Quick answers to common problems

Microsoft Windows Identity Foundation Cookbook

Over 30 recipes to master claims-based identity and access control in .NET applications, using Windows Identity Foundation, Active Directory Federation Services, and Azure Access Control Services

Foreword by Rick G. Garibay,
General Manager, CSD Practice Neudesic, Microsoft MVP,
Connected Systems Developer

Sandeep Chanda

[PACKT] PUBLISHING enterprise professional expertise distilled

Microsoft Windows Identity Foundation Cookbook

Over 30 recipes to master claims-based identity and access control in .NET applications, using Windows Identity Foundation, Active Directory Federation Services, and Azure Access Control Services

Sandeep Chanda

[PACKT] PUBLISHING enterprise professional expertise distilled

BIRMINGHAM - MUMBAI

Microsoft Windows Identity Foundation Cookbook

First published: April 2012

Production Reference: 1170412

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

ISBN 978-1-84968-620-4

www.packtpub.com

Cover Image by David Gutierrez (bilbaorocker@yahoo.co.uk)

Credits

Author
Sandeep Chanda

Reviewers
Senthil Kumar
Pushpendra Singh

Acquisition Editor
Rukshana Khambatta

Development Editor
Shreerang Deshpande

Technical Editors
Devdutt Kulkarni
Vrinda Amberkar

Project Coordinator
Michelle Quadros

Proofreaders
Clyde Jenkins
Linda Morris

Indexer
Hemangini Bari

Graphics
Valentina D'Silva
Manu Joseph

Production Coordinator
Arvindkumar Gupta

Cover Work
Arvindkumar Gupta

Foreword

I still remember sitting down with my brand new copy of *Writing Secure Code* by Michael Howard and David LeBlanc. Having moved beyond writing relatively simple intranet web reports, (before the term "BI" came to embody what at the time we thought was an incredibly innovative way to display call center metrics for managing credit card operations) I found myself in a development lead position responsible for building a web portal for managing the collections process for JP Morgan Chase's auto and home business. The portal interfaced with a number of internal assets, such as SQL Server, Oracle, and IBM Mainframes via Terminal 3270 emulation, as well as external partners, such as Experian and Equifax.

In addition to the learning curve of moving from Classic Active Server pages to production-worthy .NET Framework 1.1 and ASP.NET Web Services, we were just beginning to dramatically disrupt the enterprise as a way to minimize the friction between systems while increasing the reusability of these integration investments. As a fledgling new lead, building the portal to stop world hunger and to cure cancer (as all the intranet portals promised to do in those days), I was keenly aware that the solution had to be secure, because after all, "All Input Is Evil", and working in the financial services industry, no security breach or personal information leak goes unpunished, no matter how trivial.

For weeks I skimmed through the 600 page volume, incrementally building confidence that I was doing my due diligence in implementing a trusted subsystem, identifying and authenticating my users, applying the least privilege, and preventing the SQL injection attacks.

Things were significantly simpler in 2003. All of my users were in Active Directory, and as long as I didn't need them to do multiple hops, NTLM was just fine, thank you very much. I put a lot of thought into the roles and proudly remember showing my manager how the new users would automatically have access to the portal as soon as their account was created (provided IT assigned them to the right group! ☺).

Well, it turns out this "Web Services" thing was real, and what they did for the enterprise a decade ago pales in comparison to how service orientation has transformed the way users expect to be able to interact with software today. The proliferation of modern web applications and mobility demand a completely new perspective when designing modern applications. Whether you are building Web, desktop, or mobile solutions that reside on-premise, on the cloud, or are a hybrid thereof, identity and access control have never been more important.

Whether in the enterprise or consumer space, today's users demand access to your application from anywhere and at any time. And, for your applications to compete in the market and provide real value, they must compose a variety of assets, that is public and private, each of which carry their own requirements for authentication and authorization. In short, modern applications must be **claims-aware**.

While the options for federating identity and access control across the public and corporate assets are both varied and daunting, they also present the tremendous opportunities for unlocking the potential of your applications in taking advantage of the existing investments at a global scale. To enable this new breed of applications, Microsoft provides the **Windows Identity Framework** (**WIF**), which aims to simplify working with claims-based security by providing standardized APIs, templates, and tools that make the process of accessing, interpreting, and mapping claims tenable.

Initially provided as a standalone framework (previously known as Geneva), WIF is now included as a part of .NET 4.5, which is in beta at the time of writing this book. The inclusion of WIF in .NET is not merely a packaging decision, but a clear reflection of the commitment that Microsoft has made to this powerful security framework.

As such, Sandeep's book couldn't come at a better time.

Careful to begin with easy-to-grasp fundamentals of claims-based security, Sandeep progresses through the common WIF programming tasks using examples in ASP.NET and WCF familiar to the most .NET developers, while covering bleeding-edge scenarios including new features exposed in Windows 8 and securing Windows Metro applications.

This book offers a combination of simple, intermediate, and advanced scenarios, covering AD FS 2.0 and incorporating web identity providers such as Windows Live ID, Google, Yahoo!, and Facebook with Azure Service Bus Access Control Service. Also covered are the real-world scenarios that you are likely to encounter for securing Microsoft SharePoint, SalesForce.com, and Microsoft Dynamics CRM.

In addition to providing a hands-on pragmatic reference that will be immediately valuable to your next project, this book is a reflection of Sandeep's real-world experience, successfully applying these concepts and techniques in the field, the value of which is worth the price of this book alone.

If you are serious about building claims/identity-aware services and the applications on .NET Framework, and want to get started today, this book belongs in your library.

Rick G. Garibay
General Manager, CSD Practice Neudesic
Microsoft MVP, Connected Systems Developer

About the Author

Sandeep Chanda is a Director of Solutions at Neudesic, a Microsoft National Systems Integrator and Gold Certified Partner. He has been working on several Microsoft Technologies (including but not limited to .NET, BizTalk, SharePoint, and Dynamics CRM) for the past seven years, of which the last couple of years were spent on building claims-aware applications for leading companies in the Manufacturing and Hospitality domains. He is a technology enthusiast and a speaker at various corporate events and public webinars. He has authored several articles on Microsoft Dynamics CRM 4.0 in a popular online developer magazine. Most recently, he has been involved in evangelizing the aspects of Application Lifecycle Management and developer collaboration, using Team Foundation Server 11 Beta. He also spends quite a bit of time travelling and training the different teams on the new features of .NET Framework 4.5 and Windows 8 Metro application development. Sandeep holds an MS degree in Software Systems from BITS Pilani, and his areas of interest include Service-oriented Computing, Pervasive Computing, and Haptic Devices. He occasionally blogs at `http://vstslive.wordpress.com` and can be reached over email at `sandeep.chanda@neudesic.com`.

Currently celebrating a decade of technological innovation, Neudesic was founded in 2002 by forward-thinking industry veterans Parsa Rohani, Tim Marshall, and Anthony Ferry, who saw opportunity in the development of Microsoft's .NET platform. Neudesic has since acquired a deep understanding of Microsoft's entire technology stack. The Microsoft National Systems Integrator and Gold ISV Partner has leveraged its expertise in Microsoft's various platforms to become a leading provider of SharePoint, Dynamics CRM, Azure, and mobile solutions.

Through the years, various industry and business publications have recognized Neudesic's meteoric rise from a small startup with a vision to an established force on a mission. For the fifth straight year in 2011, Inc. Magazine named Neudesic to its list of America's fastest growing private companies.

Sandeep is associated with Neudesic India, the company's international presence in India headed by Ashish Agarwal. Ashish is an alumnus of University of South California, and joined Neudesic in the early days of its inception and has since led the India team to over 100 successful engagements associating with more than 30 clients.

Acknowledgement

The best part about writing a book is working with an awesome team that motivates you to give it your best. That you are holding this book today is attributed to the phenomenal team that made it happen.

Thanks to the entire editorial team, especially Rukshana Khambatta, Shreerang Deshpande, and Michelle Quadros, who managed the project with the meticulous planning and the coordination. This book would not have happened without Rukshana lending her ears to my original idea and giving it the shape that it needed to address the target audience, Shreerang's valuable inputs during the review, and Michelle's patience in coordinating with me and managing the schedule. An extended thanks to Vrinda Amberkar and Devdutt Kulkarni for their exhaustive scrutiny of every minute detail of the transcript and bringing out a quality blueprint for release.

To Rick Garibay, Microsoft Connected Systems MVP and GM CS Practice at Neudesic, for taking time out from his extremely busy schedule and writing a foreword for this book.

To the reviewers, Senthil Kumar and Pushpendra Singh, for their valuable inputs and expert feedback.

To Pushpendra Singh, Principal Consultant at Neudesic for his invaluable contribution to Chapter Four and timely advice in making this project a success.

To the partners at Neudesic, Parsa Rohani, Tim Marshall, Anthony Ferry, and Ashish Agarwal for creating such a wonderful company with which I am proud to be associated.

To Mickey Williams, David Pallmann, Rick Garibay, David Barkol, Mark Kuperstein, and Suman Choppala from Neudesic for being a source of inspiration and giving me the courage to write.

To Shaun Cicoria from Microsoft for getting me started on the concepts of claims-based identity and helping with resources at critical times.

To Mahesh Pesani, Rajasekhar Tonduru, Hemant Joshi, and Rajesh Nair for their friendly tips on the source code and images in several recipes.

And to my family: my wife Sarita, my daughter Aayushi, and my parents, for letting me spoil countless of their weekends during the course of writing the book.

About the Reviewers

Senthil Kumar is a Software Engineer with three years of experience in the IT industry.

He is currently working as a Software Engineer in Bangalore and works mainly on the Windows or Client Development technologies and has good working experience in C#, .NET, Delphi, WinForms, and SQL Server.

He is also a Microsoft Certified Professional (MCP) in ASP.NET. He blogs at `http://www.ginktage.com` and `http://www.windowsphonerocks.com`. He enjoys learning as much as he can about all the things related to the technologies to get a well-rounded exposure of technologies that surround him.

Senthil completed his Master of Computer Applications from Christ College (Autonomous), Bangalore in the year 2009 and is an MCA rank holder.

He is passionate about the Microsoft technologies, especially Windows Phone development.

You can connect with him on Twitter (`http://twitter.com/isenthil`), on Facebook (`http://www.facebook.com/kumarbsenthil`), and on his blog (`www.ginktage.com`).

Pushpendra Singh is a Principal Consultant at Neudesic, a Microsoft National Systems Integrator and Gold Certified Partner. He is a senior member of Custom Applications Development Practice at Neudesic and has been working on Microsoft Technologies for the past 6 years. He has played the multiple roles including that of a Senior Architect on the enterprise-scale projects spanning several domains. His recent focus has been on building scalable and future-proof applications using Microsoft .NET Framework 4.0, Windows Azure, WCF, REST, WIF, WPF, and ASP.NET MVC 3. He spends his free time reading books or playing outdoor games, such as soccer, volleyball, and cricket.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

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.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across 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 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 nine 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` on Twitter, or the *Packt Enterprise* Facebook page.

I would like to dedicate this book to my father, Salil Kumar Chanda.

Table of Contents

Preface

Implementing security as a cross-cutting concern has several challenges. Consequently, the modern application development and service-oriented computing practices are alluding to the idea of claims-based identity implementation for access control. Microsoft's Identity and Access Control paradigm leverages the industry standard open specifications on claims-based security and provides the tools, the runtime, and the platform support for facilitating the development of the claims-enabled applications.

This book explores the real world scenarios on building claims-enabled .NET Framework applications using Windows Identity Foundation (WIF), Active Directory Federation Services 2.0 (AD FS 2.0), and Windows Azure Access Control Services 2.0 (ACS 2.0), the three most widely used products from Microsoft's Identity and Access Control stack.

Packed with more than 30 hands-on recipes, the book starts with introducing you to the world of claims-based identity in .NET Framework 4.0, and then moves on to demonstrate the capabilities of the runtime and the associated SDK. This includes the steps for performing identity delegation in ASP.NET MVC 3 applications, creating WCF security token services, extending the runtime to provide support for SAML 2.0 specifications, and using Windows Azure ACS as a trusted source for implementing access control. Further, the book dives deep into the relevant support extended in some of the server technologies of the ecosystem including Microsoft SharePoint 2010, Dynamics CRM 2011 and Sales Force. In addition, it also features a chapter on the newer capabilities of the runtime including support for claims in the Windows Server 8 and Windows 8 Metro style applications.

This book provides a mixture of recipes from basic to advance to enable the professional developers to implement claims-based identity in enterprise-wide scalable and interoperable applications.

What this book covers

Chapter 1, *Overview of Claims-based Identity*, introduces readers to the concept of claims-based identity, provides an overview of the Security Assertion Mark-up Language (SAML) specification, and gets them ready to start with the rest of the book.

Chapter 2, *Programming with Windows Identity Foundation*, introduces Windows Identity Foundation that is a .NET Framework runtime feature for building claims-based applications using Microsoft's Identity and Access Management paradigm. This chapter will cover aspects of programming claims in .NET applications using WIF with real world examples.

Chapter 3, *Advanced Programming with Windows Identity Foundation*, digs deep into the anatomy of Windows Identity Foundation and cover real world examples on building custom Security Token Service (STS) and extending the runtime to support SAML 2.0 profiles.

Chapter 4, *Cloud-based Identity with Azure Access Control Service*, introduces Azure Access Control Services 2.0 that provides cloud-based identity management solutions based on Microsoft's Identity and Access Management paradigm. This chapter will cover aspects of cloud-based authentication services and showcase recipes exploring claims-based identity with ACS 2.0 in the native mobile applications.

Chapter 5, *Identity Management with Active Directory Federation Services*, introduces AD FS 2.0 that provides federation services using the claims-based identity model on Active Directory users. This chapter covers the aspects of configuring a federation server using AD FS 2.0 and using it in conjunction with WIF and Azure ACS 2.0 to serve end-to-end security needs of an enterprise.

Chapter 6, *Enterprise Server Interoperability with WIF, Azure ACS 2.0, and AD FS 2.0*, focuses on enabling claims-based identity in some of the popular enterprise servers and cloud technologies from Microsoft including Microsoft SharePoint Server 2010, Microsoft Dynamics CRM Server 2011, Windows Azure, and Microsoft Office 365. In addition, it also explores the steps to provision a seamless Single Sign-On experience in SalesForce with AD FS 2.0.

Chapter 7, *Extension and Future of Windows Identity Foundation*, provides a glimpse of the future of claims-based identity with Windows 8 and .NET Framework 4.5. In addition, you will learn about some of the enhancements in the WIF runtime to provide support for claims-based identity in Windows Workflow Foundation and enable the developers to leverage the latest SAML 2.0 specifications for building SP-Lite compliant applications.

What you need for this book

A strong foundation in the C# programming language and .NET Framework 4.0 is expected, along with a good understanding of the authentication and the authorization concepts (Windows-based and Forms-based) in .NET. In addition, having the following skill sets is desirable:

- Application development experience in mobility platforms such as Windows Phone 7 and Apple iPhone.
- Administrative knowledge of operating and configuring Windows Server 2008 R2 and Windows Server 8, Windows Azure Management Portal, Microsoft SharePoint Server 2010, Microsoft Dynamics CRM Server 2011, and Sales Force.

No prior knowledge of the subject is necessary.

Who this book is for

This book is for the professional .NET developer building access control in his applications using claims-based identity. This book is also an excellent choice for the professionals and the IT administrators trying to enable Single Sign-On across the applications within the enterprise and in the cloud spanning interoperable platforms. The book introduces the readers to the concept of claims-based identity and then walks them through the recipes addressing the complex authentication and authorization scenarios.

Conventions

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

Code words in text are shown as follows: "In this recipe, we will find out how a Windows identity can be abstracted with claims using the `System.IdentityModel` assembly in .NET Framework 4.0."

A block of code is set as follows:

```
using (WindowsClaimSet claims = new WindowsClaimSet(WindowsIdentity.
GetCurrent()))
{
foreach (var claim in claims)
  {
  Console.WriteLine(string.Format("Claim Type: {0}",
  claim.ClaimType));
  Console.WriteLine(string.Format("Resource: {0}",
  claim.Resource.ToString()));
  Console.WriteLine(string.Format("Right: {0}", claim.Right));
  Console.WriteLine
  ("*******************************************");
  }
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
<!--Commented out by FedUtil-->
<!--<authentication mode="Windows" />-->
<authorization>
<deny users="?" />
</authorization>
```

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

```
makecert -r -pe -n "CN= SamlTokenSigningCertificate" -b 01/01/2010 -e
 01/01/2012 -sky exchange -ss my
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "In the **Federation Utility wizard**, enter the application URL in the **Application URI** drop-down list, and click on **Next**."

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 may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title through 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 on `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 for all Packt books you have purchased from your account at `http://www.packtpub.com`. 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 would 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/support`, 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.

Piracy

Piracy of copyright 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

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Overview of Claims-based Identity

As a developer of the Microsoft .NET Framework 4.0 based applications, in this chapter you can look forward to learning the ways of:

- Abstracting identity with claims
- Designing the claims-based tokens using **Security Assertion Markup Language** (**SAML**)
- Augmenting security with a claims-based architecture
- Implementing federated security using a **Security Token Service** (**STS**)
- Implementing Single Sign-On using claims
- Implementing Single Sign-Out in a trust realm
- Configuring certificates for the claims-based applications

Introduction

Claims-based identity provides a standard way of acquiring identity information by heterogeneous applications to validate service requests within and outside an organization and also over the Web. This chapter is a precursor to the forthcoming chapters on **Windows Identity Foundation** (**WIF**), **Windows Azure Access Control Services** (**ACS 2.0**), and **Active Directory Federation Services v2.0** (**AD FS 2.0**), all of these being a part of the Microsoft's initiative in the identity and access management using claims. The chapter explores the recipes for abstracting identity with claims and provides an overview of the Security Assertion Markup Language specifications. In addition, this chapter also explores a few claims-based architectures that help augment existing security infrastructure. The chapter is designed towards preparing the readers for the rest of the book.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.PacktPub.com`. 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.

Abstracting identity with claims

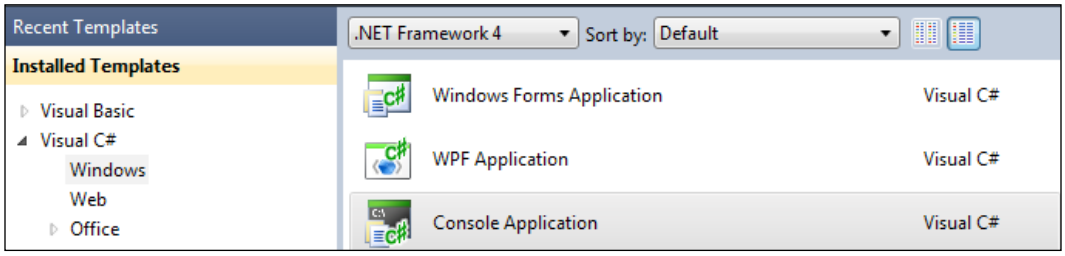
Authentication and authorization are two of the most common aspects of the application security. In Windows, security is generally handled using the Kerberos or the NTLM security tokens. The user is provided with credentials that include a domain user ID and a password, and these credentials are validated against the user's entry in the Active Directory. Role-based security is implemented with the help of authorization managers that control the level of access for the user.

This works well within the boundaries of the Windows ecosystem; however, it gets difficult if the application has to support the users that do not have Windows Active Directory credentials. In the real world, the applications spanning multiple platforms interact with each other and require the security context to be shared. Using a claims-based identity model provides a robust way of handling authentication and authorization across the discrete systems. Throughout this chapter, we will explore the recipes that will help you gain an understanding of how claims-based identity is core to the .NET Framework 4.0 and help you get started on the Microsoft's Identity and Access Management paradigm. In this recipe, we will find out how a Windows identity can be abstracted with claims using the `System.IdentityModel` assembly in .NET Framework 4.0.

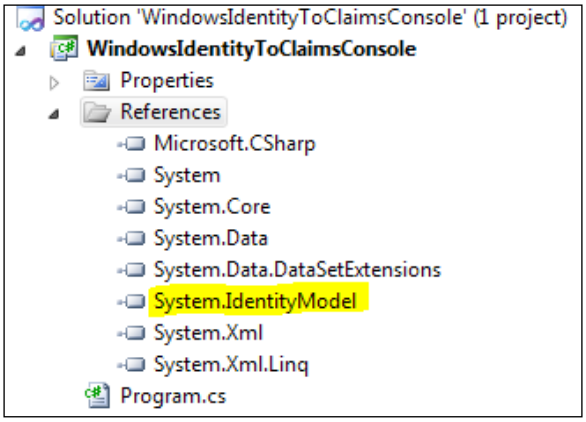
How to do it...

To create a collection of claims from a `WindowsIdentity` (`System.Security.Principal`) object, perform the following steps:

1. Create a new Visual C# Console Application project in Visual Studio 2010.



2. Add a reference to the **System.IdentityModel** assembly, as shown in the following screenshot:



3. Open the **Program.cs** file, and include the `System.IdentityModel.Claims` and the `System.Security.Principal` namespaces.
4. In the `Main` method, create a new instance of the `WindowsClaimSet` class, and pass the current context identity as a parameter to the constructor:

```
using (WindowsClaimSet claims = new
WindowsClaimSet(WindowsIdentity.GetCurrent()))
{

}
```

5. Loop through the `ClaimSet` object and print the claim information into the console output:

```
using (WindowsClaimSet claims = new
WindowsClaimSet(WindowsIdentity.GetCurrent()))
{
foreach (var claim in claims)
  {
  Console.WriteLine(string.Format("Claim Type: {0}",
  claim.ClaimType));
  Console.WriteLine(string.Format("Resource: {0}",
  claim.Resource.ToString()));
  Console.WriteLine(string.Format("Right: {0}", claim.Right));
  Console.WriteLine
  ("**********************************************");
  }
}

Console.ReadLine();
```

6. Compile and run the project. The result is displayed in the console window:

```
Claim Type: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/sid
Resource: S-1-5-21-436374069-484763869-1708537768-6674
Right: http://schemas.xmlsoap.org/ws/2005/05/identity/right/identity
**********************************************
Claim Type: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/sid
Resource: S-1-5-21-436374069-484763869-1708537768-6674
Right: http://schemas.xmlsoap.org/ws/2005/05/identity/right/possessproperty
**********************************************
Claim Type: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name
Resource: CORP\Sandeep.Chanda
Right: http://schemas.xmlsoap.org/ws/2005/05/identity/right/possessproperty
**********************************************
```

How it works...

The `WindowsClaimSet` class inherits from the `System.IdentityModel.Claims.ClaimSet`. `ClaimSet` represents a collection of claims (`System.IdentityModel.Claims.Claim`) associated with an entity. The `WindowsClaimSet` constructor accepts the current Windows user identity as a parameter and returns a `ClaimSet` object containing the collection of claims that represent the Windows Active Directory groups of the user. The current Windows user identity is fetched using the `WindowsIdentity.GetCurrent` method. Generated `ClaimSet` can be used to create a signed security token that can be passed on to a service to create a security context and implement role-based access control. We will see how to create a security token from a `ClaimSet` object later in the chapter.

The default expiration time for the claims collection is set to 10 hours. You can explicitly set the expiration time in the `WindowsClaimSet` overloaded constructor.

A claim is used to identify a user or provide access to a particular resource requested by the user. There are three properties exposed by the `Claim` class:

- `ClaimType`: It identifies the type of claim. In our example, `Sid` (security identifier) and `Name` are the two claim types displayed in the console window. A list of supported claim types is available at the following URL: `http://msdn.microsoft.com/en-us/library/system.identitymodel.claims.claimtypes.aspx`.
- `Resource`: It identifies the resource associated with the claim.
- `Right`: It is a URI representing the `Identity` or `PossessProperty` right associated with the claim. `PossessProperty` determines whether the user has the access to `Resource`.

Both the `Claim` and the `ClaimSet` classes are serialization-friendly, which allows them to be transmitted over service boundaries.

There's more...

In addition to `WindowsClaimSet`, the `System.IdentityModel.Claims` namespace provides a `DefaultClaimSet` class that allows you to create your implementation of claims, and a `X509CertificateClaimSet` class to abstract claims from an X.509 certificate.

Authorization context

The `System.IdentityModel.Policy` namespace exposes a `AuthorizationContext` class that can be used to evaluate the authorization policies in a sent message. The `AuthorizationContext` class has a `ClaimSet` property that allows a service to retrieve all the claims associated with the security token in the sent message. You can learn more with an example in the MSDN documentation at `http://msdn.microsoft.com/en-us/library/system.identitymodel.policy.authorizationcontext.claimsets.aspx`.

See also

The complete source code for this recipe can be found in the `\Chapter 1\Recipe 1\` folder.

Designing claims-based tokens using Security Assertion Markup Language

The **Security Assertion Markup Language** (**SAML**) specification is an open security standard envisioned by **Organization for the Advancement of Structured Information Standards** (**OASIS**) Technical Committee for the exchange of security context across service boundaries. The SAML tokens are XML-based (can be transmitted using SOAP/HTTP) and provide a way of implementing claims-based identity that is particularly useful in interoperable scenarios across the identity providers and the service providers. This recipe shows how to create the SAML security tokens using the `System.IdentityModel` assembly in .NET Framework 4.0.

Getting ready

If you are not familiar with the SAML v1.1 specification, you can get the standard set and the schema files from `http://www.oasis-open.org/standards#samlv1.1`.