



P r o f e s s i o n a l E x p e r t i s e D i s t i l l e d

Microsoft Dynamics AX 2012 R3 Security

A quick guide to planning, designing, and debugging operational-level security for Microsoft Dynamics AX 2012 R3 implementations

Ahmed Mohamed Rafik Moustafa

[PACKT] enterprise 
PUBLISHING professional expertise distilled

www.allitebooks.com

Microsoft Dynamics AX 2012 R3 Security

A quick guide to planning, designing, and debugging operational-level security for Microsoft Dynamics AX 2012 R3 implementations

Ahmed Mohamed Rafik Moustafa



BIRMINGHAM - MUMBAI

Microsoft Dynamics AX 2012 R3 Security

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: June 2015

Production reference: 1150615

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78217-553-7

www.packtpub.com

Credits

Author

Ahmed Mohamed Rafik Moustafa

Project Coordinator

Mary Alex

Reviewers

Abd El-Rahman Magdy Ahmed

Parag Gunwant Chapre

Muhammad Anas Khan

Isaac .W. Namukoa

Amy Walsh

Proofreader

Safis Editing

Indexer

Hemangini Bari

Commissioning Editor

Usha Iyer

Production Coordinator

Komal Ramchandani

Acquisition Editor

Vinay Argekar

Cover Work

Komal Ramchandani

Content Development Editor

Rohit Singh

Technical Editor

Mrunal M. Chavan

Copy Editors

Aditya Nair

Stuti Srivastava

About the Author

Ahmed Mohamed Rafik Moustafa is a Dynamics AX solution architect and a Dynamics AX evangelist. In November 2012, he published his first book, *Microsoft Dynamics AX 2012 Security How-To*, Packt Publishing, and he was the first Egyptian and Arabian person to have authored a book on Microsoft Dynamics ERP products. He has been ranked and listed as one of the Top 100 Most Influential People by *DynamicsWorld*, United Kingdom. He is also a columnist at *MSDynamicsWorld*, a media publishing corporate in the UK, and has been recognized as an official blogger by the Microsoft Dynamics Community.

Ahmed's professional career, spanning more than 10 years, has combined his expertise in business management and information technology in different industries, such as the retail, manufacturing, medical, and trading industries. He has led various implementations in diverse ERP modules over the Middle East in different countries to meet and exceed challenging business needs. He has carried out multiple project implementations of Microsoft Dynamics GP, Microsoft Dynamics AX, and Microsoft Dynamics Retail Management System (RMS) in diverse positions, such as project manager, team leader, and senior consultant.

In addition to his knowledge and experience of Enterprise Resource planning (ERP) systems, he is always keen to raise awareness about information system security. He has been recognized as an Information Security Awareness Expert by ASK PC, the largest Arabic IT community, in association with Information System Security Association (ISSA's Egypt chapter). Also, he is listed on ASK PC's Wall of Honor, as he published his first paper on accounting information system fraud and computer crimes on Culminis/GITCA, sponsored by Microsoft. Furthermore, he has so far published two paper books on Microsoft Dynamics AX security and plans to publish more books and articles.

In addition to these achievements, Ahmed holds these certifications: Microsoft Certified Master Great Plains (GP), Microsoft Certified Business Management Solutions Professional (GP), and Microsoft Certified Information Technology Specialist (MCITP) on Microsoft Dynamics AX products. He has also been a Microsoft Certified Trainer (MCT) for 7 years.

He is the founder of the Dynamics AX camp user group, sponsored by Microsoft Technical Communities, O'Reilly Media Corporate, Pluralsight Developer Training, and EMC Community Network. The Dynamics AX camp user group aims to share knowledge, experience, news, articles, and books in the ERP field, specifically in relation to Microsoft Dynamics AX products.

In 2013, Ahmed committed himself to helping students and graduates by providing free training seminars introducing Microsoft Dynamics ERP solutions and teaching them how to build a career in Microsoft Dynamics AX ERP products. He is considered a career coach expert and is also a keynote/guest speaker at different universities in Egypt, such as the American University in Cairo (AUC), the German University in Cairo (GUC), and the British University in Egypt (BUE).

Furthermore, he is using the science of coaching to leverage the success rate of ERP project implementations to lead the change that happens when organizations adopt the ERP solution, because he believes that success in ERP projects first depends on the people who use the ERP system first and then on everyone involved with implementing the enterprise system.

In addition to his exceptional communication skills, Ahmed has a special talent for bringing out the best in others, especially his team members, by instilling a high level of motivation in them. When he isn't focusing on his career, he enjoys playing his favorite sports, such as football, swimming, and squash. He is also a good piano player, and, as you can see, he tries to maintain a balance in his life through his diverse interests and passions because he enjoys living life with joy and passion.

About the Reviewers

Abd El-Rahman Magdy Ahmed is working as a senior ERP functional consultant at Dynamics AX at Columbus Global.

He is a Microsoft Certified Axapta Functional Consultant with more than 5 years of IT experience and expertise in MBS-Axapta implementations, functional analysis, Fit and Gap Analysis, Functional Design Document (FDD), customization with regards to designing and development, testing, and debugging. He has experience of the following:

- Supply chain functional implementation on Axapta 2009 and 2012
- Business Process Reengineering (BPR)
- Business consulting, implementations, and customer support
- ERP implementation skills (Dynamics AX) supply chain cycles: inventory, sales operations, procurement, accounts receivable, accounts payable, logistics, quality systems, shipping systems, and quarantine systems

His specialties are Microsoft Dynamics AX 2009 and 2012 (financial – trade and logistics, master planning, budgeting, and fixed assets), preparing solution designs on AX, supply chain management, system analysis, and design, Fit and Gap Analysis, data templates preparation, migration to AX, key, and end user training of trade and logistic modules on AX.

He has also worked on *Microsoft Dynamics AX 2012 Security How-To, Packt Publishing*.

I thank the author of this book, Ahmed Rafik, for writing this book.

Parag Gunwant Chapre is currently working with Tieto Software Technologies Limited as a senior technical consultant. He completed his BE in CSE at Nagpur University in 2008 with a first division. He has over 6 years of experience in MS Dynamics AX 2009/2012 and ASP.NET/C#.NET, MS CRM 2011, SSRS, Dynamics Connector, and AIF.

He has worked with top MS Dynamics AX companies, such as Systems Advisers Group (SAGlobal); Tectura Corporation, Noida; and Tata Consultancy Services, Pune. He has worked on different versions of Axapta such as AX 2009, AX 2012 R2, and R3.

His work experience includes Windows and web applications, SSRS development, Microsoft Dynamics AX 2009/2012, Application Integration Framework (AIF), Microsoft Dynamics Connector, and MS Dynamics CRM.

He has certifications in Windows and web applications (.NET), installation and configuration, introduction development, and MorphX solution development in MS Dynamics AX 2009/2012.

He has received appreciation from various clients for developing the SSRS report and for MS dynamics AX's integration with MS Dynamics CRM. He has worked as a technical reviewer on *Microsoft Dynamics AX 2012 R3 Cookbook*, Packt Publishing.

I would like to thank my parents and my sister for their continuous support, guidance, and encouragement.

Special thanks to the Packt Publishing team, who provided me with a chance to review this book.

Muhammad Anas Khan is a Microsoft Certified Professional, working as a technical consultant for Microsoft Dynamics AX at Mazik Global, where he is responsible for delivering consultancy on Dynamics AX implementation projects. His technical expertise includes Application Integration Framework (AIF), forms, SSRS and SSAS reporting, the Batch framework, role-based security, workflow development, and Enterprise Portal development.

He has more than 6 years of experience in the software industry, where he held various engineering positions in developing global enterprise systems. His career vision is to frame the right problems and find efficient solutions that deliver value to customers, partners, and shareholders. He has a master's degree in computer science from IBA University and lives with his family in Karachi.

He has also contributed to *Microsoft Dynamics AX 2012 R3 Reporting Cookbook*, Packt Publishing, as a technical reviewer.

You can find him on LinkedIn at <https://www.linkedin.com/in/muhammadanaskhan> and read his Dynamics AX blog at <https://dynamicsaxinsight.wordpress.com/>.

I would like to thank my family for their continuous support, especially my mentors for guiding me well throughout my career.

Special thanks to Mary Alex and the whole Packt Publishing team for giving me the opportunity to review this book.

Isaac .W. Namukoa has over 5 years of consulting experience and has played a variety of roles, including developer, lead developer, design authority, and technical architect, in the Dynamics AX and Microsoft technology. He lives in Nairobi, Kenya, and works as a business analyst for UAP Holdings, an investment, retirement, and insurance group that operates mainly in East Africa and plans to be a pan-African insurance company.

I would like to thank my family and friends, who have always been supportive and have shown true unconditional love and patience through my entire career. I would also like to give a shout-out to my fiancée, Joyce, for the overwhelming support she accorded me through the review.

Amy Walsh is a principal consultant at I.B.I.S., Inc., located in Atlanta, GA, and is a Microsoft Certified Business Management Solutions Professional. She is a graduate of Georgia Military College and Mercer University with dual majors in accounting and finance. Prior to joining I.B.I.S., Inc., she worked in both the public and private accounting industry. This experience includes over 15 years of working in management, financial accounting, audit, and tax with both domestic and international companies that range from start-ups to established global B2B and B2C companies.

Over the last 9 years, Amy has been focusing on Microsoft business solutions, ERP implementations, SaaS, business intelligence, reporting, business process improvement, and accounting. Her experience in these various industries has been a cornerstone in helping decision makers understand and transition into new technology that keeps businesses ahead of the competition. Her goal is to continue helping businesses succeed in their endeavors, which can be accomplished by finding the right ERP system and reporting tools.

She has worked on *Microsoft Dynamics GP 2013 Reporting - Second Edition*, Packt Publishing.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Instant updates on new Packt books

Get notified! Find out when new books are published by following [@PacktEnterprise](https://twitter.com/PacktEnterprise) on Twitter or the *Packt Enterprise* Facebook page.

This book is dedicated to the memory of my uncle, Dr. Ahmed Hegazy, as he has supported me my whole life as a father, friend, and mentor. He encouraged me to write my first book and guided me to write this second one. He is the main reason for all my life achievements. Words can't describe him, but I believe that he is reading this dedication and asking for your prayers. This book is also dedicated to my mom because without her sacrifice and guidance, I would not be the person I am today. Thanks, mom.

Table of Contents

Preface	v
Chapter 1: MorphX Security System	1
Introducing the MorphX development tool	3
Application Object Tree	5
The X++ code editor	6
Compiler	8
Debugger	9
Projects	10
The property sheet	11
The cross-reference tool	13
The Find tool	13
The table browser tool	14
The best practice tool	15
The reverse engineering tool	16
Developing a security artifact using AOT	17
Setting permissions for a form	18
Assigning permissions to privileges	19
Validating and testing a security privilege	22
Applying a configuration key	22
Summary	24

Chapter 2: Security Coding	25
The fundamentals of security coding using X++	25
Using Code Access Security	26
Securing an API on the AOS	28
Security debugging	30
Installing the debugger	30
Enabling the debugger	31
Adding users to the Debugging User local group	33
The debugger user interface	36
Debugger shortcut keys	37
Security for the display and edit methods	38
The Table Permissions Framework	41
Summary	45
Chapter 3: Developing Extensible Data Security	47
The main concepts of XDS policies	48
Designing and developing the XDS policy	48
Creating the policy	49
Adding constrained tables and views	51
Setting the XDS policy context	52
Debugging XDS policies	53
Summary	55
Chapter 4: Extending the Organization Model	57
The organizational model framework	58
Organization hierarchies	59
The organizational model scenarios	61
Integration with other frameworks' application modules	61
Custom modeling scenarios	62
Extending the organizational model	62
Creating a custom type of operating unit	62
Creating a new base enum value	63
Creating a view	63
Creating a menu item	64
Extending the hierarchy designer	65
Summary	65

Chapter 5: Enterprise Portal Security	67
The architecture of Enterprise Portal	67
Web parts	68
AOT elements	68
Datasets	69
Controls	69
Security in Enterprise Portal	70
Securing web elements	70
Record context and encryption	72
Data access security	72
Report access security	74
Assigning a user to the DynamicsAXBrower role	74
Granting a user access permission to view reports	75
Summary	75
Index	77

Preface

Welcome to *Microsoft Dynamics AX 2012 R3 Security*, where we take you on a journey, starting from the security development concepts that use Microsoft Dynamics AX 2012 R3 and ending with practical steps to make the necessary security setups, illustrated with snapshots and figures that will guide you through developing your environmental security system.

What this book covers

Chapter 1, MorphX Security System, gives you a solid introduction to MorphX development tools. You will be able to use each development feature in a smooth and fast way.

Chapter 2, Security Coding, gives you the ability to use the code access security to secure your environment and also teaches you how to debug security coding.

Chapter 3, Developing Extensible Data Security, enables you to secure your sensitive data using the extensible data security features by designing and developing XDS policies.

Chapter 4, Extending the Organization Model, helps you understand the types of organizations and the basic categories of operational units and how to use them.

Chapter 5, Enterprise Portal Security, provides you with an understanding of the architecture of the Enterprise Portal in Microsoft Dynamics AX 2012, and ensures that you are able to secure web parts and elements.

What you need for this book

You will need to properly understand every section first and then practice the examples covered in this book by moving step by step. You also need to run the Microsoft Dynamics AX 2012 R3 virtual machine to move with every step, so you will not have to worry about missing anything.

Who this book is for

If you are an AX implementer, an AX developer, a network administrator, or an IT person charged with configuring Dynamics AX 2012 R3 in your company, then this book is for you. The book assumes that you are familiar with basic security terminologies. Also, this book provides you with a good overview and covers details that make it suitable for beginners and intermediate and advanced readers.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Other metadata, such as `LayerId`, can be debugged if needed."



A block of code is set as follows:



```
SELECT [PRIMARYTABLEAOTNAME], [QUERYOBJECTAOTNAME],
       [CONSTRAINEDTABLE], [MODELEDQUERYDEBUGINFO],
       [CONTEXTTYPE], [CONTEXTSTRING],
       [ISENABLED], [ISMODELED]
FROM [AXDBDEV].[dbo].[ModelSecPolRuntimeEx]
```

Any command-line input or output is written as follows:

```
%windir%\system32\cmd.exe /c runas /savecred
/user:mywindowsdomain\axtest3 "C:\Program Files
(x86)\Microsoft Dynamics AX\6.0\Client\Bin\Ax32.exe"
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Right-click on the **Constrained Tables** node."

 Warnings or important notes appear in a box like this. 

 Tips and tricks appear like this. 

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail feedback@packtpub.com, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

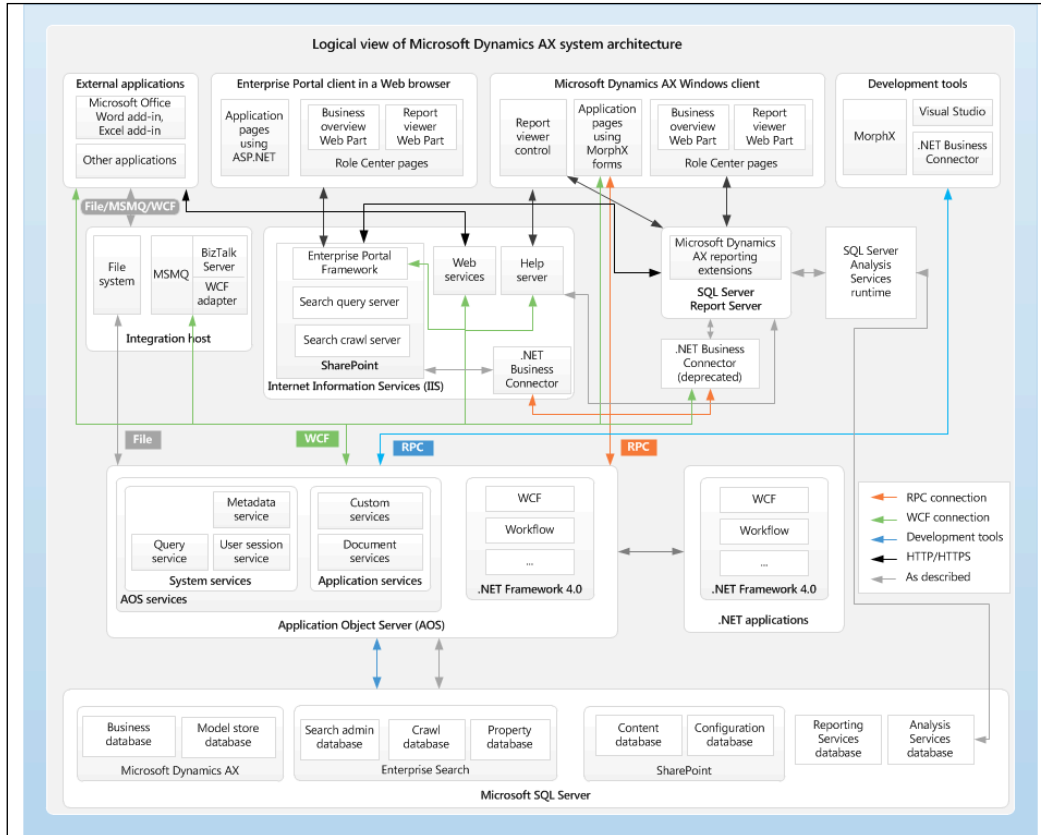
1

MorphX Security System

The security tool in Microsoft Dynamics AX Security 2012 R3 is built to help organizations create and manage secure ERP implementations. Illustrated with MorphX, X++ coding, snapshots, and figures, this chapter intends to provide you with the basics to transform the knowledge to seamlessly implement security configurations into practical steps that are needed to develop an efficient security environment. In this chapter, we will broadly look at:

- Introducing the MorphX development tool
- Developing a security artifact using AOT
- Validating and testing a security privilege
- Applying a configuration key

By going through this chapter, you will briefly know the fundamentals and security concepts in the Microsoft Dynamics AX product. The security architecture in the Microsoft Dynamics AX product consists of the infrastructure security and the application security (<https://technet.microsoft.com>):



The logical view of Microsoft Dynamics AX system architecture

The different blocks in this architecture are as follows:

- **Infrastructure security:** The Microsoft Dynamics AX infrastructure is based on the following features:
 - Active Directory services
 - Integrated windows authentication
 - Computer networking
 - Secured servers' machine

- **Application security:** Application security has the same features as those listed for infrastructure security and includes the following additional features:
 - Active Directory users added to Microsoft Dynamics AX and granted access to use the application
 - Domains that are groups of the company accounts in Dynamics AX
 - Record-level security to restrict or permit users to access specific fields and tables
 - Security keys that allow users to access specific forms, reports, or menus

By focusing on application security, we are going to break it down into the following points that should be considered when securing the Dynamics AX server:

- **Application file server:** The application files should be restricted to the application object server domain account
- **Database server:** The database server should be secured using the recommended SQL server security solution
- **Application Object Server (AOS):** The AOS should be restricted to the log directory for only the AOS account directory and the administrator
- **Enterprise Portal:** Securing the Enterprise Portal starts with Microsoft **Internet Information Services (IIS)** using the **Secure Sockets Layer (SSL)** and another built-in feature called **Business Connector**

This is the high-level security architecture of the Microsoft Dynamics AX product that you should gain knowledge about before proceeding to the security features that enable administrators, technical consultants, and programmers to secure the application. We are going step by step to deliver proper experience and practices through this chapter.


Introducing the MorphX development tool

Microsoft Dynamics AX includes a set of tools, and one of the most powerful is the MorphX development tool, which you can use to build and modify Microsoft Dynamics AX business applications. With the MorphX tool, you can create, view, modify, and delete the application model elements that contain metadata, structure, properties, and X++ code, such as tables, fields, indexes, relations, methods, and so on.

To illustrate the concept of MorphX, assume that you have the license to Microsoft Dynamics AX and you need to edit and develop any object in the standard ERP package. Therefore, this development tool will help you extend the existing functionality to fit your organization's requirements and needs as used by Microsoft to develop the application modules.

You can access these development tools from the following places:

- In the development workspace's **Tools** menu
- In the context menu of elements in **Application Object Tree (AOT)**

 To enable the development mode in Microsoft Dynamics AX 2012, press *Ctrl + Shift + W* to launch the development workspace.

The following table lists the MorphX tools and their purpose:

Tool	Purpose
AOT	This is the core of all development processes and activities. All application objects are stored in a tree organized by the object type.
X++ code editor	Inspects and writes X++ source code.
Compiler	Compiles X++ code into an executable format.
Debugger	Finds bugs in X++ code.
Projects	Groups related elements into projects.
The property sheet	The property sheet shows keys and values. The main purpose is to inspect and modify properties of elements.
The label editor	Creates and inspects localizable strings.
The cross-reference tool	Determines where an element is used.
The Find tool	Searches for code or metadata patterns in the AOT.
The table browser tool	Views the contents of a table directly from a table elements..
The best practices tool	Detects defects in the code and the elements.
The reverse engineering tool	Generates the Unified Modeling Language (UML) element or entity relationship diagrams (ERDs) to be uses in MS Visio.

Application Object Tree

The AOT is the main development menu in Microsoft Dynamics AX. It is easy to navigate through the AOT using the arrows keys on the keyboard.

The root of AOT contains element categories such as:

- **Data Dictionary**
- **Classes**
- **Tables**
- **Forms**
- **Macros**
- **Parts**
- **Data Sets**
- **SSRS Reports**
- **Reports**



Before practicing effectively on AOT, understand the naming structure of all elements. There are thousands of elements that exist in AOT.

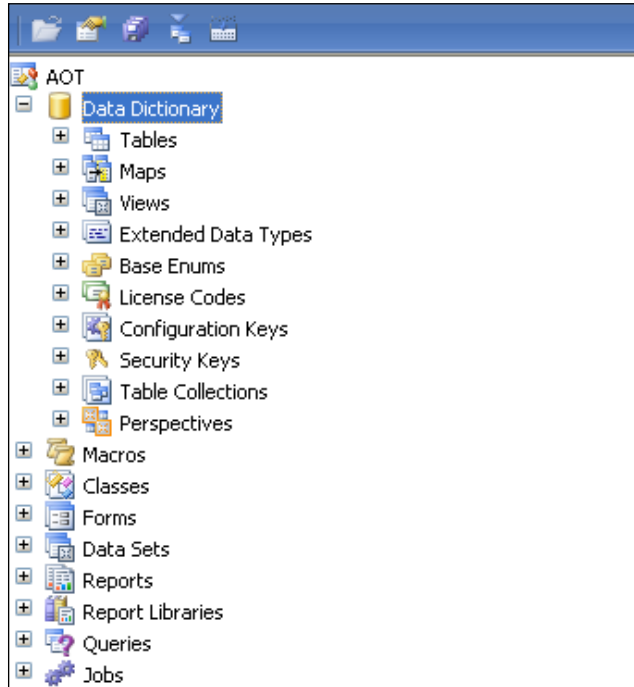
The elements are arranged alphabetically and named by the following structure:

*(Business Area Name) + (Functional Area) + (Action Performed or Type of Content)*Ex: *CustPaymReconciliationImportBusiness Area: Cust = Customer*

Functional Area: PaymReconciliation = Payment Reconciliation

Action Performed: Import = Import

The element categories are shown in the following screenshot:



You can create elements in the AOT node by following the next steps:

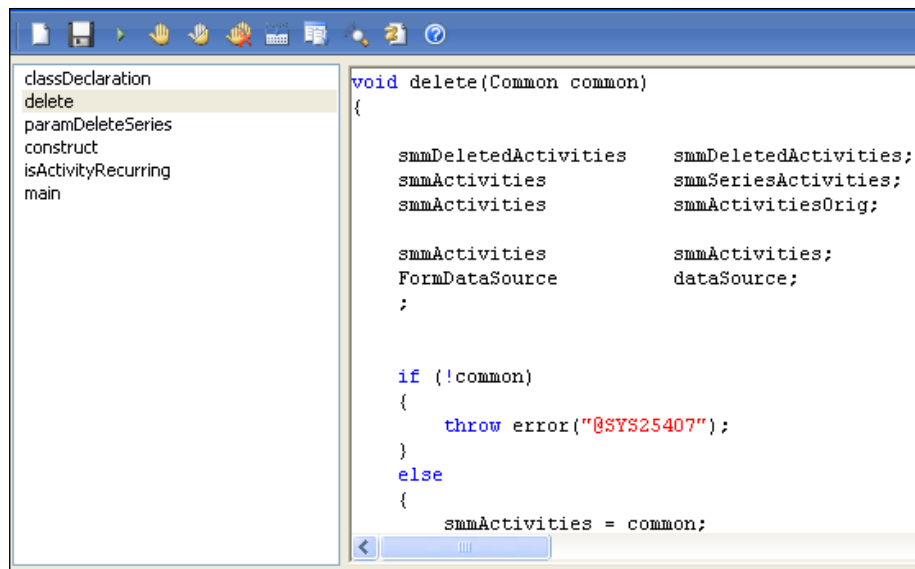
1. Right-click on the element category node.
2. Select **New <Element Type>**.

When you create a new element, generated names are automatically given, and you can replace the default name with a new name.

The X++ code editor

The X++ code editor is a text editor that contains multiple features that you can find in Visual Studio, such as scripting, multiediting, word completion, and so on.

You can write all the X++ code with the X++ code editor by selecting a node in the AOT and pressing *Enter*. As shown in the following screenshot, the X++ editor contains two panes (the left-hand side pane and the right-hand side pane). The right pane shows the X++ code for the method selected in the left-hand side pane:



The following table lists the shortcut keys for the X++ code editor:

Shortcut keys	Action
<i>F1</i>	Shows the help window
<i>F4</i>	Goes to the next error
<i>F5</i>	Executes the current element
<i>F7</i>	Compiles
<i>F9</i>	Toggles a breakpoint
<i>F12</i>	Goes to the implementation (drilled down in the code)
<i>Esc</i>	Cancels the selection
<i>Ctrl + X</i>	Deletes the current selection
<i>Ctrl + I</i>	Incremental search
<i>Ctrl + E, C</i>	Comment selection
<i>Ctrl + E, U</i>	Uncomment selection
<i>Ctrl + Tab</i>	Goes to the next method
<i>Ctrl + Shift + Tab</i>	Goes to the previous method
<i>Alt + R</i>	Run and editor script
<i>Alt + Shift + arrow keys</i>	Enables block selection
<i>Ctrl + Alt + Spacebar</i>	Opens the Label editor
<i>Ctrl + Shift + Spacebar</i>	Shows the method parameter help
<i>Ctrl + Shift + F9</i>	Removes all breakpoints

The X++ code editor contains a set of editor scripts that you can invoke by clicking on the script icon on the X++ code editor toolbar, or you can type the name of the script + *Tab* in the editor. You will notice that there are built-in scripts such as:

- Send to the file
- Send to the mail recipient
- Open the AOT for the element related to the method selected
- Generate the code for standard code patterns such as the `main`, `construct`, and `parm` methods



Parm is a short for **parameter** and is used as simple property getters and setters on classes.

You can create your own scripts by adding new methods to the `EditorScripts` class because the list of editor scripts is extendable.

Compiler

The X++ compiler is a bottleneck for anything you build or install in your own scenarios across the system modules; just as you should compile any programming language, the X++ compiler can compile your code and produce a lot of information such as compiler errors, compiler warnings, and tasks.

In earlier versions of Microsoft Dynamics AX, the compiling processes were designed in three phases:

1. Declaration and method signatures.
2. Metadata validation and p-code generation.
3. Recompilation of elements that had preliminary errors.



In earlier versions in Microsoft Dynamics AX, the phases were:

1. The compilation happens in the client.
2. Metadata is exchanged from SQL to the client and back to SQL.
3. A long compiling duration happens due to deserialization of metadata in memory cache.

In the Microsoft Dynamics AX 2012 R3 compiler, enhancements have been made from an architectural concept that makes the compiling process more productive and faster than in earlier versions.

In the Microsoft Dynamics AX 2012 R3 compiler, the following processes happen:

- The compilation happens on the AOS
- Error logs are generated in each AOS
- No metadata exchange
- X++ execution time during the compilation has been reduced
- More available memory and no cache



Note that you can compile X++ code to **common intermediate language (CIL)** used by .NET Framework to improve the performance.

X++ code is compiled to p-code and the last code is compiled to CIL by following this path: AOT | **Add-ins** | **Incremental CIL generation from X++**.


Depending on your process, the performance improvement can be between 0 and 30 percent. Therefore, you will have to test to know whether performance improves by running your process in CIL.

Debugger

The debugger is a standalone application and is not part of the Microsoft Dynamics AX shell. The debugger allows the debugging of X++ code in any of the following Dynamics AX components:

- Microsoft Dynamics AX client (the **Tools** menu | **Options** | **Development** | **Debug**).
- AOS. From AOS, navigate to the MS Dynamics AX server configuration utility | **Start** | **Administrative Tools** | **Microsoft Dynamics AX 2012 Server Configuration** | **Create a new configuration**. Select the **Enable breakpoints to debug X++ code running on this server** checkbox.
- Business Connector. For enterprise portals, navigate to the MS Dynamics AX server configuration utility | **Start** | **Administrative Tools** | **Microsoft Dynamics AX 2012 Configuration** | **Create a new configuration**. Select the **Enable global breakpoints to debug code running in the Business Connector or client** checkbox.

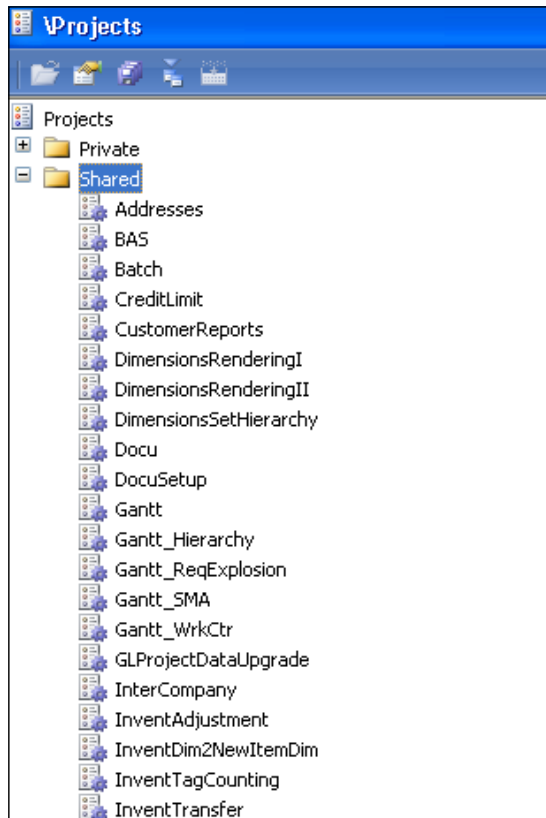
For the debugger to start, a breakpoint must be hit when the X++ code is executed. You set breakpoints using the X++ code editor in the Dynamics AX development workspace. The debugger starts automatically when any component hits a breakpoint.

 To enable or disable a breakpoint, press *Ctrl + F9*.
To list all breakpoints, press *Shift + F9*.
To set or remove breakpoints, press *F9*.
Breakpoint tables are located in **SysBreakpoints** and **SysBreakpointLists** tables.

Projects

In AOT, you can use projects to group and structure elements according to your preference. A project is a powerful tool in the AOT because you can collect all the elements you need for a feature in one project. Projects can be opened from the AOT by clicking on the project icon in the toolbar.

When you create a new project, you should decide whether it should be shared among all developers or between private developers. You can use the **Drag and Drop feature** to move a project from shared to private or vice versa.



You can determine a start up project that opens automatically when Microsoft Dynamics AX is started by specifying a certain project in the options form.

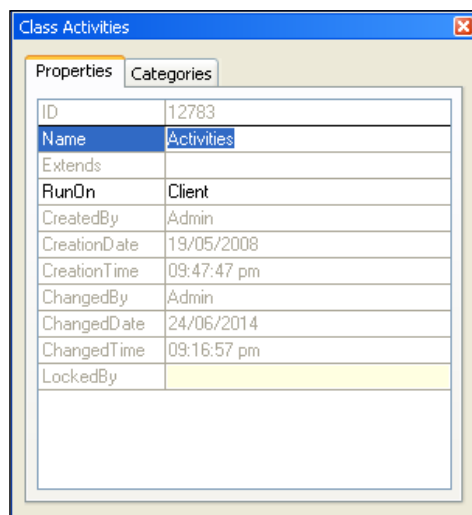
The property sheet

Properties are the backbone of the metadata system; each property is a key and value pair. You can use the property sheet to inspect and modify properties of elements.

By default, the property sheet appears when opening the development workspace. It is automatically updated to show properties for any element selected in the AOT.

The property sheet contains two columns: the key and the value pairs for each property.

In the **Categories** tab on the property sheet, you will find a lot of information related to the selected element, for example, **CreationDate**, **CreatedBy**, **CreationTime**, **ChangedBy**, and so on:



Properties	Categories
ID	12783
Name	Activities
Extends	
RunOn	Client
CreatedBy	Admin
CreationDate	19/05/2008
CreationTime	09:47:47 pm
ChangedBy	Admin
ChangedDate	24/06/2014
ChangedTime	09:16:57 pm
LockedBy	

Docking the property sheet on either side of the screen is very easy, and this can be done by right-clicking on the title bar.

Also, you will notice that there are elements that have time values and user information at the end of every property sheet, and the read-only properties appear in the gray label editor.

The label editor in Microsoft Dynamics AX 2012 is a text resource that is used throughout the whole product. It is a way to help you know more details about any element (the column header, the name of the form in the window, the help text in the status bar, captions on forms, and texts on Web forms).

You can use the label editor as a useful tool to help you when creating support service on Microsoft Dynamics AX or to know where an error message is produced, and this will give you more information related to the element selected.

Labels are localizable, and this means that they can be translated into most languages, because text resources are kept in a Unicode-based label file that must have a three letter identifier (for example, @SYS1234).

The structure of the label file is very simple:

```
@<Label File Identifier><Label ID><Label Text>
```

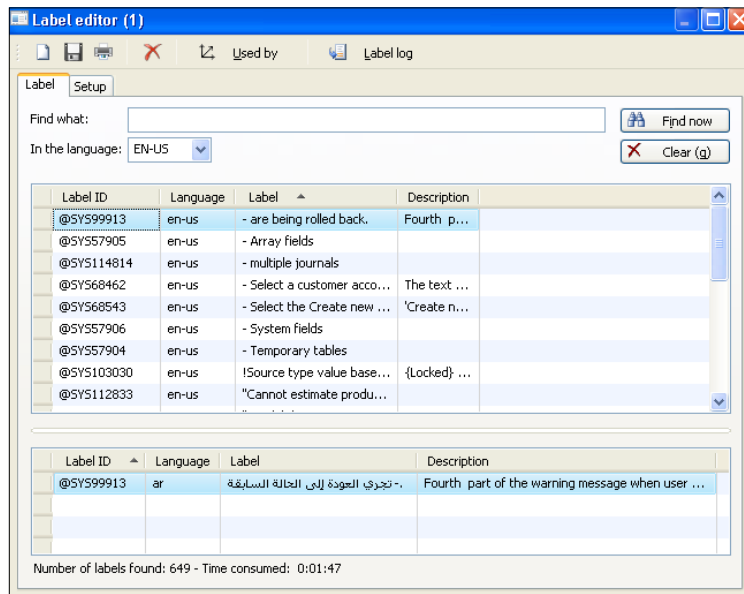


Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

You can create new label files using the **Label File Wizard**, and you can access it directly from the **Label Files** node in the AOT or from the **Tools** menu | **Wizards** | **Label File Wizard**.

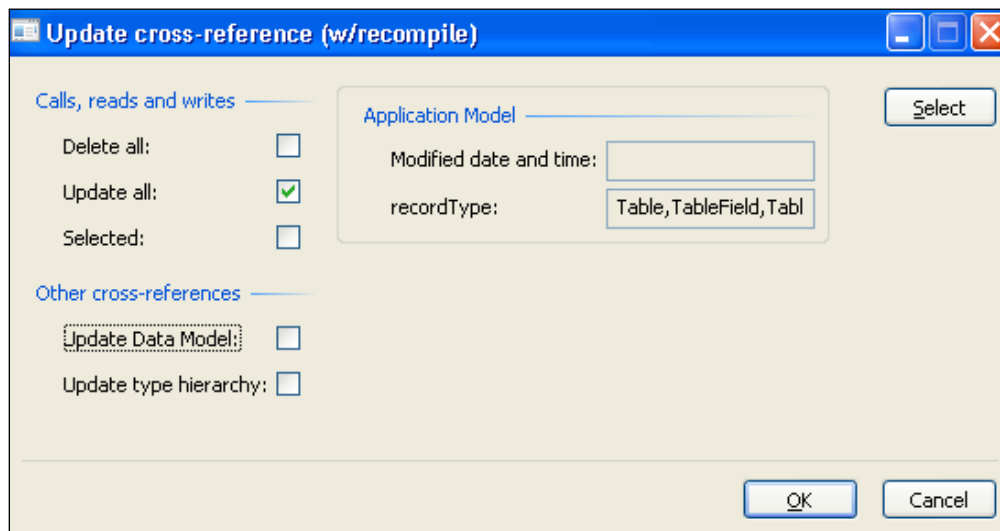
When naming a newly created label, the three-letter ID that you will choose must be unique. You can use your company's initials as an ID:



The cross-reference tool

The concept of a cross-reference tool is very simple; if you have two elements (X and Z) and you want to know which one is in use by the other one, with cross-reference you can determine which elements are in use and which elements out of use.

These relationships between objects or elements are being recorded, so it is easy for you to track changes you or others made previously on all elements, so to keep yourself updated with this information, you must update the cross-reference tool regularly to ensure accuracy. This update will take several hours because it also compiles the entire AOT.



To update the cross-reference tool, go to **Tools** menu | **Cross-reference** | **Periodic** | **Update**.

When the cross-reference tool is updating, it scans all metadata and X++ code.

To preview the whole list of cross-referenced elements, open the AOT, expand the **SystemDocumentation** node, and then click on **Enums** and **xRefKind**.

The Find tool

By pressing *Ctrl + F* from any node in the AOT, a **Find** window appears. It contains most of the useful tools to search for anything in Microsoft Dynamics AX application.

The Find tool contains multiple tabs such as **Date**, **Advanced**, **Filter**, and **Properties**.

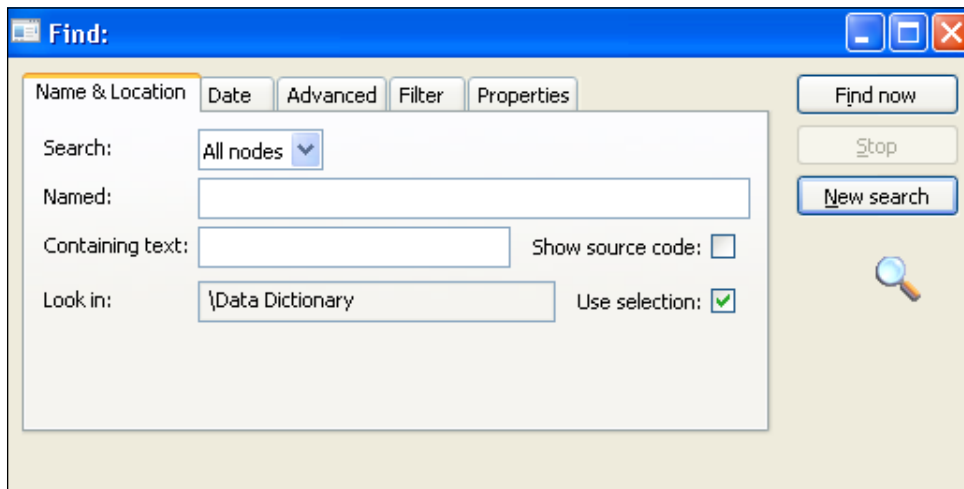
For the **Date** tab, you can specify a range of dates for your search, such as the modified date and who they were modified by.

In the **Advanced** tab, you can specify more advanced settings for your search, such as the layer to search, the size range of elements, the type of element, and the tier on which the element is set to run.

On the **Filter** tab, you can write a more complex query by using X++ and type libraries.

The **Properties** tab appears when **All nodes** is selected in the **Search** list. You can specify a search range for any property. Leaving the range blank for a property is a powerful setting when you want to inspect properties; it matches all nodes, and the property value is added as a column in the results.

The results appear at the bottom of the dialog box as they are found.



The Find tool searches the selected node and related subnodes in the AOT, and if you want to search several nodes, you can mark the **Use selection** checkbox; by unmarking this feature, you will disable this feature.

The table browser tool

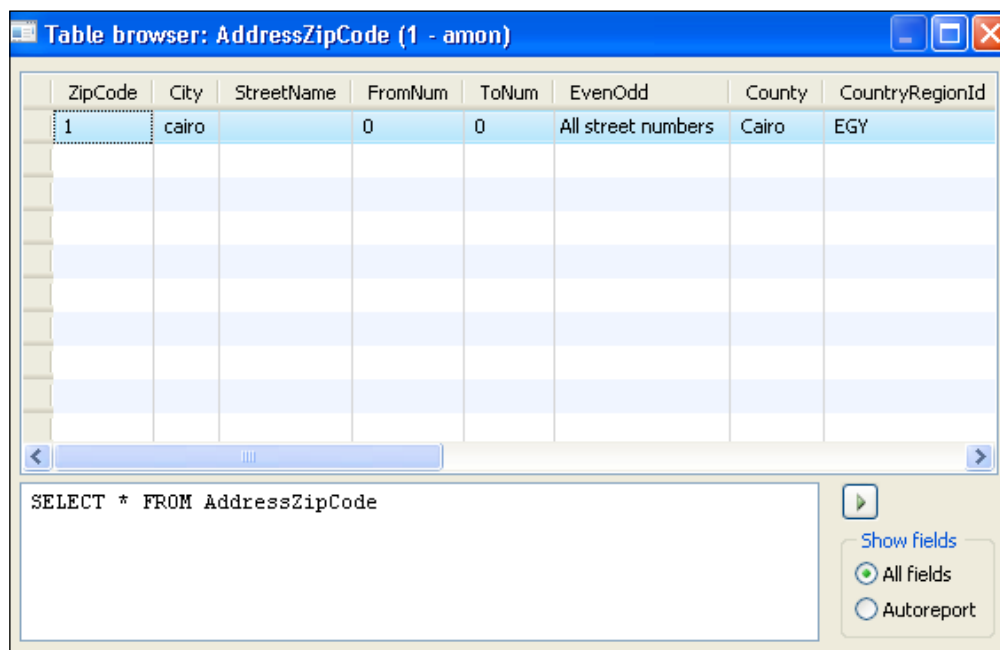
Table browser is just a standard form that uses IntelliMorph to view and edit data in tables. You can use this helpful tool in numerous scenarios, such as debugging, validating data, modifying, cleaning data, and so on.

The table browser tool is implemented in X++, and you can find it in the AOT under the name **SysTableBrowser**.

To open the table browser:

1. Locate the table that you want to view in the AOT.
2. Right-click on the table and then navigate to **Add-Ins | Table browser**.
Alternatively, you can right-click on the table and select **Open**.
3. The table browser displays data from all fields in the table.

In Microsoft Dynamics AX 2012 R3, the table browser tool can be used to run SQL statements by entering the SQL statement in the textbox and just clicking on the **Execute** button to run the SQL against the data source.



You can use the **Autoreport** field group to make it easy for you to find the values you are looking for in tables that have many fields.

The best practice tool

The best practice tool is embedded in the compiler, and its main function is to detect defects and risky code patterns in the X++ code. It is used when making customizations in the application and it is useful to decrease the time and cost that occurs when implementing the application or for any maintenance of the system.

It is the MorphX version of a static code analysis tool that allows any developer to run an analysis of his or her code and application model to ensure that it conforms to a set of predefined rules (400 rules) by displaying deviations from the best practice tool in the compiler output window.



To turn off the best practices tool, go to **Tools** menu | **Options** | **Development** | **Compiler** and set the **Diagnostic Level** value below 4.

The majority of the 400 rules focus on *errors and warnings*, and the best practice tool allows you to suppress errors and warnings and identify the deviation as reviewed and accepted.

To identify a suppressed error or warning, place a line containing the following before the deviation:

```
//BP Deviation Documented.
```

The reverse engineering tool

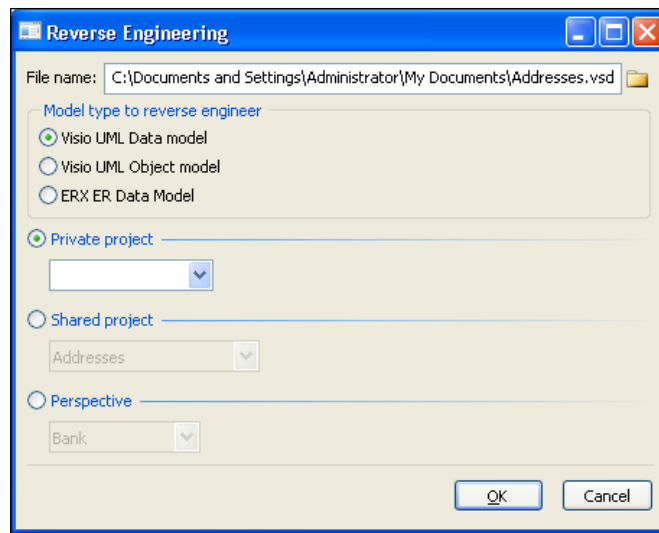
Reverse engineering is a general process of analyzing a specific technology to know how it was designed or how it operates. One of the best known tools in the world is located in Microsoft Dynamics AX as UML.

UML is a general modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of the system.

In Microsoft Dynamics AX 2012, you can generate UML Visio models from your existing metadata or an entity relationship data model and see how they relate to each other in a visualization mode. You must have Visio 2007 or higher to use the reverse engineering tool.

To open this tool, from the **Tools** menu, select **Reverse engineer**; then select the model type as **Visio UML Data model**.

This tool deals with projects, so you will have to select either **Private project**, **Shared project**, or **Perspective**.



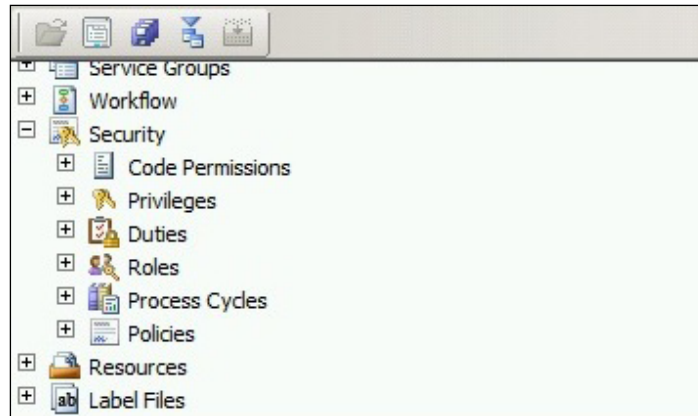
When you click on the **OK** button, all the elements in the selected project generate a Visio document that opens automatically, and any relationships between any elements will be easily visible to you.

Developing a security artifact using AOT

A security artifact is an architect of a security system within an entire system or application environment. In Microsoft Dynamics AX 2012, the security system contains the following artifacts (ordered by hierarchy):

- **Policies:** Security policies are a set of security roles that control the working environment
- **Security role:** This represents the scope of work for every person in the organization
- **Duties:** These are responsibilities that perform tasks for a specific business objective or process cycle, and they contain a set of application process privileges
- **Privileges and permissions:** These are a group of access rights granted to users

The following screenshot shows the security artifacts that you can develop from the AOT to build your security environment:

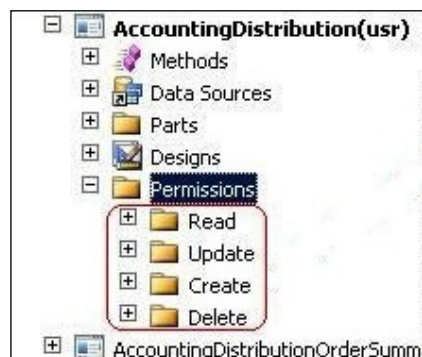


Setting permissions for a form

The first step is to control access to the data in any form in the AOT; if you open any form in the AOT, you can set permissions to CRUD:

- Create
- Read
- Update
- Delete

These types of permission are set automatically for tables that are used in the form (CRUD; this function is called **auto-inference**. Auto-inference configures table permissions in a form (CRUD), and the system automatically adds or updates the (CRUD) nodes by navigating to AOT | **Forms** | **<FormName>** | **Permissions**.



You can set up the permission manually, and you can do this not only for a form, but also for several AOT elements that include:

- **Services** | <ServiceName> | **Operations** | <OperationName>
- **Reports** | <ReportName>
- **Parts** | **Info Parts** | <InforPartName>
- **Forms** | <FormName>
- **Web** | **Web Files** | **Web Controls** | <WebControlName>

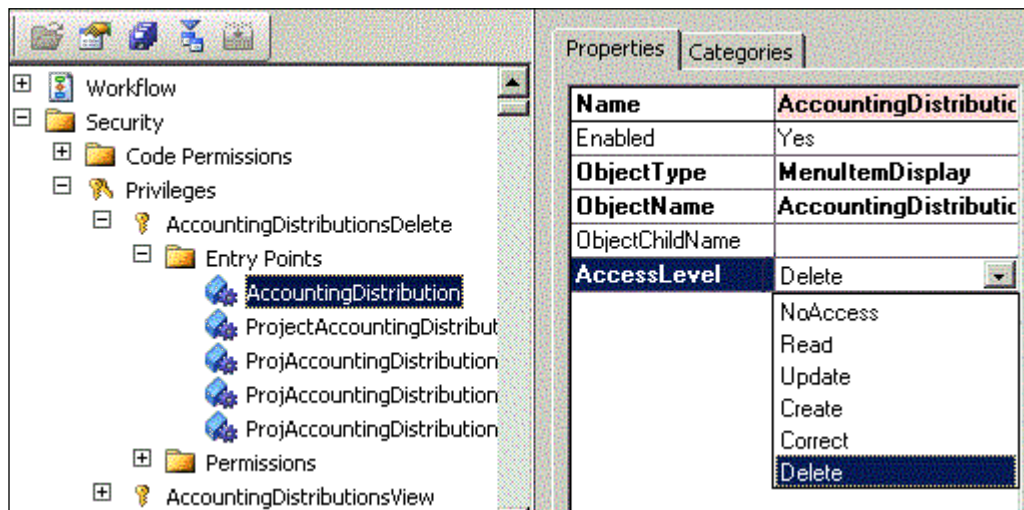


When you open the **Permission Type (Read)** node by navigating to AOT | **Forms** | <FormName> | **Permissions**, you will notice that you have the ability to set controls for a table as securable objects or server methods.

Assigning permissions to privileges

Privileges are a set of permissions that provide access to a securable object. It is the second step after specifying permissions when developing a security artifact.

Using the auto-inferred table permissions and securing menu items with privileges, you can control access to the data in a form.



In the preceding screenshot, the **AccountDistCustFreeInvoiceMaintain** privilege contains an entry point, **AccountingDistCustFreeInvoice**. This is a menu item that points to a form.

In the **Properties** tab, the **AccessLevel** value is set to **Delete**, and this means that when a user accesses the form through this particular menu item, the security framework in MS Dynamics AX will be under the **Permissions | Delete** node in this form and will grant access to the tables that are listed under that node.

In this example, you will notice a relation between the privileges, entry points, and permissions that determine the user access permissions if they access this privilege through a security role.

The menu items in the AOT act as a higher layer of abstraction for a form, reports, and so on, and it contains a complete list of the items that can be presented in a menu.

Each menu item has the following security properties:

- Create permissions
- Read permissions
- Update permissions
- Correct permissions
- Delete Permissions



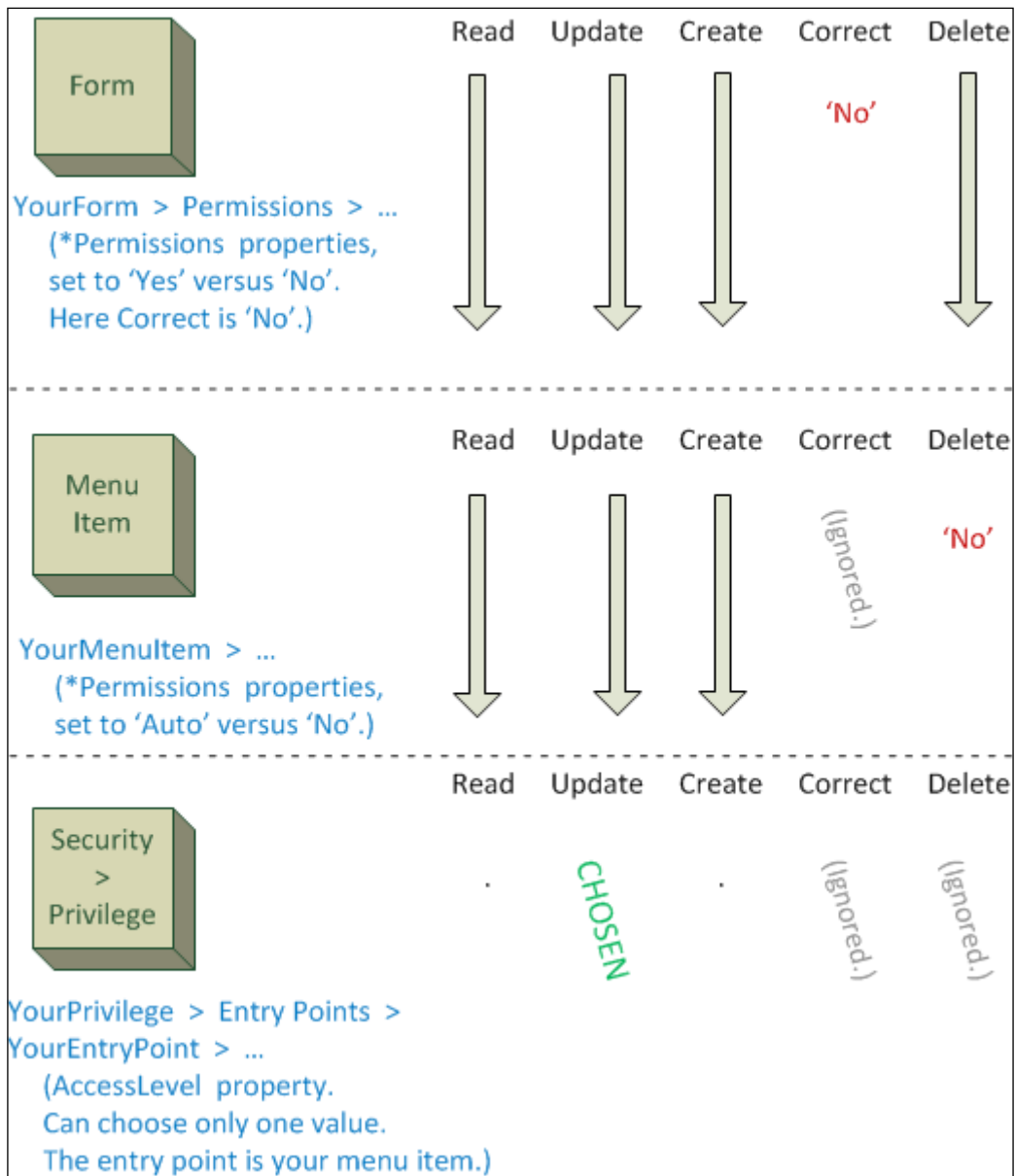
These properties can refer to the nodes under this: AOT | **Forms** | **<FormName> | Permissions**.

Entry points refer to a programming object that is at the start of an application functionality and can be directly associated with privileges.

Entry points can be referred to a lot of object types, such as form, job, info part, query, report, SSRS report, and class.

For example, for an entry point referring to a menu item that is referring to a form, permissions are defined in the AOT node as **Permissions | Form | Menu Item | Entry Point** (on a privilege).

The following figure (Developer Network on <http://msdn.microsoft.com>) illustrates the sequence of using auto-inferred permission:



As a security best practice, you have to check the following:

- One entry point must be assigned to a privilege
- Every privilege must be contained in at least one duty
- Every duty must be contained in at least one role
- Every role must be contained in at least one process cycle

Validating and testing a security privilege

After you implement the data security structure in Microsoft Dynamics AX 2012, you will want to make sure that you make accurate changes. For the testing process, you need to do the following:

1. Create a role by navigating to AOT | **Security** | **Roles**.
2. In the AOT, assign the appropriate duty or privilege to the new role.
3. Create a test user account (such as `axtest3`) by going to **System Administration** | **Common** | **Users** | **Users**.
4. Assign the user to a role by going to **System administration** | **Setup** | **Security** | **Assign users to roles**.
5. Start the application with a command line or shortcut that is similar to the following (wrapped lines):

```
%windir%\system32\cmd.exe /c runas /savecred  
/user:mywindowsdomain\axtest3 "C:\Program Files  
(x86)\Microsoft Dynamics AX\6.0\Client\Bin\Ax32.exe"
```

Applying a configuration key

Configuration keys allow administrators to set the security for a user group by minimizing access to a user group to reduce the potential attacks.

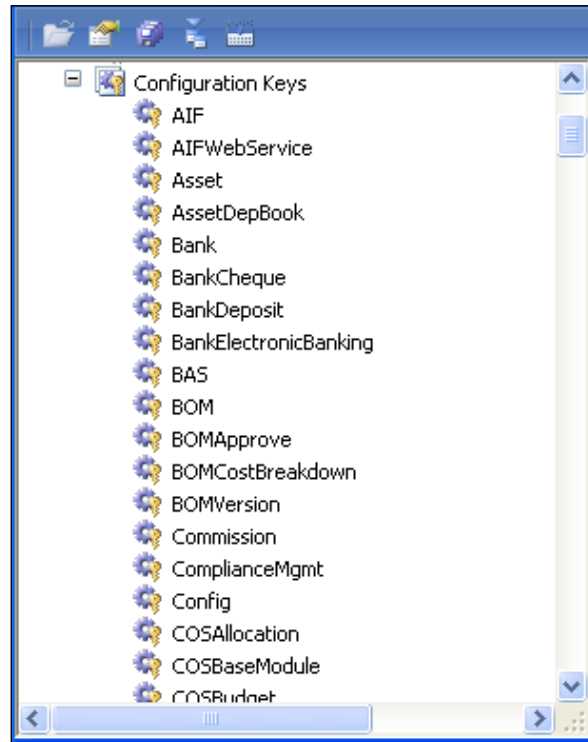
The benefit of the configuration key is to protect sensitive data in the database and prevent users from changing code and objects in the application.

Configuration keys are applied for the following:

- Tables
- Fields
- Views
- Menus
- Menu items

- Form controls
- Indexes
- Extended data types
- Report controls

The following screenshot shows the configuration keys:



To create a configuration key, follow these steps:

1. Expand the **Data Dictionary** node in the AOT.
2. Right-click on the **Configuration Keys** node and select **New Configuration Key**.
3. Right-click on the configuration key and click on **Properties**.
4. Rename the configuration key by modifying the **Name** property.
5. Right-click on the object and click on **Create** in the shortcut menu.
6. Right-click on the object and click on **Save** in the shortcut menu.



When you disable a table in the configuration key that is listed in the AOT in Microsoft Dynamics AX 2012, you must decide whether to manually delete the data that is in the table or not. In the earlier versions of Microsoft Dynamics AX, when you disable a table in the configuration key, the table is dropped from the SQL Server and all the data is deleted. This change happened in AX 2012.

Summary

Through out this chapter, you got a solid introduction to the MorphX development tools. Now, you can use each development feature in a smooth and fast way. Besides developing a security artifact by creating privileges and permissions and assigning them to each other, you are now able to validate, test, and debug the security privileges and roles that you created previously. This is a good start that will make you go through the advanced topics coming next.

In the next chapter, we will learn about the fundamentals of security coding using X++, code access security, security debugging, security in display and edit methods, and the Table Permissions Framework.

2

Security Coding

In this chapter, we are going to illustrate X++ security coding and how to use **Code Access Security (CAS)**. We will also talk about the new compiler enhancement in Microsoft Dynamics AX 2012 R3 and how it affects the compiling process within the system and the debugging tools. Finally, we will introduce the new features in Microsoft Dynamics AX 2012 R3.

The topics covered in this chapter are as follows:

- The fundamentals of security coding using X++
- Using Code Access Security
- Security debugging
- Security for the `Display` and `Edit` methods
- The Table Permissions Framework

The fundamentals of security coding using X++

In this chapter, we will cover the trustworthy computing features of Microsoft Dynamics AX 2012 and how they affect security coding.

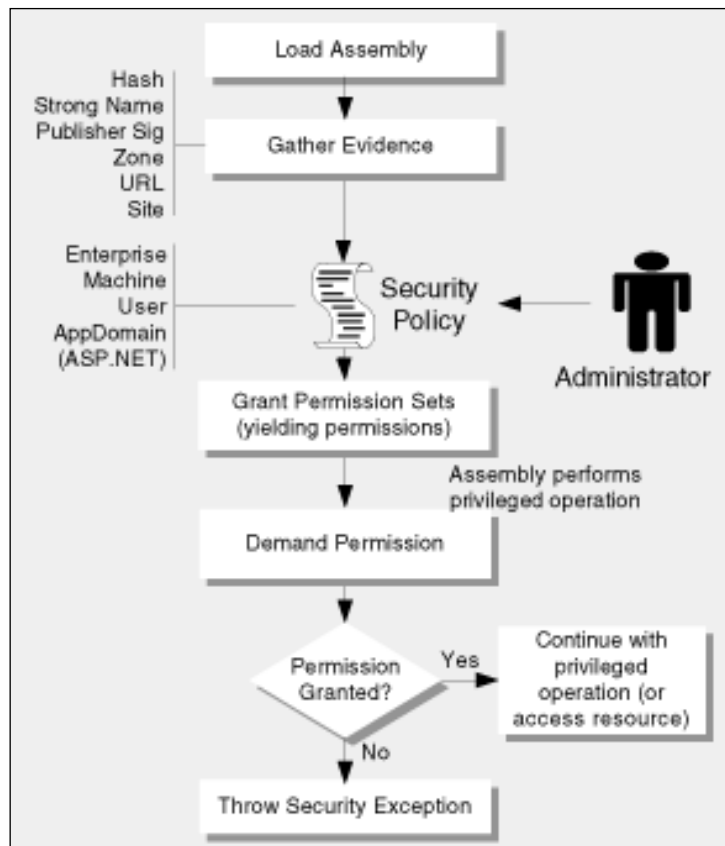
X++ is an object-oriented language like any other programming language, for example, C++. It allows developers to design data types, tables, queries, forms, menus, and reports. It also allows access to the application code by launching the X++ code editor.

X++ has a lot of features and tools that help you edit the structure of the application and avoid common programming pitfalls that can prevent you from implementing X++. It includes keywords that match most of the keywords in standard SQL.

Using Code Access Security

The CAS framework is a mechanism developed to protect systems from dangerous APIs that are invoked by untrusted code. It simply allows two pieces of code to communicate in a manner that can't be compromised.

To know what untrusted code is, we will first define what trusted code is. Trusted code is defined as code from the AOT running on the **Application Object Server (AOS)**, meaning the code must be written by a trusted developer. This is because the developer privileges and permissions are the highest level of permission over the entire application. In other words, if you execute the code outside the AOS on a client, there is the possibility that it was altered on the client side before execution, thus rendering the code untrusted. The following figure demonstrates this mechanism:





For more information on CAS, visit Microsoft MSDN:
<http://msdn.microsoft.com/en-us/library/ff648663.aspx>.

The main purpose of CAS is securing the tunnel between an API and its consumer through the `CodeAccessPermission` class or a similar class. The consumer should request to call the API, which is done by calling `CodeAccessPermission.Assert`. The API then verifies that the consumer has the correct permissions by calling `CodeAccessPermission.demand`. The demand method searches the call stack for matching assertions; then, if untrusted code exists on the call stack before the matching assertion, an exception is raised:

```

Class WinApiServer
{
//Delete any given file on the server
Public server static Boolean deleteFile(Filename_filename)
{ FileIOPermission      fileIOPerm;
//Check file I/O permission
fileIOPerm = new FileIOPermission(_filename, 'B');
//Delete the file
System.IO.File::Delete(_filename); }
}
Class Consumer
{{ //Delete the temporary file on the server
Public server static void deleteTmpFile()
{ FileIOPermission      fileIOPerm;
FileName                filename = '@d:\DBM\tmp.tmp';
// Request file I/O permission
fileIOPerm = new FileIOPermission(filename, 'b');
CodeAccessPermission::revertAssert();
// Use CAS protected API to delete the file
WinApiServer: : deleteFile(filename);
}
}

```



When using `Assert`, you have to make sure you don't create a new API that is just as dangerous as the one that CAS has secured.

`WinAPIServer::deleteFile` is recognized to be a dangerous API because it exposes the .NET API `System.IO.File::Delete(string filename)`. This allows the user to remotely delete files, which could cause the server to come down.

The Demand methods enforce security analysis in every calling stack that is examined for a specific permission, and when a demand is triggered the following occurs:

1. The stack walk begins at the caller's stack frame and not at the current stack where the demand occurs. For example, if method A calls method B and method B has a demand, the stack walk begins at method A's stack frame. Method B is never evaluated as part of the stack walk.
2. The stack walk proceeds through the call stack until it reaches the program's entry point to the stack (usually the `Main` method) or until a stack walk modifier like `Assert` is found.
3. When a demand and a stack walk modifier (`assert`, for example) for the same permission appear on the same stack frame, the demand takes precedence.



Declarative and imperative syntaxes exhibit no difference in behavior.

Note that a demand placed on your program's entry point never gets evaluated because the stack walks always begin at the calling stack frame, but in this case, there is no such calling frame to evaluate. Therefore, demands placed on a program's entry point always succeed.

For more information about the Demands method, visit MSDN:
[http://msdn.microsoft.com/en-us/library/9kc0c6st\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/9kc0c6st(v=vs.110).aspx).

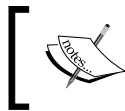
The X++ language belongs to the **Curly Brace** family of programming languages, which define statement blocks using the curly brackets (`{ }`). Thus it is influenced by the C, C++, and C# programming languages.

Securing an API on the AOS

You can make an API secure by extending the `CodeAccessPermission` class. Any class that is derived from `CodeAccessPermission` is trusted by checking for the appropriate permission.

Follow these steps to secure a class on the server tier:

1. You can derive a class from the `CodeAccessPermission` class or you can use one of the following classes that are embedded in MS Dynamics AX. Then you can skip to step 6:
 - `ExecutePermission`
 - `FileIOPermission`
 - `RunAsPermission`
 - `InteropPermission`
 - `SkipAOSValidationPermission`
 - `SqlDataDictionaryPermission`
 - `SqlStatementExecutePermission`
 - `SysDatabaseLogPermission`
2. Create a method with reference to the class.
3. Create a constructor for all class parameters that store permission data.
4. To know which permissions are required to invoke the API that you are securing, override the `CodeAccessPermission` class. Check whether the `SubsetOf` method can be compared with the derived permission class to `CodeAccessPermission`.
5. Override the `CodeAccessPermission.copy` method to return a copy of an instance of the class created in the first step to prevent the class object from being modified and passed to the secured API.
6. Call the `CodeAccessPermission.demand` method before executing the API functionality that you are securing. The method will check the call stack to determine whether the permission has been granted to the calling code.



X++ has built-in support for querying the code and the syntax statements are similar to SQL statements.

Security debugging

The debugger tool provides debugging capabilities for X++ developers and it also communicates with the Microsoft Dynamics AX client, .NET Business Connector, or batch jobs that run on the Microsoft Dynamics AX server.

We're going to illustrate the following:

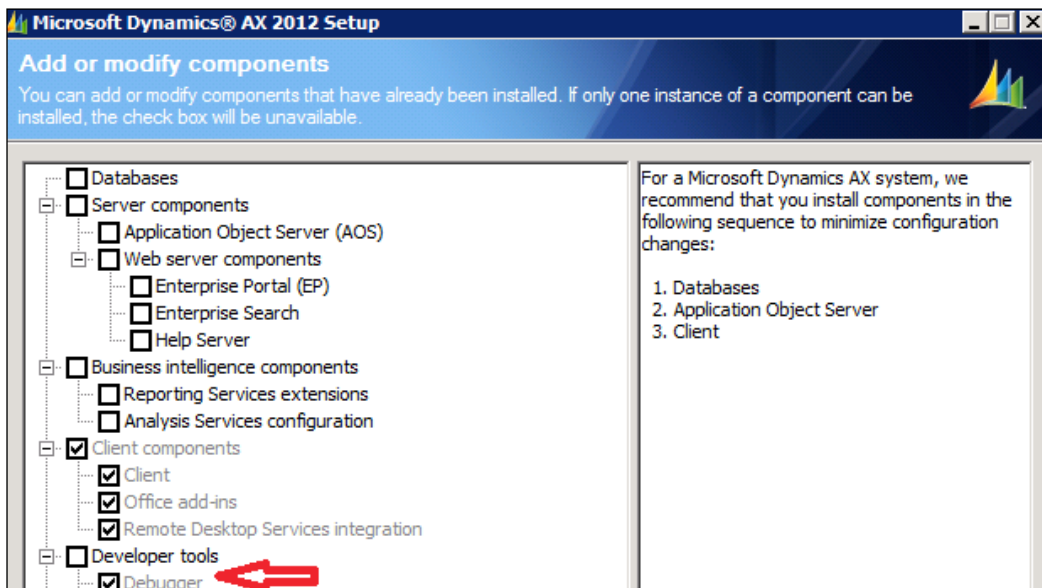
- Installing the debugger tool
- Enabling the debugger tool
- Adding users to the debugging user local group
- The debugger user interface
- Debugger shortcut keys

First you have to make sure you have installed the debugger tool from Microsoft Dynamics AX. Go to **Setup | Install | Microsoft Dynamics AX components**.

Installing the debugger

In this section, we will see how to install the debugger tool.

From the **Microsoft Dynamics AX 2012 Setup** window, you can install the debugger to add, remove, or modify any component that is installed or should be installed:



To begin installing the debugger component, go through the following steps:

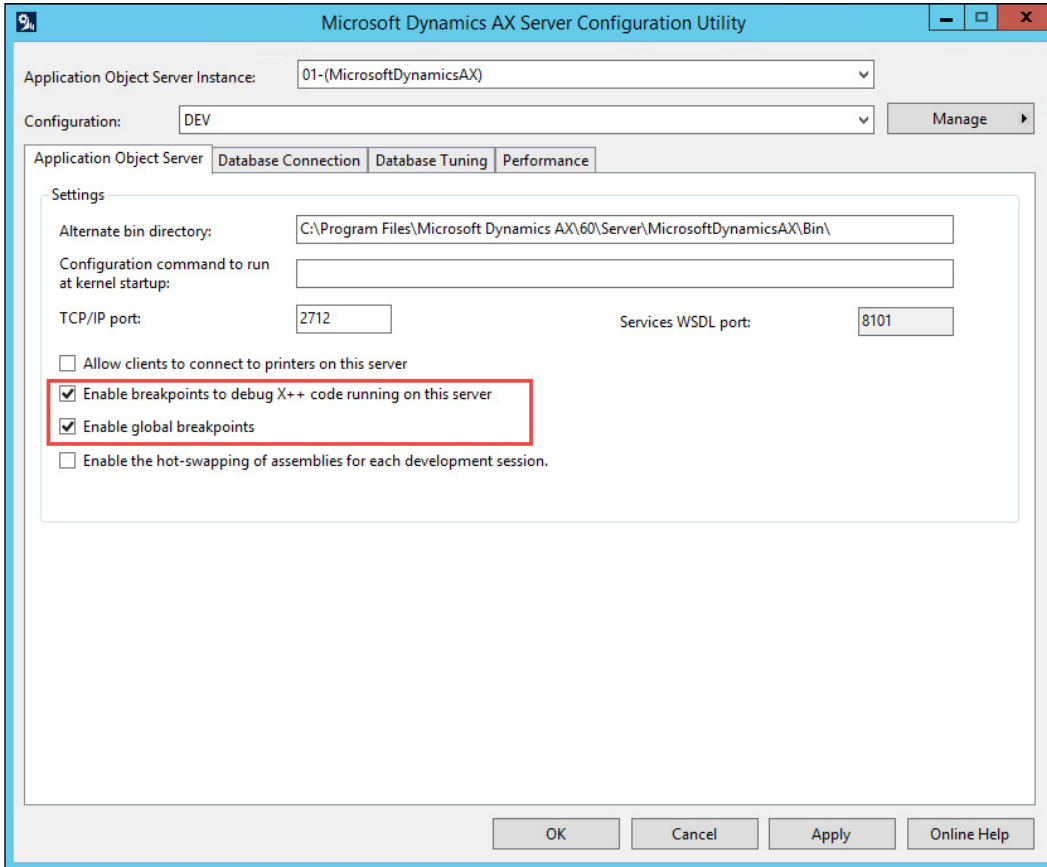
1. From the **Microsoft Dynamics AX Setup** window, go to **Install | Microsoft Dynamics AX components**.
2. Provide a file location or accept the default location.
3. On the **Ready to install** page, click on **Install**.
4. If you're installing AX 2012 R3, click on **Microsoft Dynamics AX** from the **Select an installation option** page.
5. On the **Select installation type** page, click on **Custom installation**.
6. On the **Select components** page, select **Debugger**.
7. On the **Prerequisite validation results** page, resolve any errors. When no errors remain, click on **Next**.
8. On the **Ready to install** page, click on **Install**.
9. After completing the installation process, click on **Finish** to close the wizard.

Enabling the debugger

After installing the debugger tool, you need to enable the debugging tool from the server by marking the following:

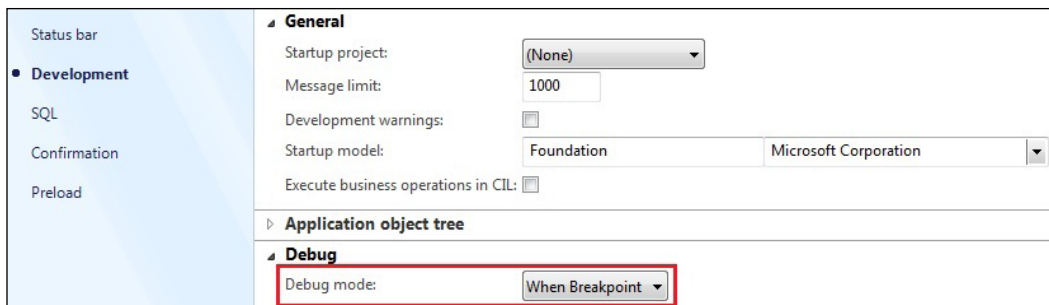
- **Enable breakpoints to debug X++ code running on this server**
- **Enable global breakpoints**

This is shown in the following screenshot:



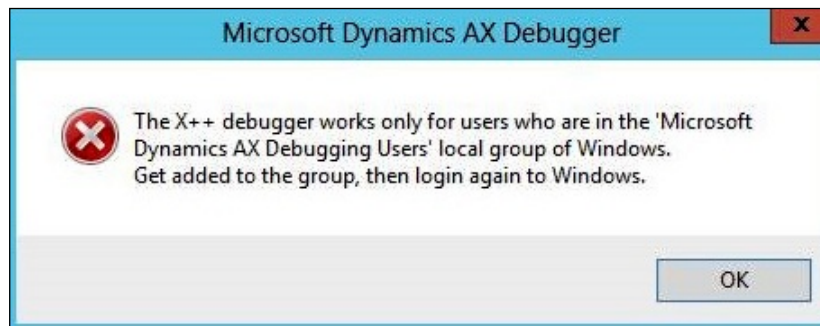
These checkboxes can be accessed by going to **Start | Administrative Tools | Microsoft Dynamics AX 2012 Server Configuration**.

By following these steps, you enabled the debugging tool from the server. To enable the debugging tool on the client side, go to **Microsoft Dynamics AX 2012 | Tools | Options | Development**. From the **Debug mode** dropdown, select **When Breakpoint**:



Adding users to the Debugging User local group

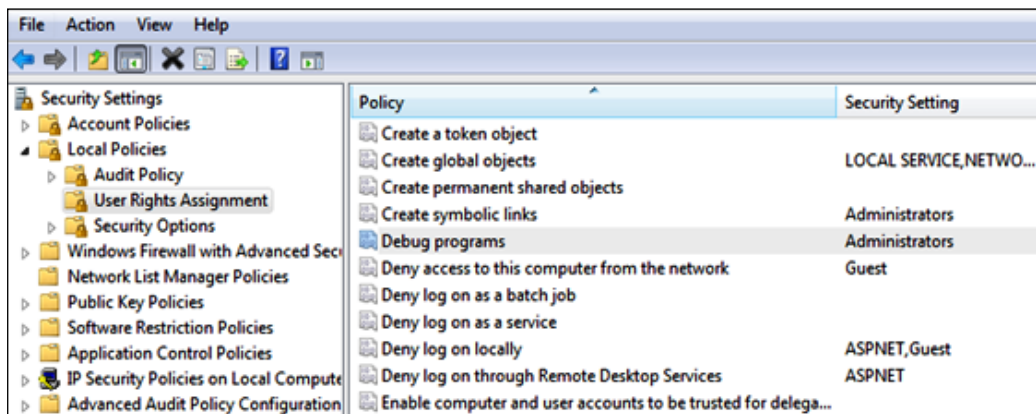
After installing the debugger tool and enabling the debugging feature (either from the server or the client), you must add users to the Debugging User local group to avoid the following error message:



This error message appears if you don't add the user to the Microsoft Dynamics AX Debugging Users local group. The administrator or the user who installed the security debugging tool is automatically added to the local group.

To start using the debugger, the user must belong to the Debugging Users local group on the computer. To add users to local group, follow these steps:

1. Click on **Start** menu and select **Control Panel**.
2. In **Control Panel**, double-click on **Administrative Tools**.
3. In **Administrative Tools**, double-click on **Local Security Policy**.
4. From the **Security Settings** window, expand the **Local Policies** folder.
5. Click on **User Rights Assignment**.
6. From the **Policy** column, double-click on **Debug programs** to view the current local group policy assignments in the **Local Security Setting** window.
7. To add new users, click on the **Add User or Group...** button.
8. The following screenshot shows the interface to start the debugger:



Before debugging security roles in Dynamics AX 2012, you must choose a user who belongs to the system administrator role, and then assign the role you want to debug to that user. Follow these steps:

1. Close all Microsoft Dynamics AX instances.
2. Open the Microsoft Dynamics AX development environment.
3. Open another instance for Microsoft Dynamics AX client.
4. Add the role you want to debug to your user ID.
5. In the development environment, set breakpoints in the X++ code.
6. Create a job, add the following line, and then execute the job:
`SecurityUtil::sysAdminMode (False) ;`
7. From the development environment, press *Ctrl + W* to open the application workspace.
8. You are able to open the client with a reduced permission and also have the ability to debug the X++ code.



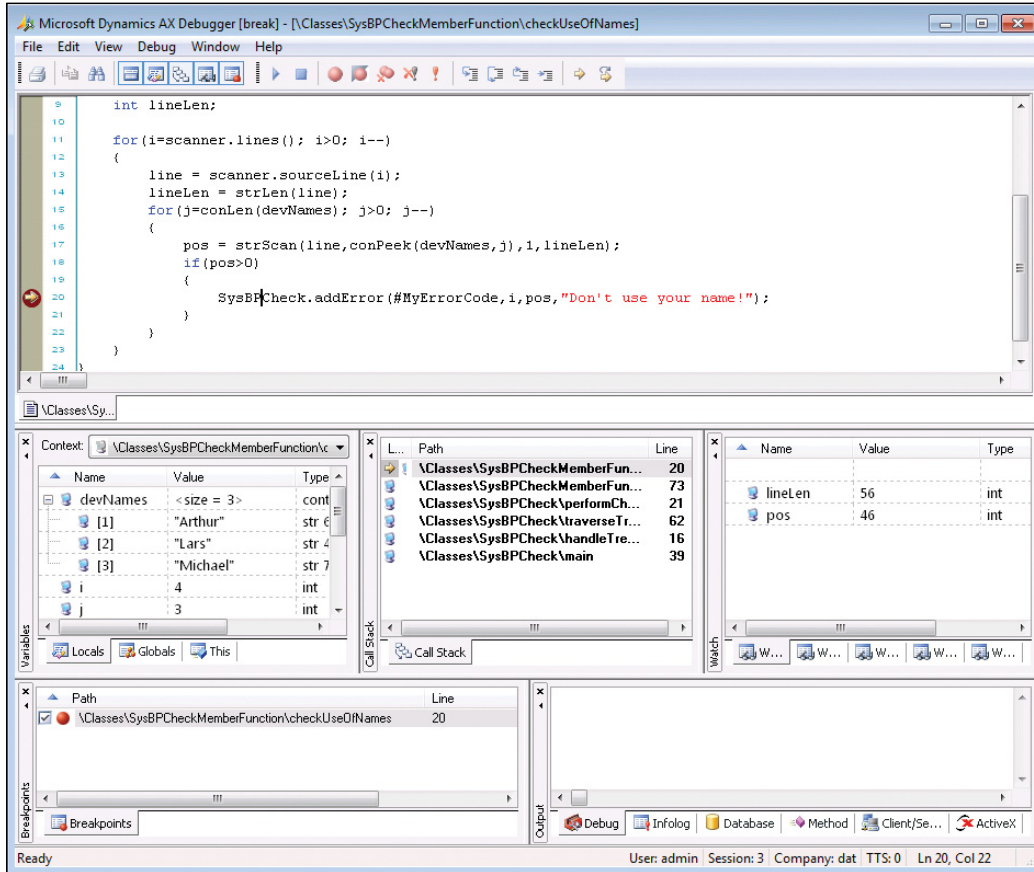
These procedures will not work for enterprise portals or for code executed using the X++ RunAs API.

9. To set the environment back to the system administrator role, update the job created before with the value set to `true`:

```
SecurityUtil::sysAdminMode (true) ;
```


The debugger user interface

In the following screenshot, you will see the debugger window opened and all windows enabled:



Source: www.microsoftpresstore.com

The following table is a brief description of every element in the preceding window:

Debugger element	Description
The Code window	The Code window shows the current X++ code.
The Breakpoints window	The Breakpoints window lists all breakpoints made by you; you can delete, enable, or disable the breakpoints through this window.
The Call Stack window	The Call Stack window displays the code path followed to arrive at a specific execution point.
The Variables window	The Variables window shows the local, global, and member variables with their names and values.
The Watch window	The Watch window shows the name, value, and type of different variables. You can drag and drop variables from the Code window to this window to inspect them as you prefer.
The Output window	The Output window shows the output sent to the info log and any traces enabled on the Development tab in the Options form.
The status bar window	The status bar shows the current user ID who is logged in to the system, the current session ID on the AOS, the current company account, and the current transaction level.

Debugger shortcut keys

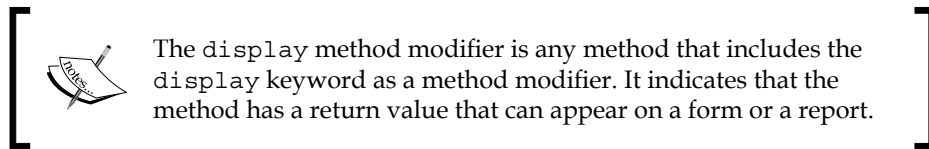
The following table lists the most important shortcut keys that are available in the debugger, and you can find out about the other shortcut keys by using **help**:

Shortcut key	Action
<i>F5</i>	Run
<i>F10</i>	Step over
<i>F11</i>	Step into
<i>Shift + F5</i>	Stop debugging
<i>Shift + F11</i>	Step out
<i>Shift + F9</i>	Toggle breakpoint
<i>Ctrl + Alt + B</i>	Breakpoint window
<i>Ctrl + Alt + C</i>	Call stack window
<i>Ctrl + Alt + O</i>	Output window
<i>Ctrl + Alt + V</i>	Variables window
<i>Ctrl + Alt + W</i>	Watch window

Security for the display and edit methods

The `display` method is not a physical field in a table but is a method to display information from another table or resource. In simple words, it can expose data from any table that includes the `display` keyword as a method modifier. The methods that can use the `display` method modifier are as follows:

- Table
- Form
- Report
- Report Design
- Data Source




The following example is a standard `display` method from the `SalesTable` table:

```
display CustName customerName()  
{  
    return this.partyTable_CustAccount().Name;  
}
```


To create a `display` method, follow these steps:

1. Place the `display` keyword in front of the method's return type.
For example:
`display Amount amount()`
2. Determine the return type. The return type should be an extended data type. A `display` method returns a calculated value like a sum or a count.
3. Determine whether to add parameters to the `display` method. The following list explains when to add a parameter:
 - A `display` method for a table, form, or report does not have any parameters. For example:
`display Amount amount()`
 - A `display` method for a form data source does require a parameter. You use the parameter to specify a table buffer. The type of table buffer has to match the type of the table in the form data source.

 X++ allows single- and multiline comments. We need to start and end with // for single-line comments, and /* for multiline comments.

The following example shows a `display` method that returns the number of customer records that appear in the form's data source. Note how the parameter has the type `CustTable`. This table is in the form data source.


```
display NumberOfRecords testMethod(CustTable myTable)
{
    NumberOfRecords recCount = this.totalNumberOfRows();
    Return recCount;
}
```

 When complex methods are used to calculate values on grids, performance dips because methods are run repeatedly with no reason.

By using `CacheAddMethod` of the `datasource` object, the display methods can be cached and the display will only be calculated when necessary, and thus the performance will be improved:

```
public void init()
{
    super();

    // Enable caching of the document handling display fields
    dirPartyTable_ds.cacheAddMethod(tablemethodstr(DirPartyTable,
        showDocHanIcon));
}
```

 The method in a class is a member that uses statements to determine the behavior of any object. Methods can be defined with `public` or `private` access modifiers, and if the access modifier is obsolete, the method is accessible. Additional modifiers supported by X++ are `Abstract`, `Client`, `delegate`, `display`, `edit`, `final`, `server`, and `static`. The method parameters have default values and are used when parameters are omitted from method invocations.

When a user has access to a certain table, they can use the `display` method to expose any data from any other table, even if they don't have authorized access to any other tables, so you have to evaluate the business impact of the retrieved data from the `display` method.

To resolve any conflict associated with the use of the `display` and `edit` methods, follow these steps:

1. Evaluate each `display` method that brings data from another row in the same table or from a different one.
2. Evaluate each `display` method regardless of whether the data is affected by any threat.
3. If the data is affected by a threat, you can perform the explicit authorization checks and throw an exception if access is unauthorized.

The following code shows an explicit authorization:

```
if (hasSecurityKeyAccess (securitykeyNum (mySecurityKey) ,
    AccessType::View))
{
    myMethod();
}

if (hasMenuItemAccess (menuItemDisplayStr (myMenuItem) ,
    MenuItemType::Display))
{
    myMethod();
}

DictTable dictTable = new DictTable (tblenum (myTable));
if (dictTable.rights >= AccessType::Insert)
{
    myMethod();
}

if (isConfigurationkeyEnabled (configurationkeyNum (myConfigurationKey))
{
    myMethod();
}
```



X++ includes statements that allow interop with .NET CLR and COM components by dispatching method calls from the Dynamics AX object to the wrapped object.

The Table Permissions Framework

The **Table Permissions Framework (TPF)** enables administrators to add an additional level of security to tables that are located in the database and available through the AOT. TPF adds table-level security that verifies the access rights to the requested data. To enable TPF, the administrator must specify a value `AOSAuthorization` property on a specific table in the AOT, and the following table describes the possible values of the `AOSAuthorization` property:

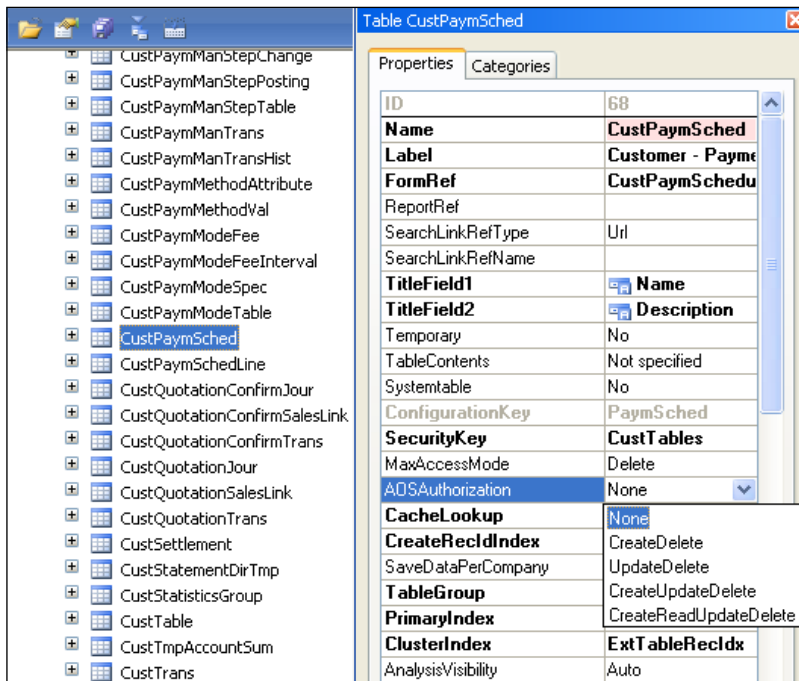
Value	Description
<code>CreateDelete</code>	Create and delete authorization validation is performed on the AOS
<code>UpdateDelete</code>	Update and delete authorization validation is performed on the AOS
<code>CreateUpdateDelete</code>	Create, update, and delete authorization validation is performed on the AOS
<code>CreateReadUpdateDelete</code>	All operations are validated on the AOS
<code>None</code>	No AOS authorization validation is performed (default value)


You can also add additional rules for validation by using the following table methods (use these methods carefully because they can slow down performance):

- `aosValidateDelete`
- `aosValidateInsert`
- `aosValidateRead`
- `aosValidateUpdate`

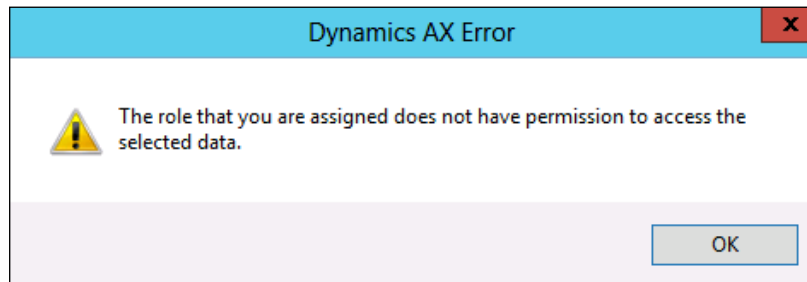
To enable TPF on a database table, follow these instructions:

1. In the AOT, go to **Data Dictionary | Tables**.
2. Right-click on a table and then click on **Properties**.
3. Click on **AOSAuthorizationProperty** and select a new value by using the drop-down list.
4. Click on **Save All**:



 The AOS acts as a security guard by verifying whether the users that are trying to access the requested table have the right permission; if not, the AOS doesn't complete the operation.

After enabling the TPF, you have to ensure that the security roles can access the data in the protected table – this will prevent the following error message from appearing:

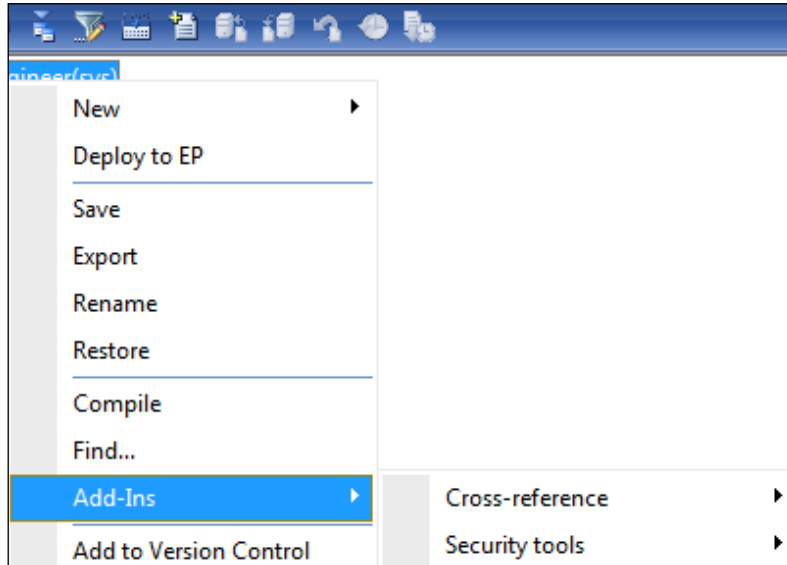


To ensure whether the roles require access to a requested table or not, follow these steps:

1. Go to the AOT and expand **Data Dictionary | Tables**.
2. Right-click on the requested table and go to **Add-Ins | Security tools | View related security roles**.
3. The **Roles related to table** form is displayed, which shows all the security roles that access the requested table and the attached privileges.
4. You can grant the role access to the table protected by TPF by using the **Override permissions** form.

If you need to know all the objects and permissions that access the protected table by TPF, just follow these steps:

1. Go to the AOT and expand **Data Dictionary | Tables**.
2. Right-click on the requested table and go to **Add-Ins | Cross-reference | Update**:



3. Right-click on the requested table again and go to **Add-Ins | Cross-reference | Used By**.
4. The **Used By** form that is displayed shows all objects that access the requested table and the permissions required to access the table.



It is recommended to perform TPF in a test environment first. This is so you can recognize the changes that happen to the selected table and the impact on users or user groups that access the selected table.

Summary

By the end of this chapter, you are able to use CAS to secure your environment and also know how to debug the security coding. You also learned how to use the TPF feature to secure your data and add an additional level of security to specific tables and fields.

In the next chapter, we are going to learn how to secure business data by using policies and authorization. We will also understand the new revolution of record-level security in Microsoft Dynamics AX 2012.

3

Developing Extensible Data Security

In any corporation, some users are restricted to work with specific sensitive data because of its confidentiality or company policies, and this type of data access authorization can be managed using **extensible data security (XDS)**. XDS is the evolution of the **record-level security (RLS)** that was available in the previous versions of Microsoft Dynamics AX. Also, Microsoft keeps the RLS in version AX 2012, so you can refer to it at any time.

The topics that will be covered in this chapter are as follows:

- The main concepts of XDS policies
- Designing and developing the XDS policy
- Creating the XDS policy
- Adding constrained tables and views
- Setting the XDS policy context
- Debugging the XDS policy

The main concepts of XDS policies

When developing an XDS policy, you need to be familiar with the following concepts:

Concept	Description
Constrained tables	A <i>constrained table</i> is the table or tables in a given security policy from which data is filtered or secured, based on the associated policy query.
Primary tables	A <i>primary table</i> is used to secure the content of the related constrained table.
Policy queries	A <i>policy query</i> is used to secure the constrained tables specified in a given extensible data security policy.
Policy context	A <i>policy context</i> is a piece of information that controls the circumstances under which a given policy is considered to be applicable. If this context is not set, then the policy, even if enabled, is not enforced.

After understanding the previous concepts of XDS, we move on to the four steps to develop an XDS policy, and they are as follows:

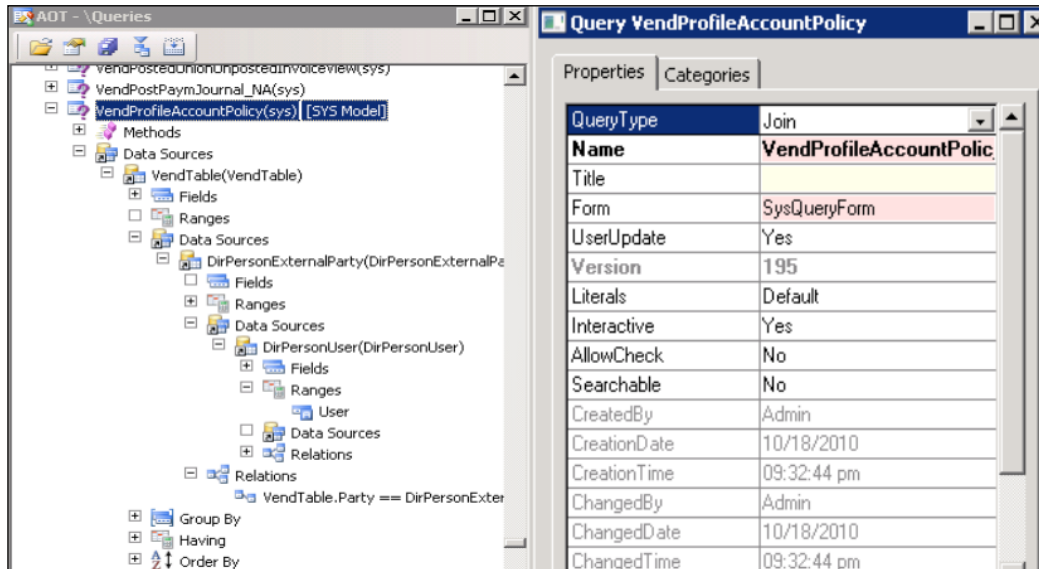
1. Design the query on the primary tables.
2. Develop the policy.
3. Add the constrained tables and views.
4. Set up the policy context.

Designing and developing the XDS policy

XDS is a powerful mechanism that allows us to express and implement data security needs. The following steps show detailed instructions on designing and developing XDS:

1. Determine the primary table; for example, `VendTable`.
2. Create a query under the AOT **Queries** node:
 - Use **VendTable** as the first data source

- Add other data sources as required by the vendor data model:



Creating the policy

Now we have to create the policy itself. Follow these steps:

1. Right-click on AOT and go to **Security | Policies**. Select **New Security Policy**.
2. Adjust the **PrimaryTable** property on the policy to **VendTable**.
3. Settle the **Query** property on the policy to **VendProfileAccountPolicy**.
4. Adjust the **PolicyGroup** property to **Vendor Self Service**.
5. Settle the **ConstrainedTable** property to **Yes** to secure the primary table using this policy.

6. Adjust the **Enabled** property to **Yes** or **No**, depending on whether or not you want to control the policy.
7. Settle the **ContextType** property to one of the following:
 - **ContextString**: Adjust the property to this value if a global context is to be used with the policy. After using **ContextString**, it needs to be set by the application using the XDS::SetContext API.
 - **RoleName**: Adjust the property to this value if the policy should be applied only if a user in a specific role needs to access the constrained tables.
 - **RoleProperty**: Adjust the property to this value if the policy is to be applied only if the user is a member of any one of roles that have the **ContextString** property settled to the same value.

The following screenshot displays the properties:

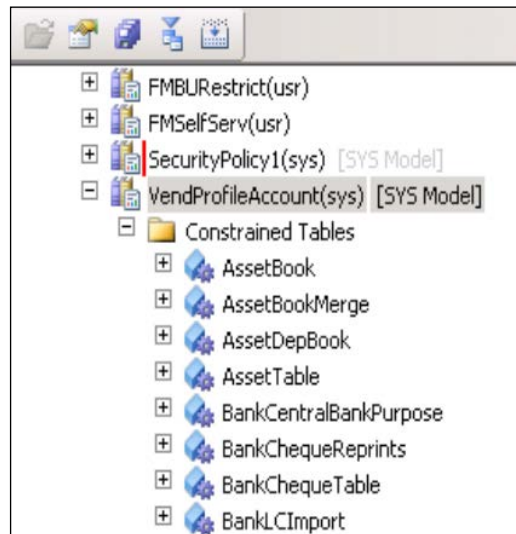


Adding constrained tables and views

After designing the query and developing the required policy, the next step is to add the constrained tables and views that contain the data by using the created policy.

By following the next steps, you will be able to add constrained tables or views:

1. Right-click on the **Constrained Tables** node.
2. Go to **New | Add table** to add a constrained table; for example, the **AssetBook** table, as shown in the following screenshot:



When adding the constrained table **AssetBook**, you must determine the relationship that should be used to join the primary table with the last constrained table.

3. Go to **New | Add View** to add a constrained view to the selected policy.

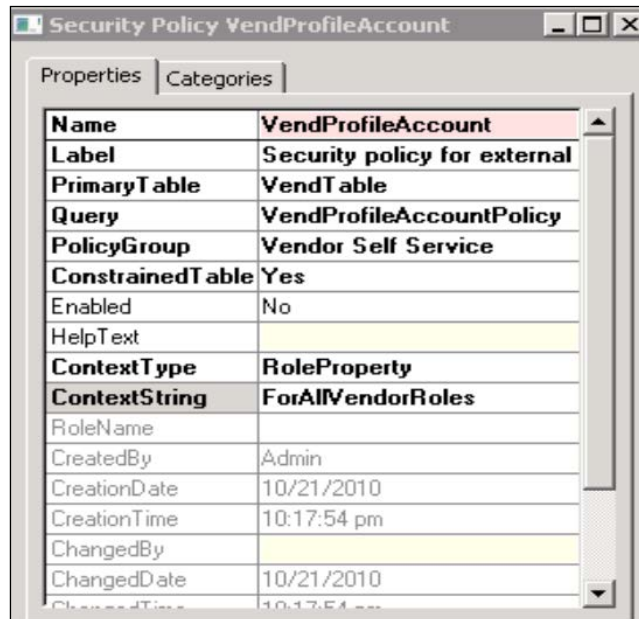
Repeat these steps for every constrained table or view that needs to be secured through this policy.

After finishing these steps, the policy will be applied for all users who are attempting to access the tables or views that are located on the constrained table's node when the **Enabled** property is set to **Yes**. Security policies are not applied to system administrators who are in the `SysAdmin` role.

Setting the XDS policy context

According to the requirements, the security policy needs to be adjusted to apply only to the users who were assigned to the vendor role. The following steps should be performed to make the appropriate adjustment:

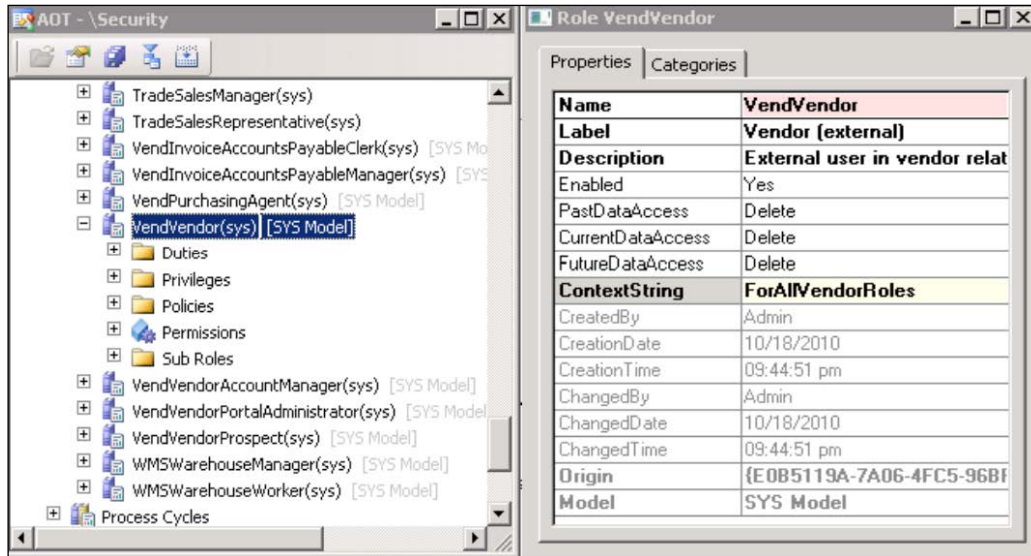
1. Adjust the **ContextType** property on the policy node to **RoleProperty**.
2. Settle the **ContextString** property on the policy node to **ForAllVendorRoles**:



To assign this policy to all the vendor roles, the **ForAllVendorRoles** context should be applied to the appropriate roles:

1. Locate each role that needs to be assigned to this policy on the AOT node; for example, the **VendVendor** role.

- Adjust the **ContextString** property on the **VendVendor** role to **ForAllVendorRoles**:



For more information, go to MSDN and refer to *Whitepapers – Developing Extensible Data Security Policies* at <https://msdn.microsoft.com/en-us/library/bb629286.aspx>.

Debugging XDS policies

One of the most common issues reported when a new XDS policy is deployed is that an unexpected number of rows are being returned from a given constrained table. For example, more sales orders are being returned than expected if the sales order table is being constrained by a given customer group.

XDS provides a method to debug these errors. We will go over it now.

Review the SQL queries that have been generated. The X++ select has been extended with a command that instructs the underlying data access framework to generate the SQL query without actually executing it.

The following job runs a select query on `SalesTable` with a generated command. It then calls the `getSQLStatement()` method on `SalesTable` and dumps the output using the info API.

```
static void VerifySalesQuery(Args _args)
{
    SalesTable salesTable;
    XDSServices xdsServices = new XDSServices();
    xdsServices.setXDSContext(1, '');
    //Only generate SQL statement for custGroup table
    select generateonly forceLiterals CustAccount, DeliveryDate from
    salesTable;
    //Print SQL statement to infolog
    info(salesTable.getSQLStatement());
    xdsServices.setXDSContext(2, '');
}
```

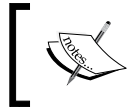
The XDS policy development framework further eases this process of doing some advanced debugging by storing the query in a human-readable form. This query and others on a given constrained table in a policy can be retrieved by using the following Transact-SQL query on the database in the development environment (AXDBDEV in this example):

```
SELECT [PRIMARYTABLEAOTNAME], [QUERYOBJECTAOTNAME],
       [CONSTRAINEDTABLE], [MODELEDQUERYDEBUGINFO],
       [CONTEXTTYPE], [CONTEXTSTRING],
       [ISENABLED], [ISMODELED]
FROM [AXDBDEV].[dbo].[ModelSecPolRuntimeEx]
```

This SQL query generates the following output:

	QUERYOBJECTAOTNAME	CONSTRAINEDTABLE	MODELEDQUERYDEBUGINFO
1	VendProfileAccountPolicy	AssetBook	SELECT * FROM AssetBook(AssetBook_1) EXISTS JOIN 'x' FROM VendTable(VendTabl...
2	VendProfileAccountPolicy	AssetBookMerge	SELECT * FROM AssetBookMerge(AssetBookMerge_1) EXISTS JOIN 'x' FROM VendTa...
3	VendProfileAccountPolicy	AssetDepBook	SELECT * FROM AssetDepBook(AssetDepBook_1) EXISTS JOIN 'x' FROM VendTable(V...
4	VendProfileAccountPolicy	AssetTable	SELECT * FROM AssetTable(AssetTable_1) EXISTS JOIN 'x' FROM VendTable(VendTa...
5	VendProfileAccountPolicy	BankCentralBankPurpose	SELECT * FROM BankCentralBankPurpose(BankCentralBankPurpose_1) EXISTS JOIN '...
6	VendProfileAccountPolicy	BankChequeReprints	SELECT * FROM BankChequeReprints(BankChequeReprints_1) EXISTS JOIN 'x' FROM ...
7	VendProfileAccountPolicy	BankChequeTable	SELECT * FROM BankChequeTable(BankChequeTable_1) EXISTS JOIN 'x' FROM Vend...

As you can see, the query that will join the `WHERE` statement of any query to the **AssetBook** table will be ready for debugging. Other metadata, such as `LayerId`, can be debugged if needed.



When multiple policies apply to a table, the results of the policies are linked together with `AND` operators.

Summary

By the end of this chapter, you are able to secure your sensitive data using the XDS features. We learned how to design and develop XDS policies, constrained tables and views, primary tables, policy queries, set the security context, run SQL queries and learned how to debug XDS policies.

In the next chapter, we are going to learn the organizational model framework, the organization model scenarios and how to extend and secure the organizational model.

4

Extending the Organization Model

To understand what extending the organization model means, we have to be aware of something called **business growth** and **business continuous improvement**.

International organizations can grow internally and externally. External growth occurs by mergers or acquisition, and that's what happened when Microsoft bought Skype with an investment of 8.5 billion dollars. Another international organization decided to target a new customer segment or create sales growth in one of its products in the next year with a certain number of sales amount and units, which is called **internal business growth**.

These two types of growth both contain a high level of risk and challenges; therefore, the CEOs need to measure the performance of these types of growth opportunities (internal or external).

They also need to monitor and control the performance improvement that happens in the lower level of the organization, such as departments, business units, and divisions.

Microsoft Dynamics AX 2012 can handle this type of growth and monitor the performance of their investments using the extended organizational model.

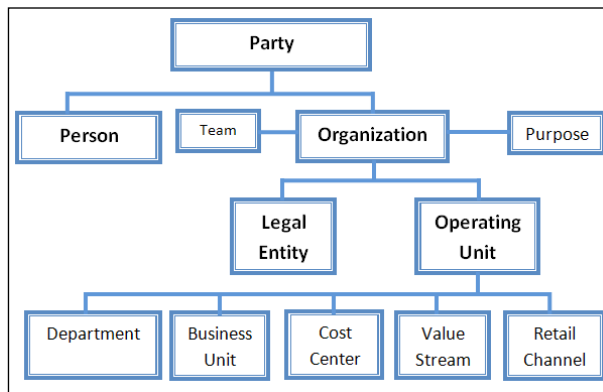
The list of topics that will be covered in this chapter is as follows:

- The organizational model framework
- The organizational model scenarios
- Extending the organizational model

The organizational model framework

The organizational model has two main components: **organization types** and **organization hierarchies**. To start modeling your business, you have to understand each one of them to be able to establish your organizational structure.

The following diagram shows the complete components of the organizational model framework:



Party records: Some organizations may have multiple legal entities or locations; therefore, you can create party records in the Global Address Book form to separate these entities or locations.

Creating separate entities using party records reduces the confusion when communicating with different areas in an organization.

The organization types in Microsoft Dynamics AX 2012, legal entities and operating unit are as follows:

- **Legal entities:** In the business world, legal entities can be described as any entity type that has a legal standing in the eyes of the law that can enter into agreements or contracts. In Microsoft Dynamics AX 2012, the legal entity is the same as a company.
- **Operating unit:** Any legal entity can divide their work into operational processes among the team, and every team is considered an operating unit that has duties and responsibilities to perform their work. Legal entities must work at least with a single operating unit or more according to their business capacity. Therefore, there are several types of operating units and they are as follows:
 - **Business unit:** This is an operating unit that focuses on achieving a business strategy, such as the product line or a specific industry

- **Cost centre:** This is a type of operating unit that is used to track and manage costs, and some types of operating units can be used as a cost controller
- **Department:** The created organizational departments are based on functional responsibility, such as accounting, sales, marketing, and customer service
- **Value stream:** This is a type of operating unit that is used in lean manufacturing that describes the flow of activities needed to create a final product or service
- **Retail channel:** This is a type of operating unit that is used in the retail industry to represent a retail channel, such as retail stores and online stores



The process of internal control in any organization can be enabled through the organization model framework in Microsoft Dynamics AX 2012, by arranging the legal entities and operating units into hierarchies and using the reporting tool.

Organization hierarchies

The organization is a group of people who are working in an operational and administration process that has a flow of different activities. The organizational hierarchy shows the internal and external relationships in the organization; therefore, it is easy to understand the flow of activities within the organization.



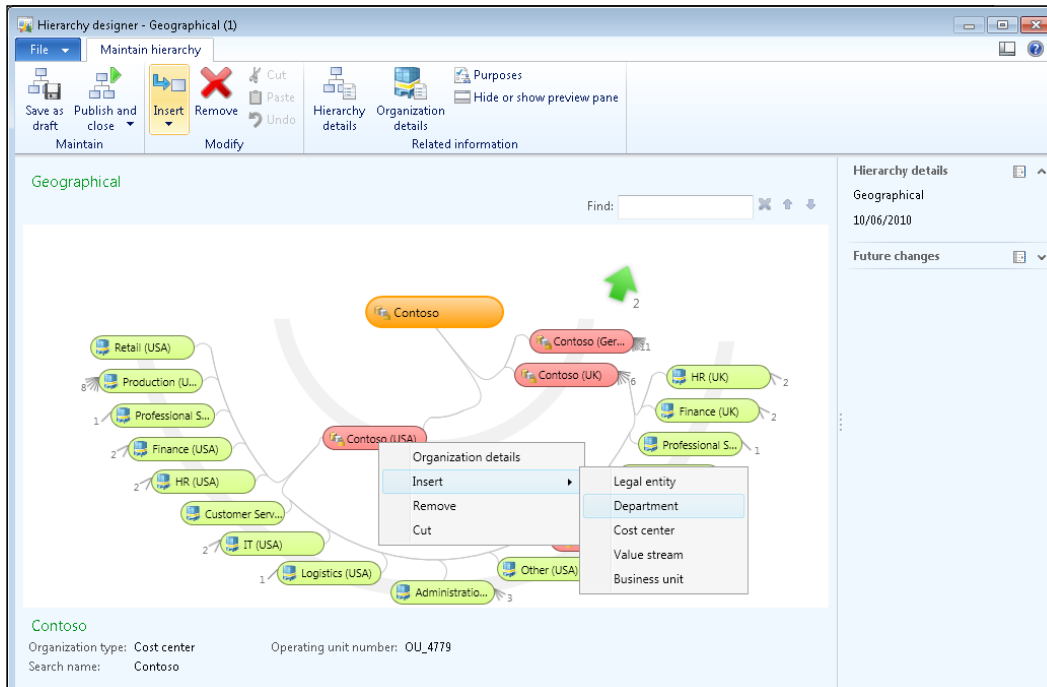
The team is a type of internal organization that can be found under several types of operating units that can be created for a specific purpose or project.

A **purpose** is set to define how the organizational hierarchy is used in organization model scenarios; some purposes can be used with a specific organization type and others can be used with all types of organizations, as shown in the following table:

Purpose	Description	Organization types
Organization chart	Used in human resource module for the reporting relationship	All types
Security	Used to determine data security access for organizations	All types
Audit internal control	Used to determine policies to identify documents for audit purposes	Legal entities

Purpose	Description	Organization types
Centralized payments	Used to make payments by one legal entity on behalf of other legal entities	Legal entities
Vendor payment internal control	Used to determine payments of vendor invoices	Legal entities
Signature authority internal control	Used to determine policies that control the spending and approval limits	All types
Procurement internal control	Used to determine policies that monitor and control the purchasing process cycle	All types
Expenditure internal control	Used to determine policies for expense reports	All types

The organization hierarchy purposes are shown in the following screenshot:



As a best practice to implement Microsoft Dynamics AX 2012, the organization model structure must be defined. Next, every functional department should participate in establishing the organizational hierarchy under the supervision of the CEO, managers, and senior executives.

The organizational model scenarios

The organizational framework is used to model how the business operates, and there are two ways to approach the organization model framework. The first method is using the built-in integration with Microsoft Dynamics AX modules and other application frameworks. The second method is modeling custom scenarios to meet the business needs and requirements:

- Integration with other frameworks' application modules
- Custom modeling scenarios

Integration with other frameworks' application modules

The main modules in Microsoft Dynamics AX that are integrated with the organizational model framework are as follows:

- **Address book:** The address book is a main feature in Microsoft Dynamics AX that stores addresses and contact information; therefore, all organization types can use the address book in an internal organization
- **Financial dimensions:** The main purpose of the financial dimension is to measure the financial performance of the organization, and it can be used with legal entities as well as operating units to evaluate the business performance for every department, business unit, and department
- **Policy framework:** The policy framework can define the internal control policy of the organization, for example, expense reports, purchase requisitions, and vendor invoice payments
- **Extensible data security:** The extensible data security framework helps secure your business data from unauthorized access and can be assigned on the organization hierarchy

Also, the organizational model framework is used in the following modules within Microsoft Dynamics AX:

- **Human resource:** All the data and transactions that occur in the human resources module can be shown and modified using the organizational model framework

- **Procurement and sourcing:** Using the organizational model framework, the purchase data can be viewed and modified similarly to the purchase requisition and the packing list
- **Travel and expenses:** Expenses that occur in a legal entity can be viewed, modified, and audited using the organizational model framework

Custom modeling scenarios

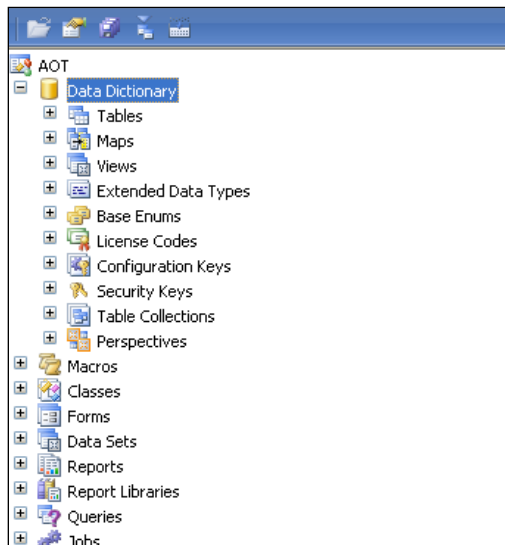
You can also model your own scenarios using the organization model framework. For example, organizations may need to limit the user access to certain data in the sales module that is related to another industry line and view only the sales data in his or her specific industry.

Extending the organizational model

You can extend the organizational model to support ISV or partner customization by creating a custom type of operating unit or extending the hierarchy designer.

Creating a custom type of operating unit

You can implement this scenario when the organization needs to accommodate a certain vertical industry, such as retail branches and stores. The organizational model framework supports the extensibility of your organization by creating a new custom type of operating unit using the AOT, and it is as follows:



The steps to create a custom type of operating unit are as follows:

1. Create a new base enum value for the operating unit type.
2. Create a view.
3. Create a menu item.

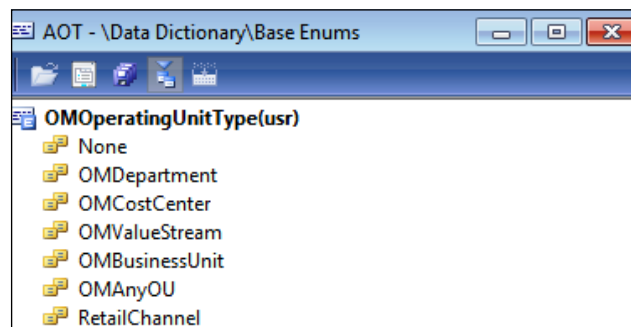
Creating a new base enum value

Base enums are used to categorize data; for example, if you need to add more categorization to inventory item types, you can create a new base enum to add more categories in the same table.

System enums are located in the AOT in this path: **SystemDocumentation | Enums**. The value of a base enum is stored in the database as an integer value.

The following steps illustrate how to create a new base enum value using the AOT:

1. Navigate to **Data Dictionary | Base Enums | OMOperatingUnitType**, and then click on **New Element**:



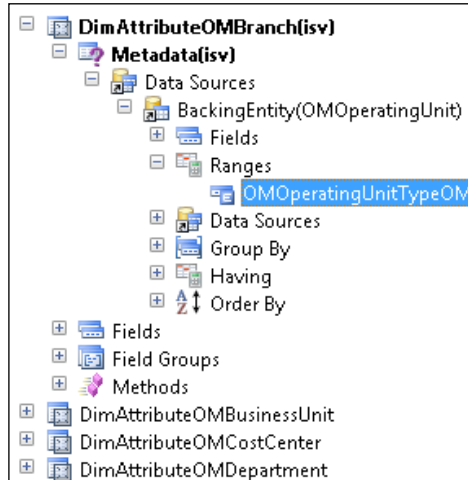
2. Enter the name and label from the **Properties** window of the new element.

Creating a view

After creating the new base enum value, the next step is to create a view from the AOT view node as follows:

1. Navigate to **Data Dictionary | Views** and then click on **New View**.

2. In **Name**, type `DimAttributeOM"Name"`, for example, `DimAttributeOMStore`:



Creating a menu item

The last step is to create a menu item for the new operating unit type, and it is created as follows:

1. Navigate to **Menu Items | Display** and then click on **New Menu Item**.
2. In the **Properties** window, set the following (**Store** is an example):
 - **Name** to **StoreMenuItem**
 - **Label** to **Store**
 - **Object** to **OMOperatingUnit**
 - **EnumTypeParameter** to **OMOperatingUnitType**
 - **EnumParameter** to **Store**

After finishing the last step, the new operating unit type, **Store**, will appear in the list of the operating unit types in the organizational model framework.

You can define the data security access and policies for these organization types based on the hierarchy.

Extending the hierarchy designer

There is limited support to extend the hierarchy designer in Microsoft Dynamics AX 2012. The hierarchy designer can be customized into four parameters within the hierarchy, and they are as follows:

- The border color
- The node image
- The top gradient color
- The bottom gradient color

This form is available through this path: **Organization Administration | Setup | Organization | Organization Hierarchies**.

Summary

By the end of this chapter, you understood the types of organizations and the basic categories of operational units, and learned how to use them.

You were also able to extend your organizational hierarchy using the organization model framework and create a new custom type of operating units when it comes to establishing new vertical industries for the growth purpose of your organization.

In the next chapter, we will learn how to be the architect of an Enterprise Portal, how to secure it, and how to secure your business data.

5

Enterprise Portal Security

An organization can expand the use of Microsoft Dynamics AX 2012 to reach their customers, vendors, employees, and business partners and enable them to access the application through the Enterprise Portal web client.

The Microsoft Dynamics AX 2012 web portal can be accessed from anywhere using a web browser to access data, reports, transactions, documents, and alerts. Also, it can be used as a web platform that contains default web pages and user roles that can be used or customized to meet your business needs.

The Enterprise Portal is a powerful tool that brings the best of Microsoft SharePoint, ASP.NET technologies to establish a web-based application that contains the best functionality in both technologies in Microsoft Dynamics AX Enterprise Portal.

The list of topics that will be covered in this chapter is as follows:

- The architect of Enterprise Portal
- Security in Enterprise Portal
- Data access security
- Report access security

The architecture of Enterprise Portal

To understand how to secure web elements, we need to identify the Enterprise Portal elements and components that make up an **Enterprise Portal** page, and they are as follows:

- Web parts
- AOT elements
- Datasets
- Controls

Web parts

Web parts support customization and personalization. They can be integrated into a web page.

Web parts contain a lot of objects, such as the action pane, business overview, connect, cues, infolog, left navigation, list, page title, quick launch, quick links, report, toolbar, unified work list, and user control.

The following table shows a description for a group of these web parts.

Objects	Description
Action pane	This is used to display the action pane and it also points to a web menu in the AOT. <i>AxActionPane</i> control in a web control does the same job.
Business overview	This is used to display the Key Performance Indicators (KPIs) and analytical data in role centers.
Connect	This is used to display the links to the information located in the Microsoft Dynamics AX community website.
Cues	This is used to display numeric information, and it is added to role center pages and points to a cue group in the AOT.
Report	This is used to display Microsoft SSRS reports (SQL Server Reporting Services).

AOT elements

The AOT has several elements that are used with Enterprise Portal, and they are available in *Chapter 1, MorphX Security System*. These elements include:

- **Tables:** Tables are located in AOT | **Data Dictionary** | **Tables**. They store the business data and have a corresponding table in Microsoft SQL Server database.
- **Views:** Views are located in AOT | **Data Dictionary** | **Views**. They are X++ SQL statements that are reused in other X++ SQL statements to view the business data from different tables.
- **Classes:** They are listed in **System Documentation** | **Classes**. They contain the source code X++ for the application classes.
- **Forms:** This is a dialog box in the user interface that is used to access the database.

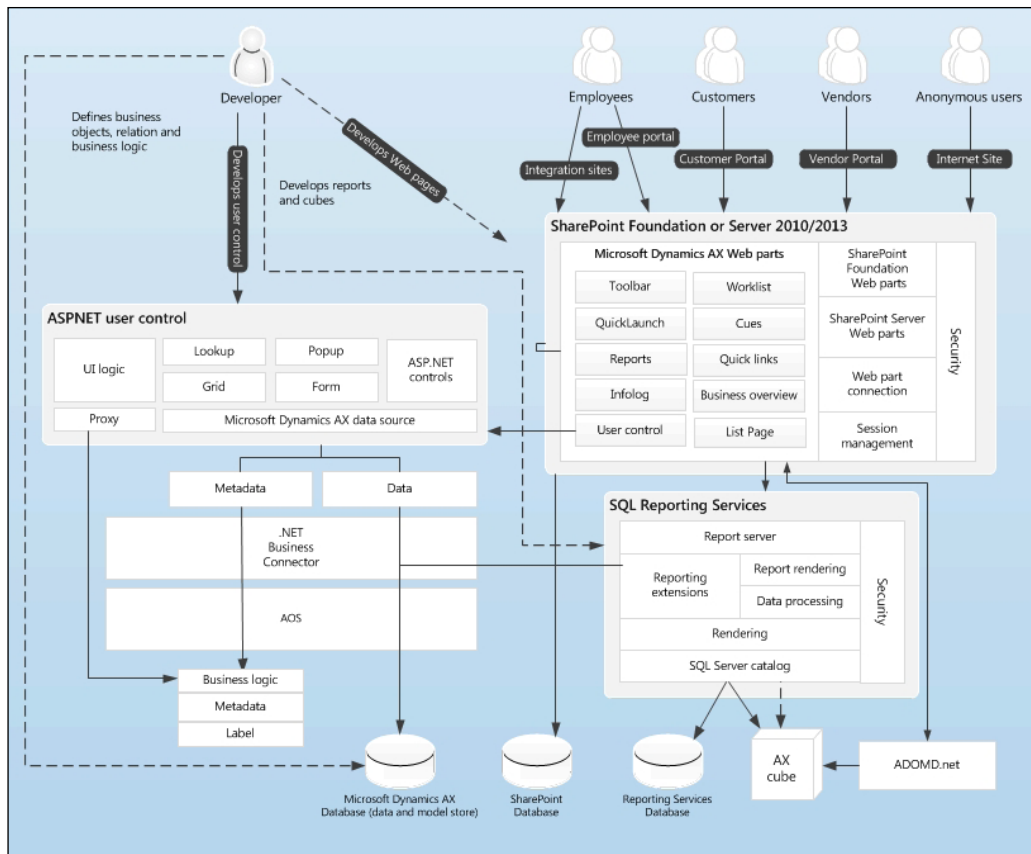
Datasets

To access any business data in Microsoft Dynamics AX using the Enterprise Portal, you have to create the dataset. Datasets can be created using MorphX. Datasets contain a collection of data that is presented in a grid table to be viewed in the web portal.

Controls

The Microsoft Dynamics AX web portal contains a set of controls that can be used to access, display, and edit business data. The most common controls are AxDataSource, AxForm, AxMultiSection, AxSection, AxMultiColumn, AxColumn, AxGridView, and AxLookup.

The following figure illustrates the high-level overview of the Enterprise Portal architect (Microsoft TechNet: <http://technet.microsoft.com/en-us/library/dd362005.aspx>):



Security in Enterprise Portal

The security in Microsoft Dynamics AX Enterprise Portal depends on the underlying technologies, such as SharePoint and **Internet Information Services (IIS)**. There are two types of web portal users: the public user and the dynamics user. The public user allows for the viewing of products, requests for products, creation of account, and so on.

The public users have an anonymous authentication, so it is available for anyone who uses the Internet. Therefore, there is a built-in guest user account that is a part of the Enterprise Portal that's connected to Microsoft Dynamics AX components with a limited access that is necessary for the website to function in a proper manner and also for security reasons.

The dynamics users are authenticated employees, customers, and vendors. They have a complete portal to make transactions and view reports and charts, and they are referred to the security policies on Microsoft Dynamics Enterprise Portal.

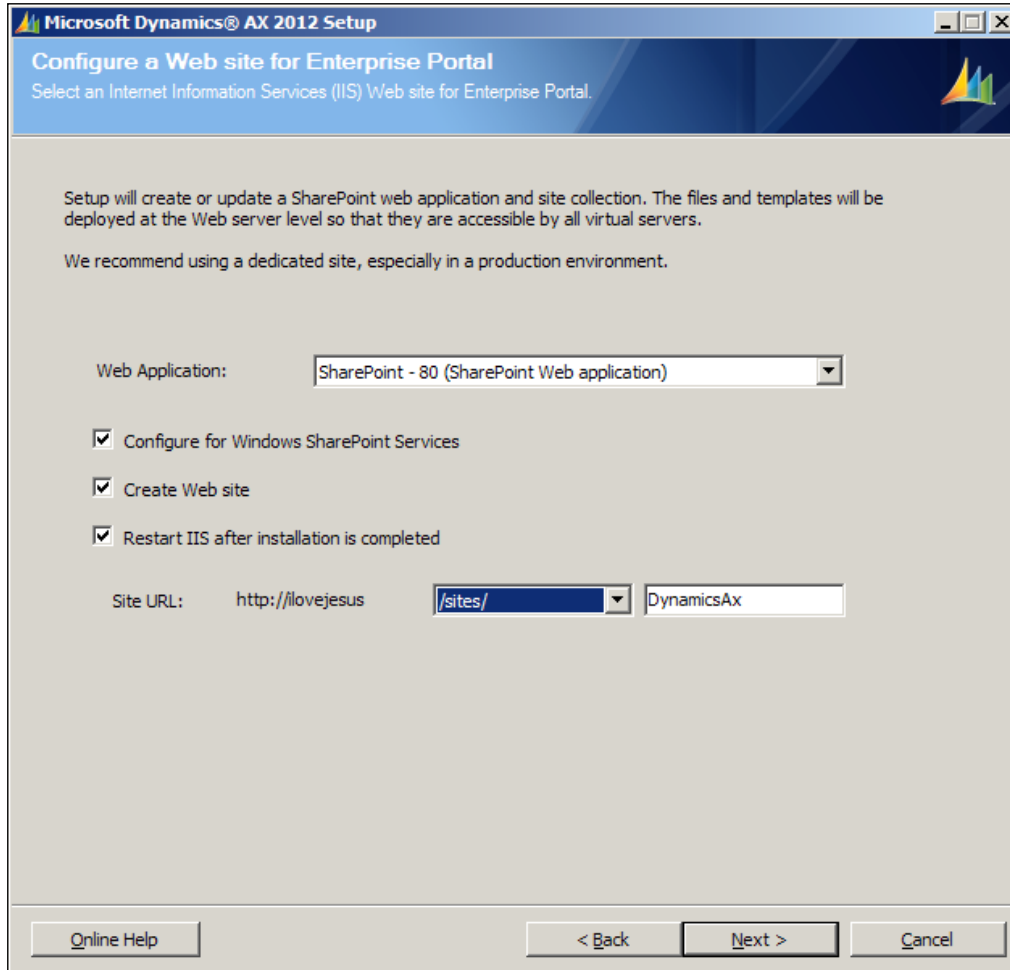
Securing web elements

It is a best practice to secure both the web menu item (**Web | Web Menu Items | URLs**) and the managed web content (**Web | Web Content | Managed**). If you only secure the web menu item, the user can still access the managed web content. You can use privileges to secure these elements or actions (**Web | Web Menu Items | Actions**) as entry points for privileges to control users from accessing them.

If a user doesn't have access to a web menu item, this item doesn't appear on the user's web menu.

If a link in the web menu item appears in other web user controls that the user has access to, the item linked with the web menu item shows as text rather than a link.

The following screenshot shows how to secure the web elements:

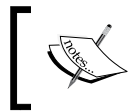


Record context and encryption

The Enterprise Portal uses the record context to locate a record in the database and display it in a web form, so it will be easy to view and edit the displayed data. The record context works as an interface to pass through the information from the query to a web part page and retrieve a record from Microsoft Dynamics AX.

The query strings that are used to pass the record context to a web part page as follows:

- **WTID:** This equals the table ID
- **WREC:** This equals the record ID
- **WKEY:** This equals the unique record key (the value of the field of the record to be retrieved)



To secure the Enterprise Portal, use a hash parameter; this parameter ensures that a URL created by one user can't be used by another one.

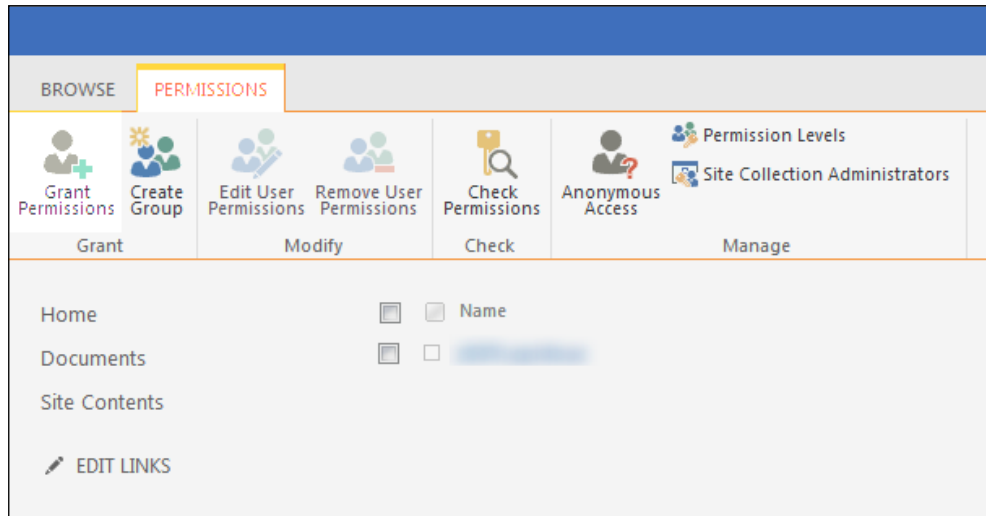
The use of the hash parameter on the Enterprise Portal is recommended to keep the encryption turned on. For debugging purposes, you can turn off the encryption for better performance by navigating to **System Administration | Setup | Enterprise Portal | Web Sites**.

Data access security

The Enterprise Portal in Microsoft Dynamics AX 2012 enables the administrators to grant users (public or dynamic) access to the web portal to view or edit the business data. This can be done by first adding users to the SharePoint before accessing the web portal, as follows:

1. Start the Enterprise Portal site in a web browser. The URL is `http://server_name/sites/Dynamicsax`.
2. Navigate to **Site Actions | Site Permissions**.
3. Click on **Grant Permissions**.
4. From the **Users/Groups** textbox, enter the name of each user and then click on **Check Names**.
5. Click on the permission level that you want to set up (**Read Permission**, **Contribute Permission**, or **Design Permission**).

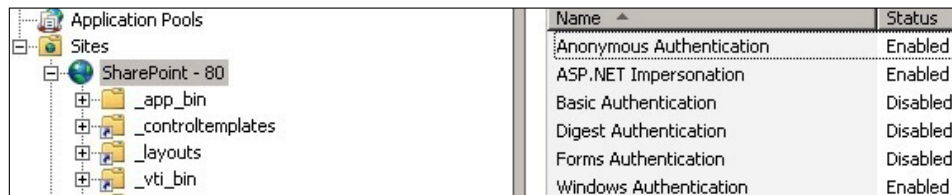
6. Click on the **OK** button. The output is as follows:



Now, the internal users with role centers can access the Enterprise Portal and view the content in their role centers. According to the security role in Microsoft Dynamics AX and the permission granted to the users in SharePoint, the web page and its content is displayed according to this security enablement.

To grant public users access to the public site, you have to follow these steps:

1. Create the public site (shown in the second figure in the chapter).
2. Enable **Anonymous Authentication** on the public site:



It is recommended that you enable **Anonymous Authentication** in IIS.

3. Assign the guest user account to the guest security role.



The environment of the Enterprise Portal solution must be secured by performing security tasks for the following:

- The web server security setting
- The client-server communication using **Secure Sockets Layer (SSL)**
- The IIS setting
- The SharePoint security setting

Report access security

There are two procedures that grant a user access to reports, whether you are using Microsoft SQL Server Reporting Services in the native mode or the SharePoint integrated mode that is available only in Microsoft Dynamics AX 2012 R2 and R3:

- Assign users to the DynamicsAXBrowser role
- Grant them access permission to view reports

Assigning a user to the DynamicsAXBrowser role

You must assign users or groups to the DynamicsAXBrowser role in the report manager by performing the following steps:

1. Click on the **DynamicsAX** folder.
2. Click on **Folder Settings**.
3. Click on **Security**.
4. Click on **New Role Assignment**.
5. Enter the Active Directory username or group to assign to the DynamicsAXBrowser role.
6. Select the **DynamicsAXBrowser** role.
7. Click on the **OK** button.

Granting a user access permission to view reports

After finishing the first step, you need to grant the users or groups access to view reports in SharePoint by enabling read permission on the site. The following steps illustrate how to grant read permission to users:

1. Open the SharePoint site that contains the document library that stores the reports.
2. Navigate to **Site Actions | Site Permissions**.
3. Click on **Grant Permissions**.
4. Enter the Active Directory names of the users whose reports you want to view from the **Users/Groups** field.
5. Select the **Grant users permission directly** option from the **Grant Permissions** area.
6. Select the **Read** checkbox.
7. Select the **Design** checkbox if you want the users to be able to filter reports (optional).
8. Click on the **OK** button.

Summary

By the end of this chapter, you understood the architecture of the Enterprise Portal in Microsoft Dynamics AX 2012, and you were able to secure web parts and elements. Also, you learned how to grant users permission access to the web portal to view the web content and reports. Securing your web portal is a key to successfully implementing the Enterprise Portal in Microsoft Dynamics AX product.

At the end of this book, we suggest that read the chapters one by one again, and after finishing every chapter, you practice it well and then move to the other chapter.

You will be able to develop a security artifact using AOT, code access security, extensible data security policies, and debugging XDS policies.

Index

A

AOT elements, Enterprise Portal

- about 68
- classes 68
- forms 68
- tables 68
- views 68

Application Object Tree (AOT) 5, 6

application security

- application file server 3
- Application Object Server (AOS) 3
- database server 3
- Enterprise Portal 3
- features 3

auto-inference 18

B

best practice tool 15

Business Connector 3

business continuous improvement 57

business growth 57

C

Code Access Security (CAS)

- about 25
- API, securing on AOS 28, 29
- using 26-28

common intermediate language (CIL) 9

controls 69

cross-reference tool 13

custom modeling scenarios 62

D

data access security 72, 73

datasets 69

debugger

- about 9, 30
- enabling 31
- installing 30, 31
- shortcut keys 37
- user interface 36, 37
- users, adding to Debugging User
 - local group 33-35

display method

- creating 38, 39

Drag and Drop feature 10

E

encryption 72

Enterprise Portal

- about 67
 - AOT elements 68
 - architecture 67
 - controls 69
 - datasets 69
 - reference 69
 - security 70
 - web parts 68
- ### **entity relationship diagrams (ERDs) 4**
- ### **extensible data security (XDS) 47**

F

Find tool 13, 14

I

internal business growth 57
Internet Information Services (IIS) 3, 70

K

Key Performance Indicators (KPIs) 68

L

legal entities 58

M

Microsoft Dynamics AX 2012
organization types 58
Microsoft Dynamics AX system architecture
about 2
application security 3
infrastructure security 2
Microsoft SSRS reports (SQL Server Reporting Services) 68
modules, Microsoft Dynamics AX
address book 61
extensible data security 61
financial dimensions 61
human resource 61
policy framework 61
procurement and sourcing 62
travel and expenses 62
MorphX development tool
about 3, 4
Application Object Tree 5, 6
best practice tool 15
cross-reference tool 13
debugger 9
Find tool 13, 14
projects 10, 11
property sheet 11
reverse engineering tool 16
table browser tool 14, 15
X++ code editor 6-8
X++ compiler 8
MorphX tools 4

O

operating unit
about 58
business unit 58
cost centre 59
department 59
retail channel 59
value stream 59
organizational model
about 58
components 58
organization hierarchies 58
organization types 58
scenarios 61
organizational model, extending
about 62
custom type of operating unit,
creating 62, 63
hierarchy designer, extending 65
menu item, creating 64
new base enum value, creating 63
view, creating 63
organization hierarchies
about 59, 60
purposes 59, 60
organization types, Microsoft Dynamics AX 2012
legal entities 58
operating unit 58

P

parm 8
projects 10, 11
property sheet 11

R

record context 72
record-level security (RLS) 47
report access security
about 74
read permission, granting to users 75
user, assigning to DynamicsAXBrowser
role 74
reverse engineering tool 16

S

scenarios, organizational model

- about 61
- custom modeling scenarios 62
- integration, with other frameworks'
 - application modules 61

Secure Sockets Layer (SSL) 3

security artifact

- about 17
- configuration key, applying 22, 23
- developing, AOT used 17
- duties 17
- permissions 17
- permissions, assigning to privileges 19-22
- permissions, setting for form 18
- policies 17
- privileges 17
- security privilege, testing 22
- security privilege, validating 22
- security role 17

security coding

- X++, using 25

security debugging

- about 30
- debugger shortcut keys 37
- debugger tool, enabling 31
- debugger tool, installing 30, 31
- debugger user interface 36, 37
- users, adding to Debugging User
 - local group 33-35

security, Enterprise Portal

- about 70
- data access security 72, 73
- encryption 72
- record context 72
- report access security 74
- web elements, securing 70

security, for display method 38

T

table browser tool 14, 15

Table Permissions Framework (TPF)

- about 41
- enabling, on database table 42-44

U

Unified Modeling Language (UML)

- element 4

W

web parts, Enterprise Portal 68

WKEY 72

WREC 72

WTID 72

X

X++

- about 25
- using 25

X++ code editor

- about 6-8
- shortcut keys 7

X++ compiler 8

XDS policy

- constrained table 48
- constrained tables, adding 51
- context, setting 52
- creating 49, 50
- debugging 53, 54
- designing 48
- developing 48
- main concepts 48
- policy context 48
- policy query 48
- primary table 48
- views, adding 51



Thank you for buying Microsoft Dynamics AX 2012 R3 Security

About Packt Publishing

Packt, pronounced 'packed', published its first book, *Mastering phpMyAdmin for Effective MySQL Management*, in April 2004, and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern yet unique publishing company that focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website at www.packtpub.com.

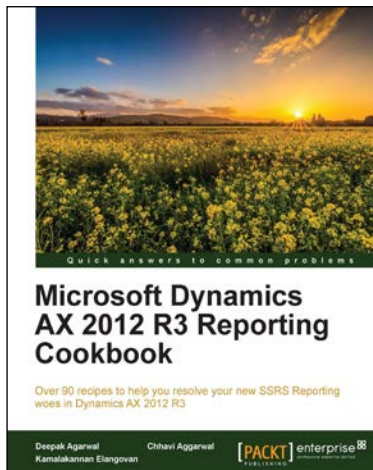
About Packt Enterprise

In 2010, Packt launched two new brands, Packt Enterprise and Packt Open Source, in order to continue its focus on specialization. This book is part of the Packt Enterprise brand, home to books published on enterprise software – software created by major vendors, including (but not limited to) IBM, Microsoft, and Oracle, often for use in other corporations. Its titles will offer information relevant to a range of users of this software, including administrators, developers, architects, and end users.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, then please contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

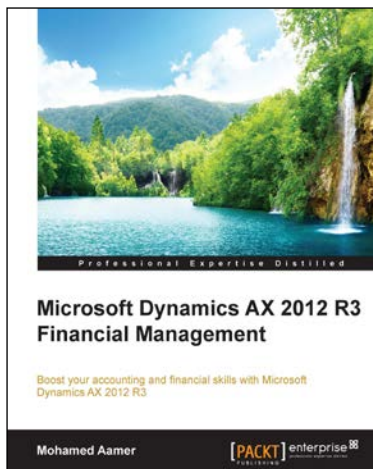


Microsoft Dynamics AX 2012 R3 Reporting Cookbook

ISBN: 978-1-78439-538-4 Paperback: 352 pages

Over 90 recipes to help you resolve your new SSRS Reporting woes in Dynamics AX 2012 R3

1. Easy and effortless deployment of SSRS reports.
2. One stop solution for developers to customize existing SSRS reports in Dynamics AX R3.
3. Step-by-step tutorial with solutions to writing unit classes for reports.



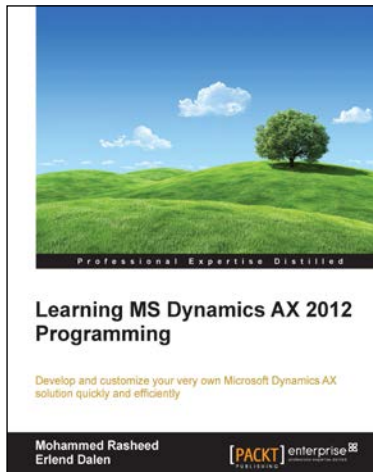
Microsoft Dynamics AX 2012 R3 Financial Management

ISBN: 978-1-78439-098-3 Paperback: 352 pages

Boost your accounting and financial skills with Microsoft Dynamics AX 2012 R3

1. Understand Microsoft Dynamics AX financial management and successfully configure and set up your software.
2. Manage the AX 2012 R3 financial module with the help of highly useful tips and tricks.
3. Administer customer relations and plan enterprise resources with this systematic guide.

Please check www.PacktPub.com for information on our titles

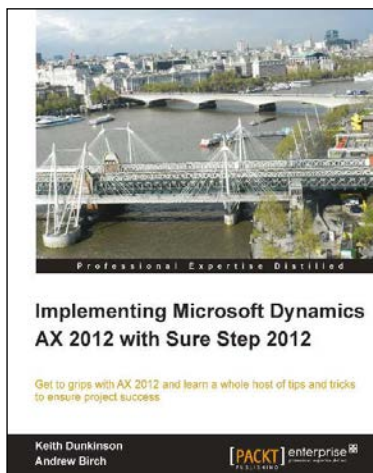


Learning MS Dynamics AX 2012 Programming

ISBN: 978-1-78217-126-3 Paperback: 370 pages

Develop and customize your very own Microsoft Dynamics AX solution quickly and efficiently

1. Structured learning for new developers and technical consultants.
2. Concise and easy-to-follow walkthroughs of X++ code.
3. Examples and key tips on how to avoid potential pitfalls.



Implementing Microsoft Dynamics AX 2012 with Sure Step 2012

ISBN: 978-1-84968-704-1 Paperback: 234 pages

Get to grips with AX 2012 and learn a whole host of tips and tricks to ensure project success

1. Get the confidence to implement AX 2012 projects effectively using the Sure Step 2012 Methodology.
2. Packed with practical real-world examples as well as helpful diagrams and images that make learning easier for you.
3. Dive deep into AX 2012 to learn key technical concepts to implement and manage a project.

Please check www.PacktPub.com for information on our titles