Volume
9

Architectural Design

Rintagi
Applications

User
Guide

robocodercorporation

Table of Contents

Chapter

1

Introduction

Rintagi is a renewable .NET open source code generator that that embraces late-change-no-penalty on the development and maintenance of enterprise business process management solutions.

The following design principles are adhered to when constructing Rintagi and its generated applications:

1. Secure;
2. Robust;
3. Scalable;
4. High Performance;
5. Low maintenance;
6. Consistent;

Rintagi jump-starts the development process by generating most, if not all, of the desired application. The code produced is consistent and can replicate the same look and feel throughout the application's user interface automatically. Complex and interactive user interfaces can be created with database-connected components including forms, reports, filters, and navigation features.

Rintagi Applications consist of three major components:

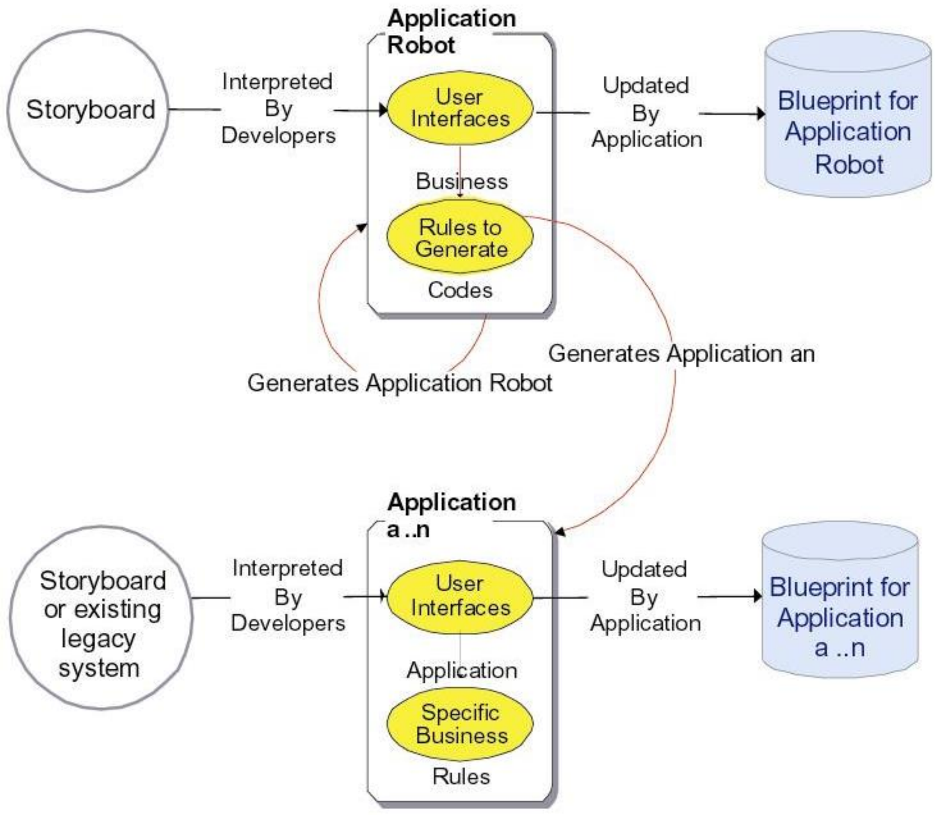
1. User interfaces that capture data and display information.
2. Business rules that transform data into valuable information.
3. Data storage and retrieval.

Rintagi's ability to self-generate allows customization for itself and the applications it creates. Your corporate standards and requirements are met easily and efficiently.

Although Rintagi can be modified as easily as the applications it generates, you can see from the above that steps 2 and 3 are extra steps to be taken in order for Rintagi to 'learn' and apply new features to self and other applications.

Although these extra steps require more time, if the same feature is needed in many other areas of the same or different applications, it becomes very economical for Rintagi to 'learn' and do the repetitive work. This only applies to features not included in the original Rintagi.

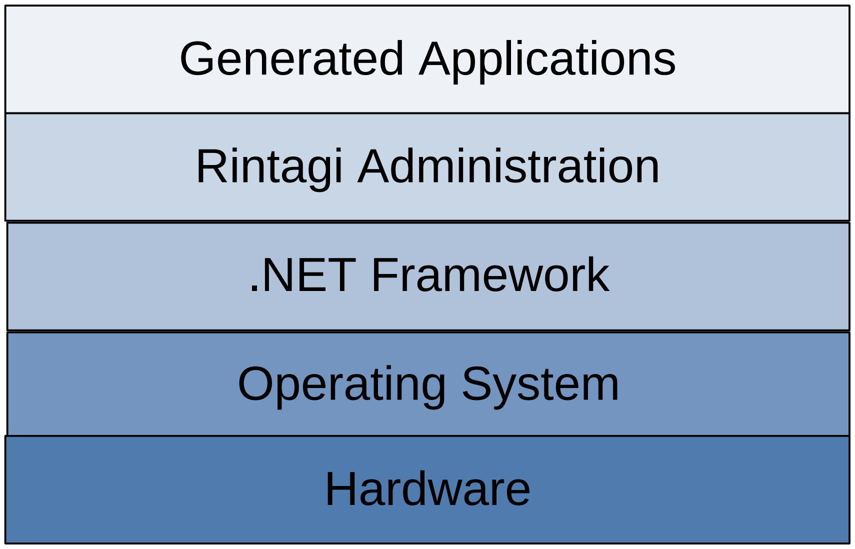
Rintagi allows software developers to intuitively translate user requirements into metadata and uses them to generate the desired application. Rintagi itself is generated by metadata captured over time as depicted in the following diagram. Unlike other code-generation technology, this self-generating ability has enabled Rintagi to be enhanced just as easily as the applications it generates. As a matter of fact Rintagi has 'grown' itself over the years. See the diagram below:



Chapter 2

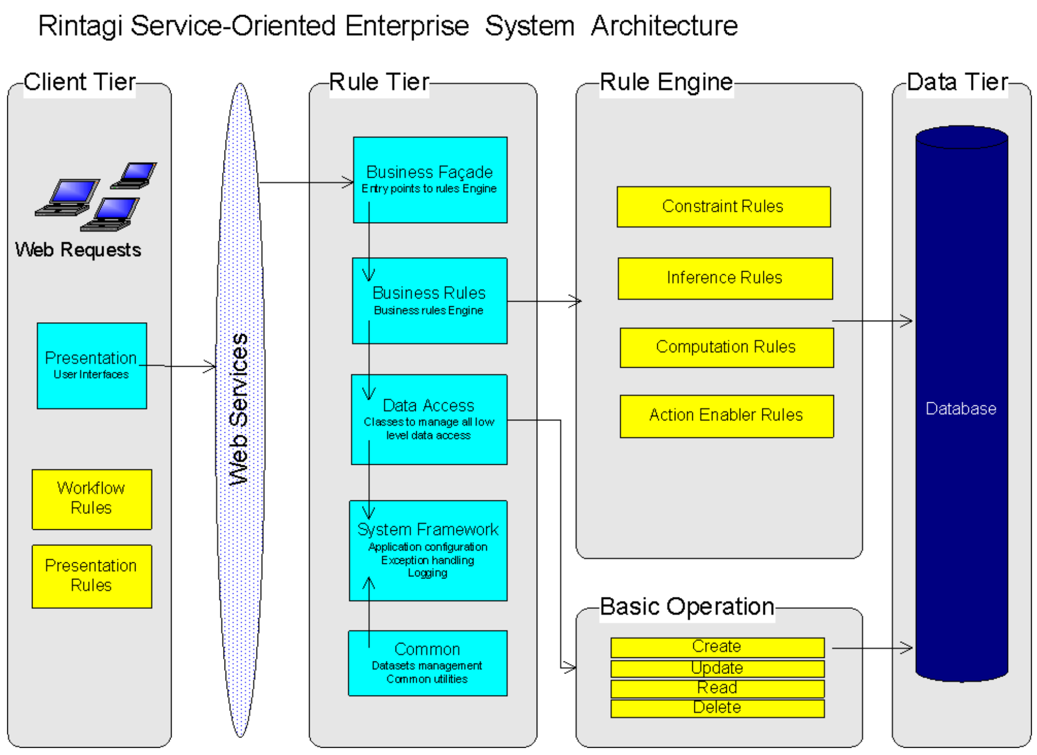
Architecture

Rintagi runs above Microsoft .NET framework. The generated code includes complex features like multi-table joins, multi-table access and distributed update on multi-servers. Rintagi generates native Microsoft .NET code, with ASPX pages, ASCX controls, SQL stored procedures and C# code. Applications generated by Rintagi are automatically multi-tenant, multi-project, and multi-system. Many companies and their projects plus their data can coexist on the same database.



Rintagi builds complete N-tier applications on a Service-Oriented Architecture. Developers can choose to directly call the programs in the Business Façade layer, or via the web services. It is just a setting in Web.config.

See the diagram below:



Enterprise-ready web-based service-oriented N-Tier applications

Rintagi generates an enterprise-ready, web-based N-Tier application on a service-oriented architecture. Rintagi generates this same N-Tier application for its enterprise applications including:

Web Pages	**Data Access Layer**
Presentation Layer	**Database Layer**
Business Rules Layer	**Framework**

Web Pages: The ASPX web pages generated by Rintagi are CSS 2.0/3.0 and XHTML 1.0/HTML5 compliant. All pages generated are browsers independent. Based on user-specific design, Rintagi generates applications user interface, including the web pages, dialog pages, menus, and search and navigation components. Rintagi generates complex display components including tab folders with data from multiple database tables and from multiple servers. These user interfaces can be further customized using any HTML editor including Microsoft Visual Studio.

Presentation Layer: Rintagi generates the code behind each of the ASPX pages and the user controls as modules from those pages.

Rintagi can generate a data grid comprised of data joined from multiple database tables and servers. A single update can simultaneously update multiple tables on different servers.

Reporting criteria is available to all reports, providing additional reporting flexibility to end-users. Last selected criteria are also cached for individual users. Users may designate each set of criteria with a name for the ease of running reports with the same criteria at a later time or at intervals.

Rintagi generates all of the user interface code necessary to support the application's web-based user interface, including filtering criteria and data validation logic for built-in data types. User-defined constraint rules can be set up to perform custom validation.

All web pages and presentation layers run on the client tier can be on a separate server from other tiers.

Business Layer: Adding custom application logic to applications generated by end-users is easy. Rintagi builds applications that are specifically designed to support all types of business rules: Constraints, Computation, Inference and Action Enablers. These rules can be applied instantly on the web page as client rules, applied upon post-back as web rules, and applied before or after data manipulation as server rules. Applications can be regenerated repeatedly without re-integrating these code extensions.

Business Rule Types

- Constraints
- Computation
- Inference
- Action Enablers

Data Access Layer: As part of data security, all data manipulation must to go through this layer and can only be called via a business façade layer. Only the business façade layer is exposed. The data access codes on this layer encapsulate the characteristics of the databases so that the calling functions can focus on business logic issues.

Each transaction is automatically bounded by Commit and Rollback and strictly adheres to the ACID test. ACID (atomicity, consistency, isolation, durability) is a set of properties that guarantee database transactions are processed reliably.

Atomicity

Atomicity requires that database modifications must follow an "all or nothing" rule. Each transaction is said to be atomic. If one part of the transaction fails, the entire transaction fails and the database state is left unchanged. It is critical that the database management system maintain the atomic nature of transactions in spite of any application, DBMS (Database Management System), operating system or hardware failure.

An atomic transfer cannot be subdivided and must be processed in its entirety or not at all. Atomicity means that users do not have to worry about the effect of incomplete transactions.

Consistency

The consistency property ensures that any transaction the database performs will take it from one consistent state to another. Consistency states that only valid data will be written to the database.

Application developers are responsible for ensuring application level consistency, over and above that offered by the DBMS. Thus, if a user withdraws funds from an account and the new balance is lower than the account's minimum balance threshold, as far as the DBMS is concerned, the database is in a consistent state even though this rule (unknown to the DBMS) has been violated.

Isolation

Isolation refers to the requirement that other operations cannot access data that has been modified during a transaction that has not yet completed. The question of isolation occurs in case of concurrent transactions (multiple transactions occurring at the same time). Each transaction must remain unaware of other concurrently executing transactions, except that one transaction may be forced to wait for the completion of another transaction that has modified data that the waiting transaction requires. If the isolation system does not exist, then the data could be put into an inconsistent state. This could happen, if one transaction is in the process of modifying data but has not yet completed, and then a second transaction reads and modifies that uncommitted data from the first transaction. If the first transaction fails and the second one succeeds, that violation of transactional isolation will cause data inconsistency. Rintagi uses optimistic locking mechanism so that each transaction does not need to wait for the others. If the transaction being edited has been changed by others, a warning message will be displayed to request the transaction to be processed again.

Durability

Durability is the ability of the DBMS to recover the committed transaction updates against any kind of system failure (hardware or software). Durability is the DBMS's guarantee that once the user has been notified of a transaction's success the transaction will not be lost, the transaction's data changes will survive system failure, and that all integrity constraints have been satisfied, so the DBMS won't need to reverse the transaction. Many DBMSs implement durability by writing transactions into a transaction log that can be reprocessed to recreate the system state right before any later failure. A transaction is deemed committed only after it is entered in the log.

Durability does not imply a permanent state of the database. A subsequent transaction may modify data changed by a prior transaction without violating the durability principle.

Database Layer: SQL stored procedures and queries needed for storing, retrieving, and filtering data are contained in the database. Rintagi will even generate complex queries including multi-table joins with one-to-many and many-to-many relationships. Most of the generated SQL is packaged as a set of database stored procedures in order to provide the best execution and performance with the fewest round trips to the database. Thus network traffic is significantly minimized.

Rintagi automatically constructs the SQL queries required for each screen, import wizard and report. Rintagi also generates the database access logic that executes the SQL queries, including all data storage management.

Although this database layer can be a separate server like the previous two layers, all three tiers can run on a single server with superb performances because of the optimized codes which become well-tuned over time.

Framework: Rintagi generates web-based applications that run on the Microsoft .NET framework, making applications easier to integrate, deploy, and manage. The industry-standard codes generated can be modified easily using Microsoft Visual Studio. Development time is dramatically reduced leaving the developers to focus on customization and integration. Reports can be generated to run on Crystal Report engine or SQL Reporting Service.

Chapter
3

Identity & Access Management

Authentication

When transmitted on a network, digital identity is represented by some kind of token. In Rintagi, each user is identified by a username / password token. The username is matched against the username stored in the database while password is hashed to match the hashed password stored in the database. If both of these match, the user is identified as a valid user of the application.

Authorization

Each user is designated one or more user-defined user groups and a specific human language with a country-specific locale. Each user may belong to one or more user groups.

Each user group can have different roles (read, write, export, print, all or selected data) within multiple companies, projects and systems.

Each role can be defined and overridden further by rows and columns.

Each column can further be designated as visible/invisible, read-only/editable and exportable/non-exportable for each individual user, group, company or project. Attributes such as tool tips, labels, and error messages can be customized too.

Each menu item can also be granted /deny access to selected user, group, company or project.

Modifying User Preference

User Preference should be used to change the functionality or look-and-feel of overall design theme for selected users. The colors, fonts, positioning, and other stylistic elements specified by the above style sheet can be overridden by the style sheet text box in this User Preference area. Splash image, menu options, shortcuts settings, quick launch, etc. can also be specified.

Each user-defined preference can be assigned to individual users, user groups, companies, projects, or the entire system.

Integrating Other Applications

Applications generated by Rintagi can be easily integrated with other existing applications and web pages. Rintagi-generated applications can operate as separate, independent applications because of the state-less nature of web programming. One application may call into another via the URL mechanism inherent in all web applications.

For example, Rintagi-generated applications can call any web page from within its menu.

Other applications can call any Rintagi-generated programs via a standard html anchor.

Integration can be done via web rule, as well, where other web pages and web services of other applications can be called and the results integrated into Rintagi-generated applications.

Session Management

Session management is the process of keeping track of a user's activity across sessions of interaction with Rintagi. Session cookies have the 'secure' directive set to ensure that session cookies aren't sent over an unsecure channel.

Typical session management tasks in a desktop environment might include keeping track of which applications are open and which documents each application has opened, so that the same state can be restored when the user logs out and logs in later. For Rintagi, session management requirs the user to re-login if the session has expired (i.e., a certain time limit has passed without user activity). It is also used to store information on the server-side between HTTP requests so that a call from one system to another system can occur without losing its identity.

Each screen or report maintains its own session so that a user can open multiple tabs on a browser to access multiple screens or reports on different systems of the same application.

Chapter

4

Cryptography

Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription. Some experts argue that cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet.

Within the context of any application-to-application communication, there are some specific security requirements, including:

Authentication: The process of proving one's identity. This has been discussed in Chapter 1.

Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver.

Integrity: Assuring the receiver that the received message has not been altered in any way from the original.

Non-repudiation: A mechanism to prove that the sender really sent this message.

Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication.

There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions. Only the first and last are being used in Rintagi and Rintagi generated application. They are being described below.

Secret key (or symmetric) cryptography uses a single key for both encryption and decryption. Rintagi uses that to protect the password of SQL server login. The class to encrypt is described below:

The secret key cryptography algorithm that we use is Triple-DES (3DES). It is a variant of DES that employs up to three 56-bit keys and makes three encryption/decryption passes over the block. 3DES is the recommended replacement to DES by many cryptography professions.

Data Encryption Standard (DES) is the most common secret key cryptography scheme used today. DES was designed by IBM in the 1970s and is widely adopted for commercial and unclassified government applications. DES is a block-cipher employing a 56-bit key that operates on 64-bit blocks. DES has a complex set of rules and transformations that were designed specifically to yield fast hardware implementations and slow software implementations, although this latter point is becoming less significant today since the speed of computer processors is several orders of magnitude faster today than twenty years ago.

DES is defined in American National Standard X3.92 and three Federal Information Processing Standards (FIPS):

We also use a **hash function** as a mathematical transformation to irreversibly encrypt each user's password and stored them securely in the SQL database. Upon login request, the input password will be hashed before submitting to a stored procedure in the SQL server for validation. The class to perform this one-way hash is described below:

The secure hash algorithm that we use is SHA-1, an algorithm for Secure Hash Standard, which produces a 160-bit hash value and was originally published as FIPS (Federal Information Processing Standards) 180-1. It uses no key but instead a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered.

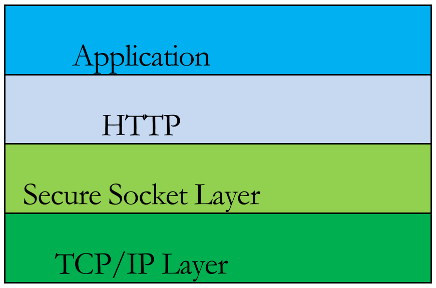
Chapter

5

Protocols

Rintagi uses Hypertext Transfer Protocol Secure (HTTPS) for authentication and to securely transmit data. This is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server. HTTPS connections are often used for payment transactions on the World Wide Web and for sensitive transactions in corporate information systems.

HTTP operates at the Application layer. The security protocol operates at a lower network layer, encrypting an HTTP message prior to transmission and decrypting a message upon arrival. Strictly speaking, HTTPS is not a separate protocol, but refers to use of ordinary HTTP over an encrypted SSL/TLS connection.



HTTPS runs above the Transmission Control Protocol/Internet Protocol (TCP/IP).

HTTPS URLs begin with "https://" and use port 443 by default, where HTTP URLs begin with "http://" and use port 80 by default.

HTTP is unsecured and is subject to man-in-the-middle and eavesdropping attacks, which can let attackers gain access to website accounts and sensitive information. HTTPS is designed to withstand such attacks and is considered secure against such attacks.

These capabilities of HTTPS address fundamental concerns about communication over the Internet and other TCP/IP networks:

SSL server authentication allows a user to confirm a server's identity. Most internet browsers today are SSL-enabled and can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs.

Example certificate authorities are VeriSign, RapidSSL, etc. This confirmation might be important if the user is sending sensitive information over the network and wants to check the receiving server's identity.

SSL client authentication allows a server to confirm a user's identity. Using the same techniques as those used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server is sending confidential information to a user and wants to check the recipient's identity.

An encrypted SSL connection requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties to any private transaction. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering.

Weak SSL Ciphers are disabled. The only SSL protocols that are allowed are SSLv3 and TSLv1.

Chapter

6

Up-to-date Documentation

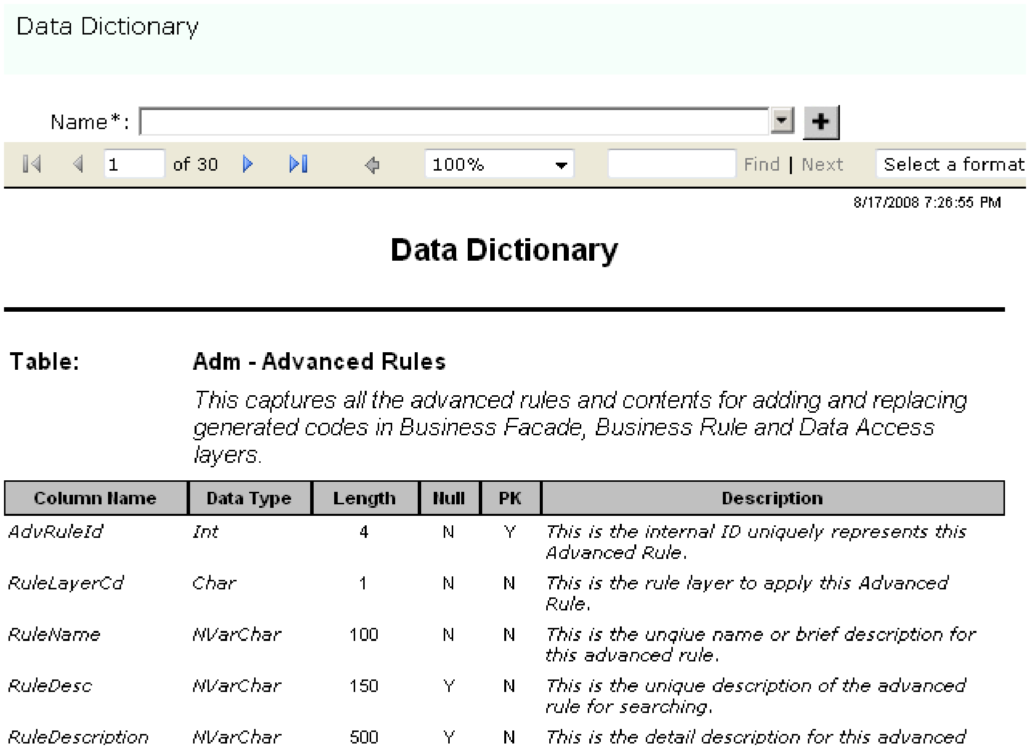
Rintagi automatically provides and updates four kinds of documentations that are usually difficult to keep up-to-date when applications are being enhanced frequently.

Data Dictionary	**Program Listing**
Business Rules Listing	**Message Listing**

Data Dictionary

Each application generated by Rintagi keeps its own data model as meta-data or a blueprint. If a description has not been entered for each data column, the tool tips are copied to describe the data columns as they are entered. These descriptions can be modified from time to time to reflect the current definitions. As the application grows and changes, each additional data table or column is tracked and updated to the data dictionary.

Data Dictionary Screen



Data Dictionary

Name*:

1 of 30 | 100% | Find | Next | Select a format

8/17/2008 7:26:55 PM

Data Dictionary

Table: **Adm - Advanced Rules**

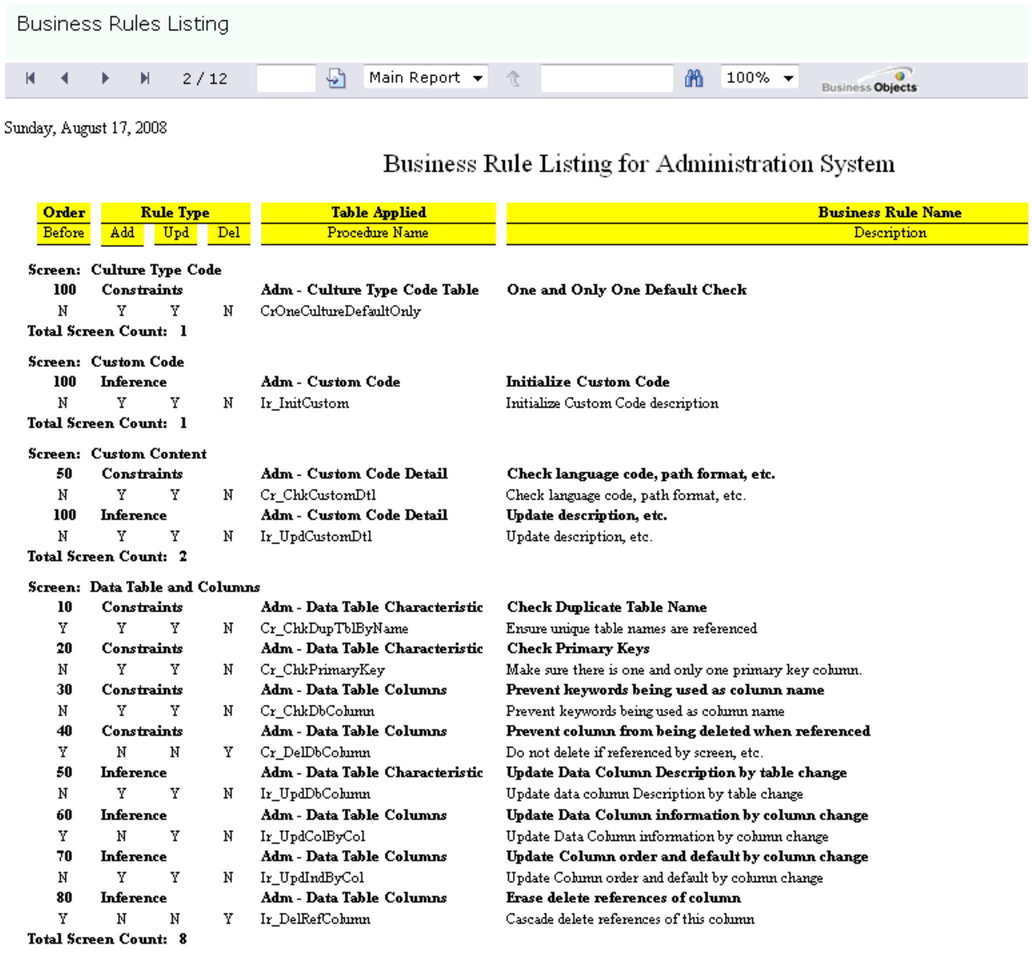
This captures all the advanced rules and contents for adding and replacing generated codes in Business Facade, Business Rule and Data Access layers.

Column Name	Data Type	Length	Null	PK	Description
AdvRuleId	*Int*	4	N	Y	*This is the internal ID uniquely represents this Advanced Rule.*
RuleLayerCd	*Char*	1	N	N	*This is the rule layer to apply this Advanced Rule.*
RuleName	*NVarChar*	100	N	N	*This is the unqiue name or brief description for this advanced rule.*
RuleDesc	*NVarChar*	150	Y	N	*This is the unique description of the advanced rule for searching.*
RuleDescription	*NVarChar*	500	Y	N	*This is the detail description for this advanced*

Business Rule Listing

Business rules will be covered in later volumes of advanced topics. For sophisticated developers, many different kinds and different types of business rules may apply to Rintagi-generated applications. In the traditional programming world, these business rules are usually embedded or hidden in the programs and are very difficult to extract. Thanks to the blueprint structure used by Rintagi, not only can business rules be extracted easily for modification, they can be listed anytime for reference and communication.

Business Rules Listing Report



Business Rules Listing

2 / 12 Main Report 100%

Sunday, August 17, 2008

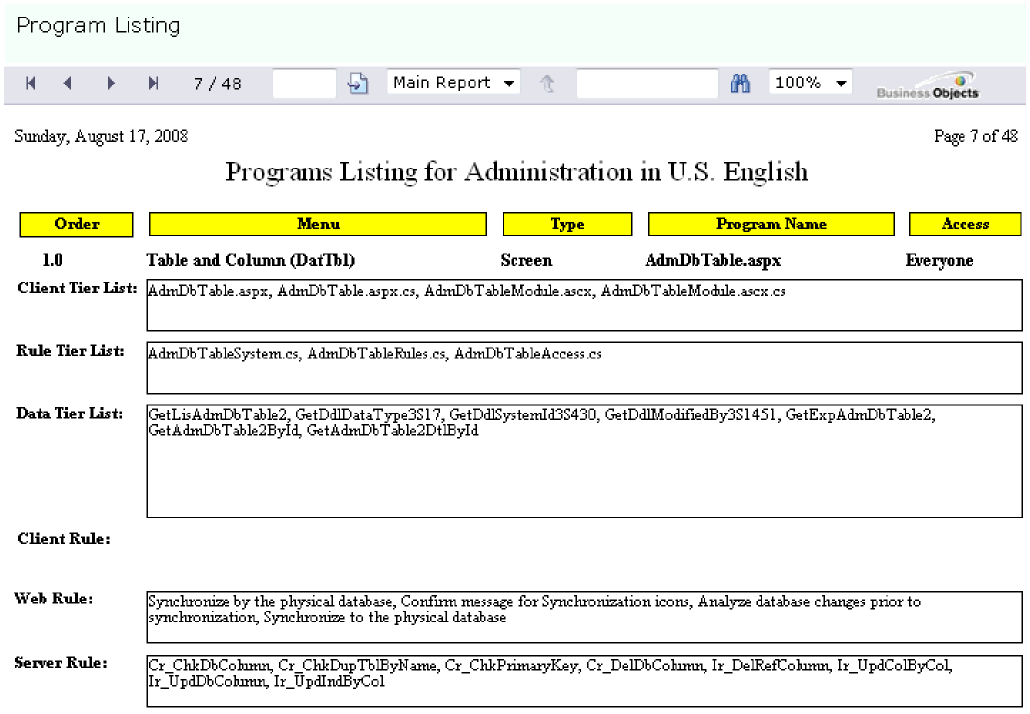
Business Rule Listing for Administration System

Order	Rule Type			Table Applied	Business Rule Name
Before	Add	Upd	Del	Procedure Name	Description
Screen: Culture Type Code					
100	**Constraints**			**Adm - Culture Type Code Table**	**One and Only One Default Check**
N	Y	Y	N	CrOneCultureDefaultOnly	
Total Screen Count: 1					
Screen: Custom Code					
100	**Inference**			**Adm - Custom Code**	**Initialize Custom Code**
N	Y	Y	N	Ir_InitCustom	Initialize Custom Code description
Total Screen Count: 1					
Screen: Custom Content					
50	**Constraints**			**Adm - Custom Code Detail**	**Check language code, path format, etc.**
N	Y	Y	N	Cr_ChkCustomDtl	Check language code, path format, etc.
100	**Inference**			**Adm - Custom Code Detail**	**Update description, etc.**
N	Y	Y	N	Ir_UpdCustomDtl	Update description, etc.
Total Screen Count: 2					
Screen: Data Table and Columns					
10	**Constraints**			**Adm - Data Table Characteristic**	**Check Duplicate Table Name**
Y	Y	Y	N	Cr_ChkDupTblByName	Ensure unique table names are referenced
20	**Constraints**			**Adm - Data Table Characteristic**	**Check Primary Keys**
N	Y	Y	N	Cr_ChkPrimaryKey	Make sure there is one and only one primary key column.
30	**Constraints**			**Adm - Data Table Columns**	**Prevent keywords being used as column name**
N	Y	Y	N	Cr_ChkDbColumn	Prevent keywords being used as column name
40	**Constraints**			**Adm - Data Table Columns**	**Prevent column from being deleted when referenced**
Y	N	N	Y	Cr_DelDbColumn	Do not delete if referenced by screen, etc.
50	**Inference**			**Adm - Data Table Characteristic**	**Update Data Column Description by table change**
N	Y	Y	N	Ir_UpdDbColumn	Update data column Description by table change
60	**Inference**			**Adm - Data Table Columns**	**Update Data Column information by column change**
Y	N	Y	N	Ir_UpdColByCol	Update Data Column information by column change
70	**Inference**			**Adm - Data Table Columns**	**Update Column order and default by column change**
N	Y	Y	N	Ir_UpdIndByCol	Update Column order and default by column change
80	**Inference**			**Adm - Data Table Columns**	**Erase delete references of column**
Y	N	N	Y	Ir_DelRefColumn	Cascade delete references of this column
Total Screen Count: 8					

Program Listing

In the past the tracking of computer programs was an arduous process. Not so with Rintagi as most programs are being generated and maintained constantly by this renewable code generator. Custom programs can also be included in an up-to-date listing of all of these programs if they are declared to Rintagi.

Program Listing Report



Sunday, August 17, 2008 | Page 7 of 48

Programs Listing for Administration in U.S. English

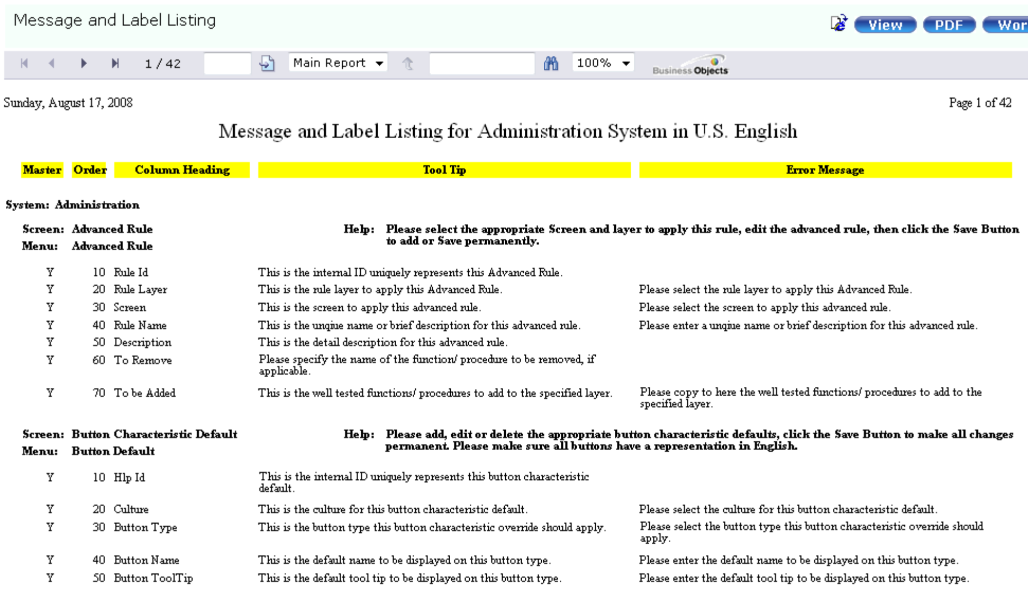
Order	Menu	Type	Program Name	Access
1.0	Table and Column (DatTbl)	Screen	AdmDbTable.aspx	Everyone

Client Tier List:	AdmDbTable.aspx, AdmDbTable.aspx.cs, AdmDbTableModule.ascx, AdmDbTableModule.ascx.cs
Rule Tier List:	AdmDbTableSystem.cs, AdmDbTableRules.cs, AdmDbTableAccess.cs
Data Tier List:	GetLisAdmDbTable2, GetDdlDataType3S17, GetDdlSystemId3S430, GetDdlModifiedBy3S1451, GetExpAdmDbTable2, GetAdmDbTable2ById, GetAdmDbTable2DtlById
Client Rule:	
Web Rule:	Synchronize by the physical database, Confirm message for Synchronization icons, Analyze database changes prior to synchronization, Synchronize to the physical database
Server Rule:	Cr_ChkDbColumn, Cr_ChkDupTblByName, Cr_ChkPrimaryKey, Cr_DelDbColumn, Ir_DelRefColumn, Ir_UpdColByCol, Ir_UpdDbColumn, Ir_UpdIndByCol

Message Listing

By selecting a desired human language, an up-to-date listing of all of the screen online help messages, labels, tool tips and error messages are immediately available.

Message Listing Report



Sunday, August 17, 2008

Page 1 of 42

Message and Label Listing for Administration System in U.S. English

System: Administration

Screen: Advanced Rule
Menu: Advanced Rule

Help: Please select the appropriate Screen and layer to apply this rule, edit the advanced rule, then click the Save Button to add or Save permanently.

Master	Order	Column Heading	Tool Tip	Error Message
Y	10	Rule Id	This is the internal ID uniquely represents this Advanced Rule.	
Y	20	Rule Layer	This is the rule layer to apply this Advanced Rule.	Please select the rule layer to apply this Advanced Rule.
Y	30	Screen	This is the screen to apply this advanced rule.	Please select the screen to apply this advanced rule.
Y	40	Rule Name	This is the unqiue name or brief description for this advanced rule.	Please enter a unqiue name or brief description for this advanced rule.
Y	50	Description	This is the detail description for this advanced rule.	
Y	60	To Remove	Please specify the name of the function/ procedure to be removed, if applicable.	
Y	70	To be Added	This is the well tested functions/ procedures to add to the specified layer.	Please copy to here the well tested functions/ procedures to add to the specified layer.

Screen: Button Characteristic Default
Menu: Button Default

Help: Please add, edit or delete the appropriate button characteristic defaults, click the Save Button to make all changes permanent. Please make sure all buttons have a representation in English.

Master	Order	Column Heading	Tool Tip	Error Message
Y	10	Hlp Id	This is the internal ID uniquely represents this button characteristic default.	
Y	20	Culture	This is the culture for this button characteristic default.	Please select the culture for this button characteristic default.
Y	30	Button Type	This is the button type this button characteristic override should apply.	Please select the button type this button characteristic override should apply.
Y	40	Button Name	This is the default name to be displayed on this button type.	Please enter the default name to be displayed on this button type.
Y	50	Button ToolTip	This is the default tool tip to be displayed on this button type.	Please enter the default tool tip to be displayed on this button type.