

## IBM Rational Modeling Extension for Microsoft .NET

---

### Highlights

---

- ***Supports UML-based model-driven development of applications to be implemented fully or partially on the Microsoft .NET platform***
- ***Integrates with and complements Microsoft Visual Studio 2005 technology***
- ***Supports conceptual modeling of architectures and applications using UML***
- ***Offers high extensibility through its foundation on the open Eclipse 3.4 platform***
- ***Supports UML-to-C# and C#-to-UML transformations***
- ***Complements other IBM Rational Software Delivery Platform offerings to provide a comprehensive, full-lifecycle governance solution for enterprises supporting mixed-mode development***
- ***Installs into an IBM Rational architecture management tool such as IBM Rational Software Modeler or IBM Rational Systems Developer software***

Software architects and senior developers within a development team play a pivotal role in bringing technical professionals together. They need a powerful and configurable environment to help them create and maintain the architecture, despite challenges like heterogeneous applications and globally distributed development teams. Other members of the delivery team need to do their part as they all collaborate—regardless of application complexity and geographic location—to complete the software project.

### **Bringing open-standard UML to bear on your Microsoft .NET development**

IBM Rational® Modeling Extension for Microsoft® .NET software, when combined with the IBM Rational Software Modeler product or one of its siblings, is a powerful, integrated design and construction environment that helps with understanding, designing, managing and evolving enterprise solutions and services across the team, across the world and across different areas of technical expertise. Together, these products provide extensive support for the Object Management Group's (OMG's) industry-standard Unified

Modeling Language (UML), as well as many powerful domain-specific visual modeling capabilities. The Rational Modeling Extension for Microsoft .NET product is designed to work with other development tools in a variety of process environments to improve productivity, enhance architectural control and ease the design-to-code experience across a project portfolio that might include Microsoft .NET/C#; Java™/Java Platform, Enterprise Edition (Java EE); Web services; service-oriented architecture (SOA); and C/C++ applications.

UML serves as the common architectural language that bridges all projects regardless of the implementation technologies they target. The UML modeling capabilities available for use with Rational Modeling Extension for Microsoft .NET include object diagram support and extended capabilities for modeling activities, composite structures, components, interactions (sequence and communication diagrams) and state machines—in addition to providing use-case, class and deployment diagramming capabilities.

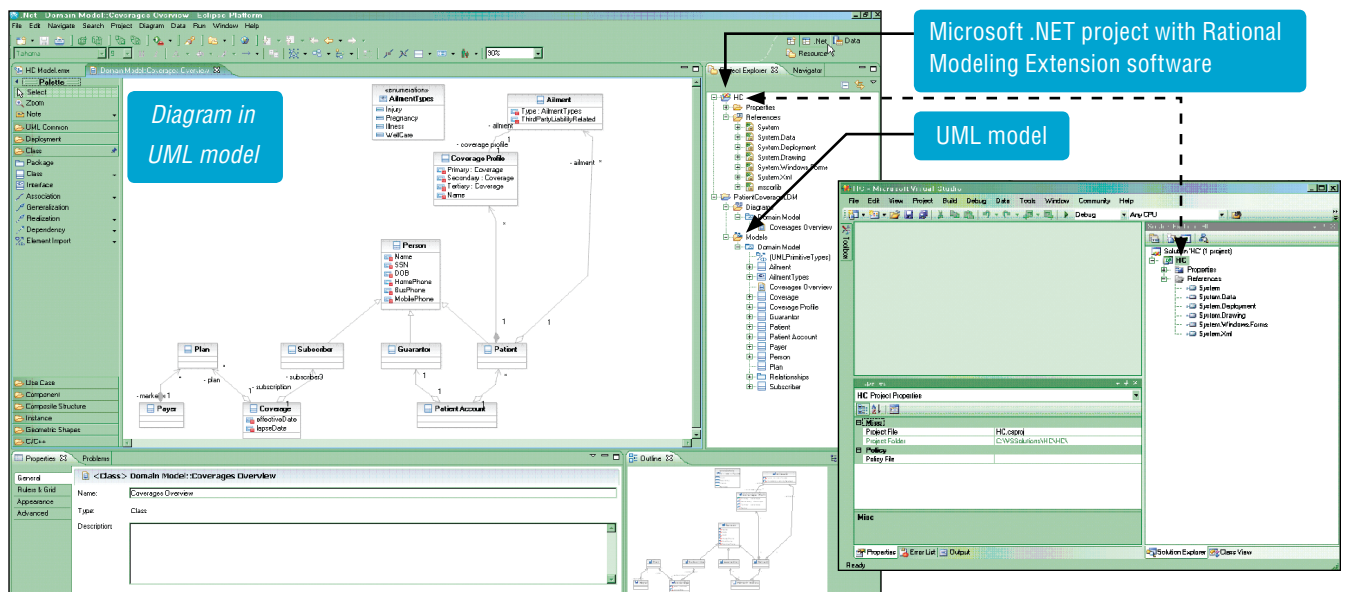


Figure 1: Rational Modeling Extension for Microsoft .NET software works with host products to enable you to more quickly build the content of UML analysis and design models, or it can help you create your own UML-based patterns.

### Keeping your costs down and your options open

IBM Rational Modeling Extension for Microsoft .NET is part of a flexible family of business-driven development products that are built on the open Eclipse 3.4 integration platform. Providing the ability to specify requirements, architectures and designs across heterogeneous project portfolios that include .NET projects, Rational Modeling Extension for Microsoft .NET integrates with Microsoft Visual Studio 2005 technology, as well as with a full set of lifecycle tools (for example, the IBM Rational Software Delivery Platform) that also integrate with the Visual Studio 2005 platform. This enables the delivery team to learn and administrate a single set of tools, and then use those skills across virtually all projects in the portfolio, whether they target Microsoft .NET, Java EE, UNIX®/C++, IBM System z® or other platforms. And these tools scale to meet nearly any challenge in terms of team size, organization or geographic distribution.

As we all know, the one constant in today's world is change. Are you currently favoring Microsoft .NET technology as a target platform but want to keep your options open for a possible future technology migration? You can help position yourself by adopting the UML approach with tools that support a variety of implementation technologies.

### Tapping into the power of model-based automations

Modeling alone adds value and helps reduce project risk, but even greater benefits are realized when models are used to automate the creation of other development artifacts, including other models, code and more. Rational Modeling Extension for Microsoft .NET plugs in to host products such as Rational Software Modeler. The host products include UML-based design patterns to help you more quickly build the content of your UML analysis and design models. Or, you can create your own UML-based patterns to extend this benefit even further.

### Improving productivity with transformation automation

Depending on which host product is used, it may also include transformations for generating Java, C++, Enterprise JavaBeans (EJB), Web Services Description Language (WSDL), XML Schema Definition (XSD), CORBA Interface Description Language (IDL) and structured query language (SQL)-based logical data models from UML models, as well as reverse transformations from the Java language and C++ to UML. Then, Rational Modeling Extension for Microsoft .NET extends its host to provide transformations from UML to C#, and from C# to UML, as well as a capability to compare conceptual models (UML) and code models (C#) to determine whether a C# implementation conforms to a specified architecture and design contract.

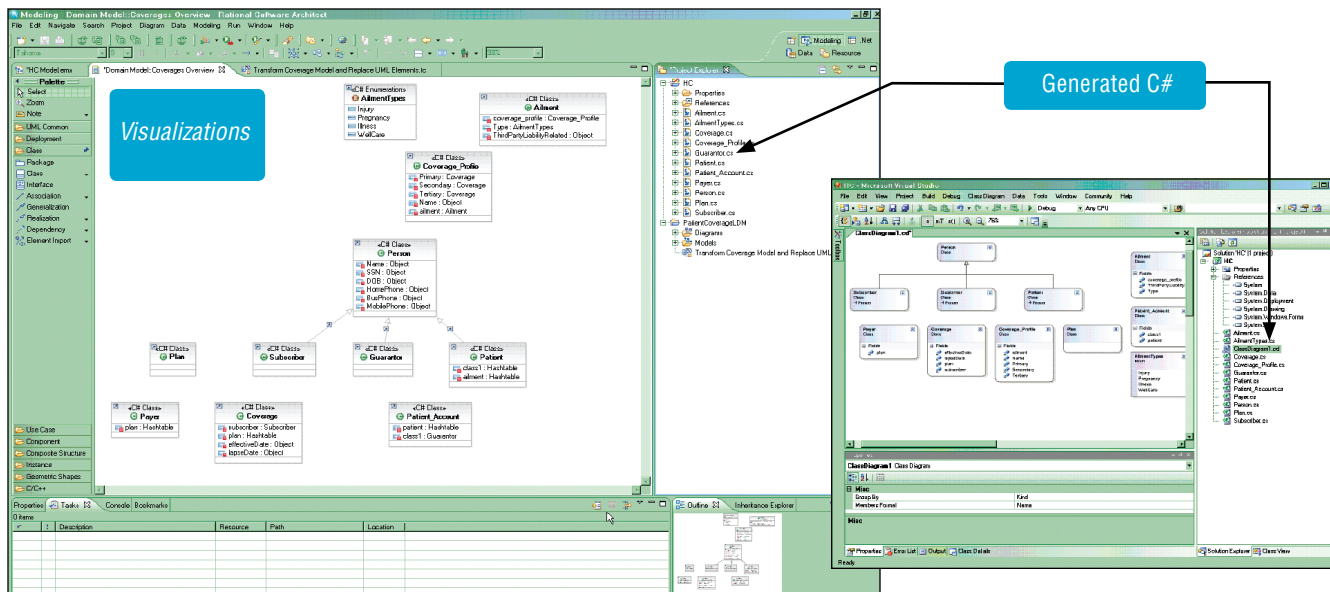


Figure 2: Extending the capabilities of its host product, Rational Modeling Extension for Microsoft .NET software supports transformations from UML to C#, and from C# to UML.

And note that the models that analysts and architects create with Rational Modeling Extension for Microsoft .NET also work with other IBM model-driven development products such as IBM Rational Software Architect, IBM Rational Software Modeler and IBM Rational Systems Developer software.

### Modeling flexibility for teams

Rational Modeling Extension for Microsoft .NET, through its host products, offers great flexibility in managing UML models. Capabilities for model fusing (such as “combine these models”) and model decomposition (such as “make this package into its own model”) let you evolve the logical organization of model content into model files that adapt to changing needs. Model files can be further deconstructed into smaller, transparent fragments—files that are version controllable at a modular level to sup-

port team development approaches based on exclusive checkout policies. And if you prefer to follow true parallel development policies with nonexclusive checkouts, the Rational Modeling Extension for Microsoft .NET software's fast and efficient model-merging capabilities compare, merge and reconcile parallel changes as they are contributed. Improve productivity with powerful utilities to manage multiple changes within one model.

### Taking advantage of the Eclipse open and extensible platform

Because Rational Modeling Extension for Microsoft .NET is built on top of the open Eclipse 3.4 platform, you can more easily extend its features to meet specific project requirements through an ecosystem of third-party plug-ins. Eclipse is written in the Java language, which means you can outfit your team across both Microsoft Windows® and Linux® desktops.

### Integrating with other facets of the lifecycle

Integrated design and development improves traceability between code and its most immediately related artifacts. But complex software projects need traceability throughout the lifecycle. When requirements change, architects need to know what part of the architecture is affected. Such projects also require the management of change, as it impacts model files and other lifecycle artifacts. This all becomes quite complex and challenges even the best-managed change processes. Failing to address these issues introduces increased risk to overall project success.



The Rational Modeling Extension for Microsoft .NET application, plus the Rational Software Modeler product, is designed to help you integrate with other facets of the lifecycle. Requirements stored and managed in IBM Rational RequisitePro® software can be accessed, associated to corresponding modeling elements and synchronized with user-selectable rules. Users can generate reports highlighting traceability from requirements to design. Modeling files can be managed by the included IBM Rational ClearCase® LT tool, IBM's robust software configuration management product. Alternatively, the Rational Software Modeler and Rational Modeling Extension for Microsoft .NET products integrate with concurrent versions system (CVS) for users already committed to that tool. And the integration with process configurations provided by IBM Rational software or created with IBM Rational Method Composer software gives teams the ability to work through all of this with common, online and customizable process guidance.

The Rational Modeling Extension for Microsoft .NET software integrates with these and other aspects of the IBM Rational Team Unifying Platform™ solution, providing requirements management, traceability, source code control, automated documentation and other team management functions throughout the lifecycle. This helps reduce the risk associated with software development and helps make application development more predictable.

### For more information

To learn more about IBM Rational Modeling Extension for Microsoft .NET software, contact your IBM representative or IBM Business Partner, or visit:

[ibm.com//software/awdtools/  
modeling/ext](http://ibm.com/software/awdtools/modeling/ext)

© Copyright IBM Corporation 2008

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
October 2008  
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.