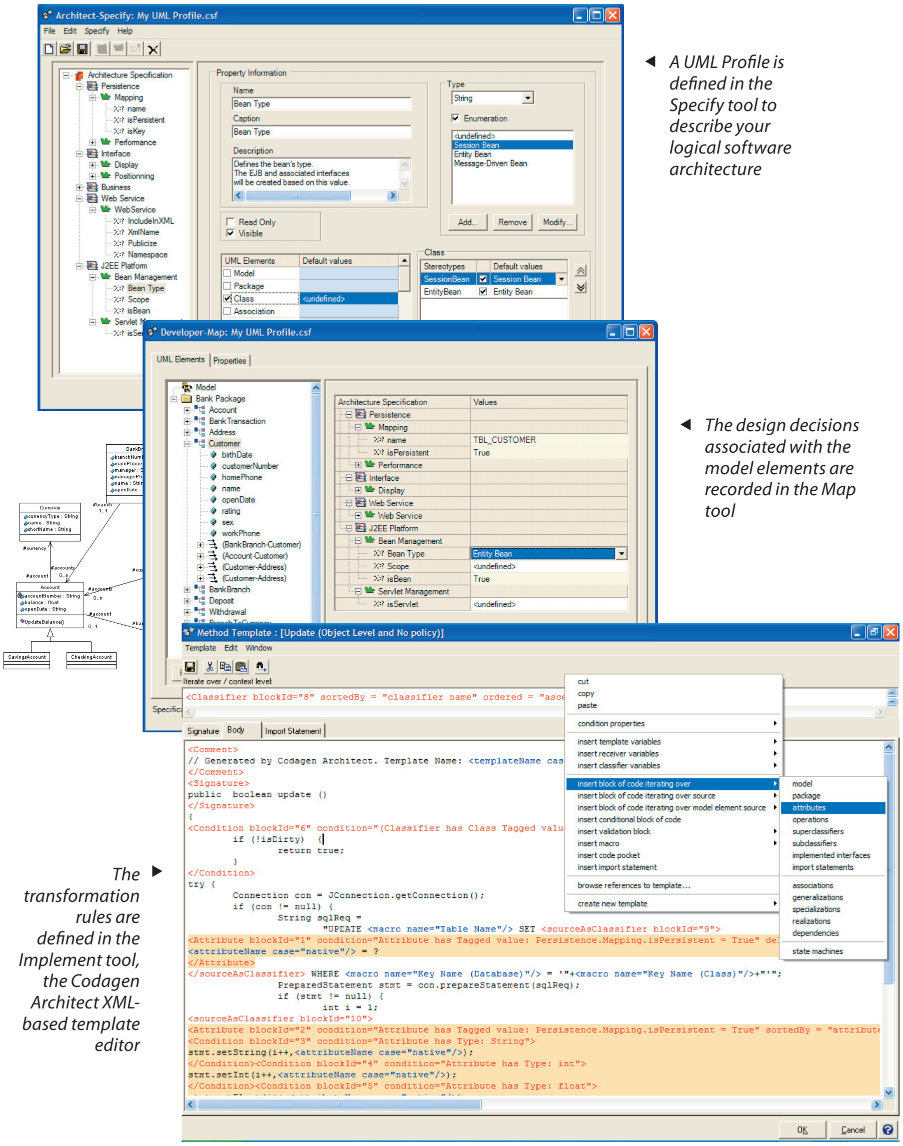
TRANSFORM. Codagen Architect provides an open XML-based environment that helps you create the transformation rules (i.e., templates) that represent the physical implementation of your logical architecture. Without the need for a scripting language, you define the model-to-code transformation using the intuitive Codagen Architect template editor. Inserted XML tokens control the logic of code generation and provide flexible navigation over the model elements and recorded design decisions. The generated code retains the same quality as code written by your most talented programmers since Codagen Codagen Architect lets you take full control of the transformation process.

Codagen Architect Benefits

- Accelerates the application development cycle
- Standardizes the development process while lowering the cost of most phases
- Dramatically increases the level of code reuse in development projects
- Allows for the evolution of the developed application independent from its underlying technologies
- Builds in portability and interoperability
- Protects organization's valuable intellectual property
- Reduces the learning curve associated with the implementation of a technology or framework

Product Features

- Fully customizable model-to-code and model-to-model transformation
- Modeling tool adapters for Rational Rose (any edition), Borland Together ControlCenter, Microsoft Visio UML, MagicDraw, and Rational XDE
- Graphical UML adapter for any modeling tool compliant with XMI
- Support for various UML diagrams: class, statechart, activity, sequence, collaboration, and use case
- Support for the J2EE and .NET platforms
- Full-blown, ready to use Technology Accelerators™ for the most popular technologies and frameworks
- Compliance with the OMG's Model Driven Architecture™
- Generation of source code in Java, C#, C++, and Visual Basic
- Generation of text-based files such as XML, HTML, JSP, SQL scripts, etc.
- Open, XML-based, model transformation templates featuring rich contextual functions for navigating in models and generated code
- Compatibility with both commercial and in-house frameworks
- Ability to create Code Pockets™: distinctive areas delimited in the templates and source code for sections manually maintained by developers
- Integrated, non-disruptive regeneration over existing source code
- Generated source code that does not require a proprietary run-time library or framework



A UML Profile is defined in the Specify tool to describe your logical software architecture

The design decisions associated with the model elements are recorded in the Map tool

The transformation rules are defined in the Implement tool, the Codagen Architect XML-based template editor

Architect-Specify: My UML Profile.csf

Property Information
Name: Bean Type
Caption: Bean Type
Description: Defines the bean's type. The EJB and associated interfaces will be created based on this value.
Type: String
☑ Enumeration
<undefined>
Session Bean
Entity Bean
Message-Driven Bean
□ Read Only
☑ Visible

UML Elements	Default values
□ Model	
□ Package	
☑ Class	<undefined>
□ Association	

Class

Stereotypes		Default values
SessionBean	☑	Session Bean
EntityBean	☑	Entity Bean

Developer-Map: My UML Profile.csf

Architecture Specification	Values
Persistence	
Mapping	
name	TBL_CUSTOMER
isPersistent	True
Performance	
Interface	
Display	
Web Service	
Web Service	
J2EE Platform	
Bean Management	
Bean Type	Entity Bean
Scope	<undefined>
isBean	True
Servlet Management	
isServlet	<undefined>

Method Template : [Update (Object Level and No policy)]

Iterate over / context level:
```
<Classifier blockId="8" sortedBy = "classifier name" ordered = "asce
```

Body:
```
<Comment>
// Generated by Codagen Architect. Template Name: <templateName cas
</Comment>
<Signature>
public  boolean update ()
</Signature>
{
<Condition blockId="6" condition="(Classifier has Class Tagged valu
        if (!isDirty)  {
                return true;
        }
</Condition>
try {
        Connection con = JConnection.getConnection();
        if (con != null) {
                String sqlReq =
                        "UPDATE <macro name="Table Name"/> SET <sourceAsClassifier blockId="9">
<Attribute blockId="1" condition="Attribute has Tagged value: Persistence.Mapping.isPersistent = True" de
<attributeName case="native"/> = ?
</Attribute>
</sourceAsClassifier> WHERE <macro name="Key Name (Database)"/> = '"+<macro name="Key Name (Class)"/>+"'";
                PreparedStatement stmt = con.prepareStatement(sqlReq);
                if (stmt != null) {
                        int i = 1;
<sourceAsClassifier blockId="10">
<Attribute blockId="2" condition="Attribute has Tagged value: Persistence.Mapping.isPersistent = True" sortedBy = "attribute
<Condition blockId="3" condition="Attribute has Type: String">
stmt.setString(i++,<attributeName case="native"/>);
</Condition><Condition blockId="4" condition="Attribute has Type: int">
stmt.setInt(i++,<attributeName case="native"/>);
</Condition><Condition blockId="5" condition="Attribute has Type: float">
```

Codagen

Codagen Technologies Corporation
2075 University, Suite 1020
Montreal (Quebec)
Canada
H3A 2L1

www.codagen.com