

# Enhancing Roles

## DotNetNuke Roadmap

Charles Nurse



Version 1.0.0

Last Updated: June 21, 2006

Category: DotNetNuke 3.3/4.1



## Enhancing Roles

*Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user.*

*The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Perpetual Motion Interactive Systems, Inc. Perpetual Motion Interactive Systems may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Perpetual Motion, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*Copyright © 2005, Perpetual Motion Interactive Systems, Inc. All Rights Reserved.*

*DotNetNuke® and the DotNetNuke logo are either registered trademarks or trademarks of Perpetual Motion Interactive Systems, Inc. in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

## Abstract

**This document describes proposed enhancements of the Roles capabilities of DotNetNuke.**

## Contents

<b>Introduction .....</b>	<b>1</b>
<b>Role Group Enhancement .....</b>	<b>2</b>
Introduction .....	2
New RoleGroupInfo class.....	2
Additions to RoleInfo .....	3
New Methods in Provider .....	3
Updates to Data Layer.....	4
Updates to User Interface .....	4
<b>SecurityPrinciples.....</b>	<b>7</b>
Introduction .....	7
<b>Effective Date Enhancement .....</b>	<b>11</b>
Introduction .....	11
Additions to UserRoleInfo .....	11
Updates to Data Layer.....	11
Updates to User Interface .....	11
<b>RSVP Code Enhancement .....</b>	<b>13</b>
Introduction .....	13



## Enhancing Roles

Additions to RoleInfo .....	13
Updates to Data Layer.....	13
Updates to User Interface .....	13
<b>Additional Information.....</b>	<b>16</b>
Errors and Omissions .....	16
<b>Appendix A: Document History .....</b>	<b>17</b>

## Introduction

This document proposes three enhancements for Roles.

- ✧ Role Groups – In some situations, for instance on [www.dotnetnuke.com](http://www.dotnetnuke.com), there is a huge number of Roles, and grouping them together would provide a better administration experience. This feature is purely a convenience feature for role administration – it is the role itself that a user is assigned to or is given permission.
- ✧ Effective Date – we currently provide an ExpiryDate, but if we also provided an EffectiveDate, this would allow roles to be assigned for a “future” need
- ✧ RSVP Code – this would be used to “Auto-Assign” somebody to a role, if they had been provided with a code. This works similarly to the special discount codes given in retail propmotions.

# Role Group Enhancement

## Introduction

This enhancement provides support for the concept of “Role Groups”. In some situations, for instance on [www.dotnetnuke.com](http://www.dotnetnuke.com), there is a huge number of Roles, and grouping them together would provide a better administration experience. This feature is purely a convenience feature for role administration – it is the role itself that a user is assigned to or is given permission.

## New RoleGroupInfo class

A new RoleGroupInfo class will need to be added to support the concept of Role groups. It is important to emphasize that this concept is for “convenience” when there are a large number of roles. It is still the role itself that is important, but role grouping allows the permissions grids to be more manageable.

### RoleGroupId

The integer property representing the Id of the Role group

### PortalId

The Id of the Portal

### RoleGroupName

A string property, representing the name of the Role group.

### RoleGroupDescription

A string property, representing the description of the Role group.

## Additions to RoleInfo

### RoleGroupId

This field provides a link to the Role Group to which this role belongs. This field can be null, which indicates that the role is “global” to a portal.

## New Methods in Provider

### CreateRoleGroup

Adds a new Role Group

```
Public MustOverride Function CreateRoleGroup(ByVal roleGroup As RoleGroupInfo) As Integer
```

### DeleteRoleGroup

Deletes an existing Role Group

```
Public MustOverride Sub DeleteRoleGroup(ByVal roleGroup As RoleGroupInfo)
```

### GetRoleGroup

Gets a specific Role Group

```
Public MustOverride Function GetRoleGroup(ByVal portalId As Integer, ByVal roleGroupId As Integer) As RoleGroupInfo
```

### GetRoleGroups

Gets all the Role Groups in a Portal

```
Public MustOverride Function GetRoleGroups(ByVal portalId As Integer) As ArrayList
```

### GetRolesByGroup

Gets all the Roles in a Group (use -1 to get all roles in the global or unassigned group)

## Enhancing Roles

```
Public MustOverride Function GetRolesByGroup(ByVal portalId As Integer, ByVal roleGroupId  
As Integer) As ArrayList
```

### UpdateRoleGroup

Updates an Existing Role Group

```
Public MustOverride Sub UpdateRoleGroup(ByVal roleGroup As RoleGroupInfo)
```

In addition similar methods will need to be added to the RoleController.

### Updates to Data Layer

In addition to new Data Layer methods to support the new Provider methods, the Add and Update Role methods will need to be updated to support the RoleGroupId property of the Role.

### Updates to User Interface

In order to support the RoleGroup concept the Security Roles Admin UI and the Permission Grids will need to be updated.

#### Filter Security Roles By Group

The Security Roles grid will need to be “filtered” by Role Group

## Enhancing Roles



## Manage Roles

The role groups will need to be managed (see AddNew Role Group in figure above)



## Assign a Role to a Group

The Role will need to be able to be assigned to the Role Group

## Enhancing Roles

▼ 🔑 **Edit Security Roles** ?

☐ **Basic Settings**

In this section, you can set up the basic settings for this role.

🔑 **Role Name:** My New Role

🔑 **Description:**

🔑 **Role Group:** 

< Global Roles > ▼

< Global Roles >  
My New Role Group

🔑 **Public Role?**

🔑 **Auto Assignment?**

☐ **Advanced Settings**

[Update](#) [Cancel](#) [Delete](#) [Manage Users in this Role](#)

## Update Permission Grids

The Permission Grids will need to be updated to allow the roles to be filtered by Group

🔑 **Permissions:**

**Filter By Group:** 

< Global Roles > ▼

< All Roles >  
< Global Roles >  
My New Role Group

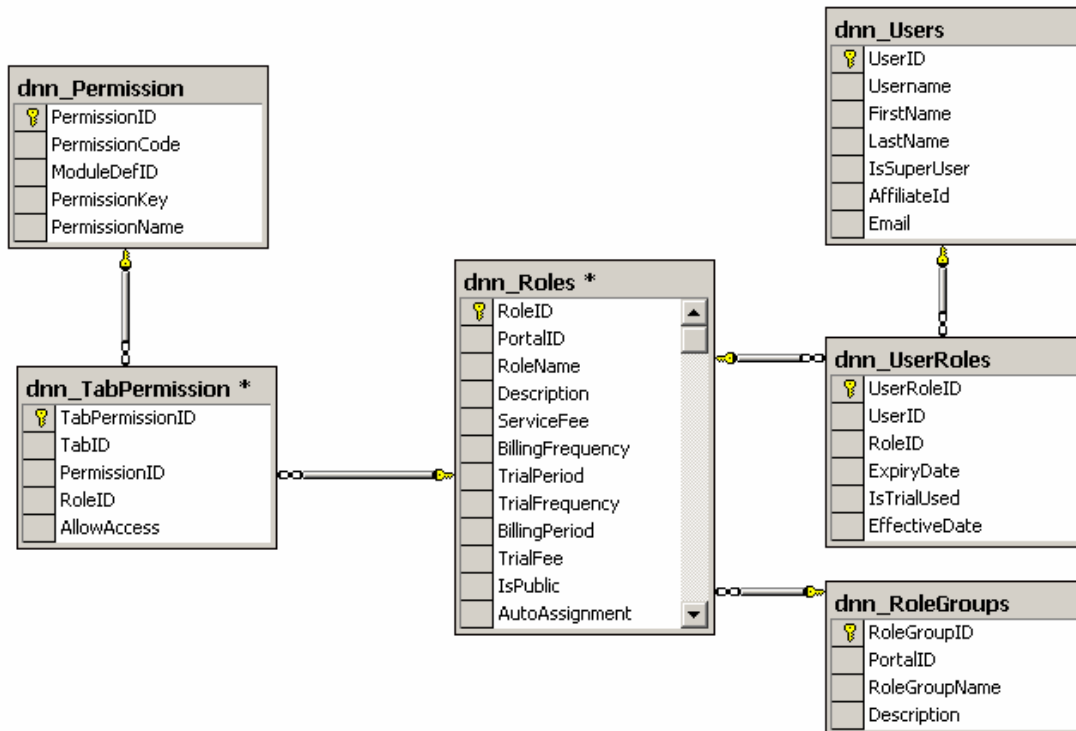
Administrators	<input type="checkbox"/>	<input type="checkbox"/>
<b>All Users</b>	<input type="checkbox"/>	<input type="checkbox"/>
My New Role	<input type="checkbox"/>	<input type="checkbox"/>
<b>New Global Role</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Registered Users	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>Subscribers</b>	<input type="checkbox"/>	<input type="checkbox"/>
Unauthenticated Users	<input type="checkbox"/>	<input type="checkbox"/>

# SecurityPrinciples

## Introduction

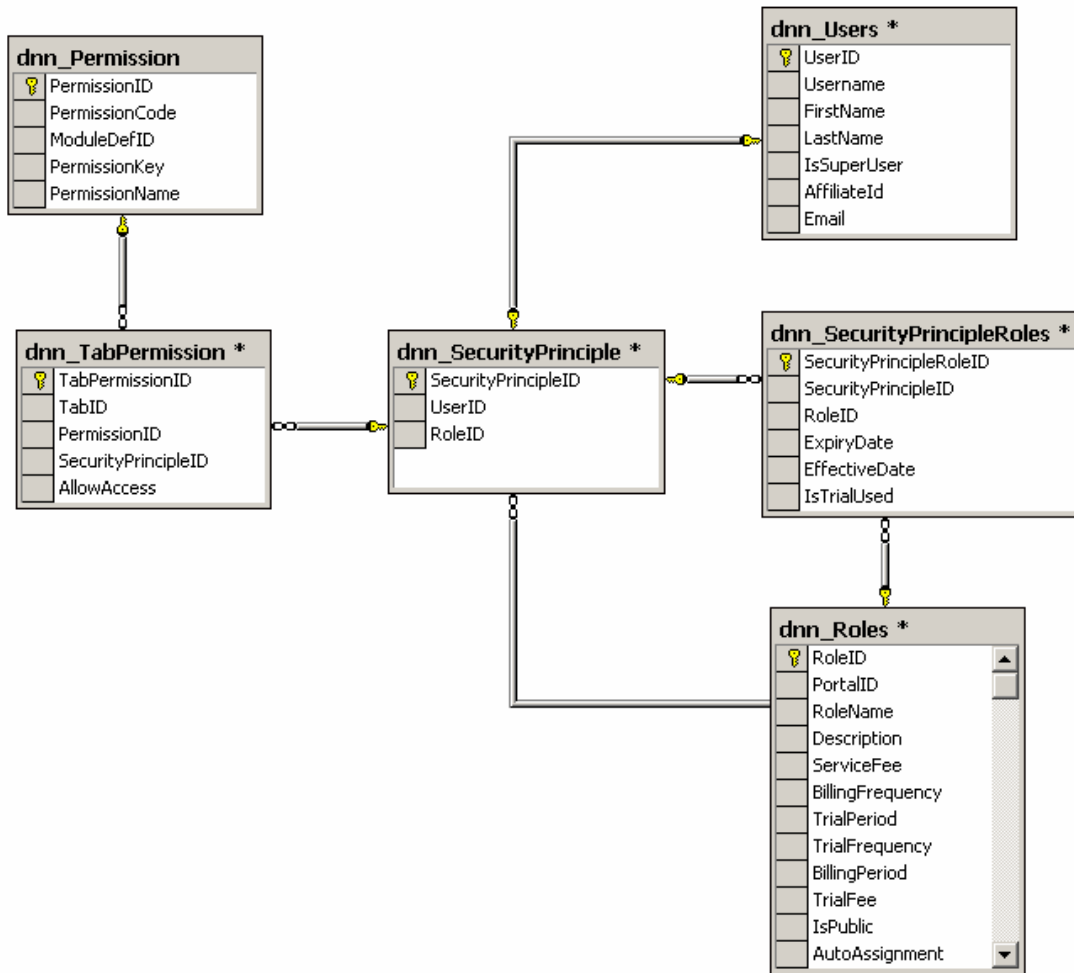
This enhancement extends the ideas proposed in the previous enhancement to add the concept that any User, Role or Role Group could be assigned permissions to a resource.

Currently the Security Model is that Users can be assigned to Roles and that Roles can be assigned to permissions. In some instances, roles need to be created for the sole purpose of providing a User permission to a resource. Thus everything revolves around roles.



## Enhancing Roles

This enhancement proposes that permissions may be assigned to “all” Security objects, Role Groups, Roles and Users. In fact strictly speaking to take this enhancement to its logical conclusion, it also proposes that Roles (which are synonymous with UserGroups in Windows terms) can consist of other Roles and Users, and either Roles or Users are SecurityPrinciples that can be assigned to permissions. This is summarized in the Data Entity diagram below:



The diagram shows that now it is the SecurityPrinciple object that is assigned a permission. The SecurityPrinciple is either a role or a user and a Roles membership is made up of SecurityPrinciples (ie both roles and users).

## Enhancing Roles

### ISecurityPrinciple

The discussions so far have focused on the enhancement from a Database perspective. What would we need to do from a Business Layer perspective. One solution would be to create an interface `ISecurityPrinciple`. Both roles and users would implement `ISecurityPrinciple`. In addition the `UserRole` classes would need to be modified to support any `ISecurityPrinciple` being a member of a role.

### Impact of this change

Roles and permissions are used all over the application. The `TabInfo` object carries three sets of permissions.

- ✧ `AuthorizedRoles` – a semi-colon delimited list of rolenames authorized to view the tab
- ✧ `AdministratorRoles` – a semi-colon delimited list of rolenames authorized to manage the tab
- ✧ `TabPermissions` – collection of `TabPermission` objects

The `ModuleInfo` object has a similar set of permissions

- ✧ `AuthorizedEditRoles` – a semi-colon delimited list of rolenames authorized to edit the module
- ✧ `AuthorizedViewRoles` – a semi-colon delimited list of rolenames authorized to view the module
- ✧ `ModulePermissions` – collection `ModulePermission` objects

In reality both objects really only need the `TabPermissions` and `ModulePermissions` respectively. The other two properties are holdovers from v2.x. However, while they are not necessary because the `Permissions` collection is a better model, there are still a lot of instances where they are still used for convenience. eg.

```
PortalSecurity.IsInRoles(PortalSettings.ActiveTab.AdministratorRoles.ToString) = True
```

Thus if we changed the security model to allow a `SecurityPrinciple` to be assigned permissions, then these calls would all need to be refactored to correctly use calls like:

## Enhancing Roles

```
TabPermissionController.HasTabPermission(PortalSettings.ActiveTab.TabPermissions,  
PermissionKey)
```

This is not a huge undertaking. There are somewhere in the vicinity of 80-100 references to the semi-colon delimited roles lists that would need to be refactored.

The concept of FolderPermission was added at the time that the 3.x permission model was introduced so we really only need to worry about Tabs and Modules.

All of the semi-colon delimited roles lists are used in conjunction with the IsInRoles method of PortalSecurity. In fact, even though Folders never used the “roles list” concept in 2.x, as PortalSecurity.IsInRoles was already used for ModulePermissions and TabPermissions, a method was added to FileController to generate a semi-colon delimited list to pass to the PortalSecurity method.

So in order to implement the concept of a SecurityPrinciple throughout the application, we would need to replace PortalSecurity.IsInRoles with a HasPermission method which could be passed a collection of permissions to see if the user has the permission to access the resource.

The biggest impact would be in the UI. Managing the permissions grid was a challenge with just roles to consider on big sites. Managing roles and users is a much bigger challenge.

## Implementing the change incrementally

Probably the best approach is to implement some of these changes incrementally. For example, there are three obvious steps to a possible implementation:

1. Remove all use of the semi-colon delimited roles lists and replace the use of PortalSecurity.IsInRole by PortalSecurity.HasPermission.
2. Add the ability for roles to contain roles.
3. Implement the concept of a SecurityPrinciple (role or user) in the permissions management, and the ISecurityPrinciple Interface, to abstract users and roles in permission management.

# Effective Date Enhancement

## Introduction

EffectiveDate – we currently provide an ExpiryDate, but if we also provided an EffectiveDate, this would allow roles to be assigned for a “future” need.

## Additions to UserRoleInfo

In order to support the feature proposed above the following property would need to be added to the UserRoleInfo class.

### EffectiveDate

A DateTime property that represents the date that the role becomes effective. This will work similarly to ExpiryDate, except that it defines the date the role begins rather than ends.


## Updates to Data Layer

The User Role stored procedures and Data Layer methods will need to be updated to reflect the Effective Date property. Also the GetRolesByUser stored procedure will need to be updated to reflect the EffectiveDate in the WHERE clause, so as not to return roles that have not started yet.

## Updates to User Interface

In order to support the Effective Date the Manage Users in Role (or Manage Roles for User) control will need to be modified to accommodate the Effective Date field in both the Add/Update functionality and the display grid of UserRoles.

## Enhancing Roles

▼ User Roles 

Manage Roles for User Charles Nurse (Id: 14)

Security Role       Effective Date       Expiry Date

Administrators  [Calendar](#)       [Calendar](#)      [Add Role](#)

Send Notification?

---

	Security Role	Effective Date	Expiry Date
×	My New Role		
×	New Global Role		1/19/2008
×	Registered Users		

---

[Cancel](#)

## RSVP Code Enhancement

### Introduction

RSVPCode – this would be used to “Auto-Assign” somebody to a role, if they had been provided with a code. This works similarly to the special discount codes given in retail propmotions.

### Additions to RoleInfo

In order to support the feature proposed above the following property would need to be added to the RoleInfo class.

#### RSVPCode

A String that identifies the code for the role.

### Updates to Data Layer

The Add/Get/Update Role stored procedures and Data Layer methods will need to be updated to reflect the RSVPCode property.

### Updates to User Interface

In order to support the RSVPCode the Edit Roles User Interface and Register User Interface will need to be updated.

#### Ability to manage RSVPCode in EditRoles User Interface.

A text box to enter the code will need to be added to Edit Roles UI

## Enhancing Roles

### Advanced Settings

In this section, you can set up more advanced settings for this role.

[Update](#)
[Cancel](#)
[Delete](#)
[Manage Users in this Role](#)

### Ability for User to subscribe to role by entering RSVP Code

The Register User Interface will need to provide the ability for a “User” to enter a provided code, and automatically subscribe to a role.

### Membership Services

[Subscribe](#)

	Name	Description	Fee	Every	Period	Trial	Every	Period	Expiry Date
<a href="#">Subscribe</a>	My New Role	My New Role							
<a href="#">Unsubscribe</a>	New Global Role	New Global Role	5.00	1	Year(s)				1/19/2008
<a href="#">Subscribe</a>	Subscribers	A public role for portal subscriptions							

[Update](#)
[Cancel](#)
[Unregister](#)

#### Note:

The implementation would fetch all the roles from the database and automatically subscribe the user to any role that matches the RSVP Code. Thus it could be used to:

- Subscribe a User to one public free role
- Subscribe a User to one fee paid role (without paying the fee)
- Subscribe a User to more than one role (if two or more roles have the same code)

## Enhancing Roles

In addition if partial matching logic is used (rather than exact matching, it would be possible to do the following).

- Role A: RSVPCode = 111
- Role B: RSVPCode = 111,222
- Role C: RSVPCode = 222

A User entering 111 would subscribe to Roles A+B, while a user entering 222 would subscribe to Roles B+C

## Additional Information

The DotNetNuke Portal Application Framework is constantly being revised and improved. To ensure that you have the most recent version of the software and this document, please visit the DotNetNuke website at: <http://www.dotnetnuke.com>

The following additional websites provide helpful information about technologies and concepts related to DotNetNuke:

### **DotNetNuke Community Forums**

<http://www.dotnetnuke.com/tabid/795/Default.aspx>

### **Microsoft® ASP.Net**

<http://www.asp.net>

### **Open Source**

<http://www.opensource.org/>

### **W3C Cascading Style Sheets, level 1**

<http://www.w3.org/TR/CSS1>

## Errors and Omissions

If you discover any errors or omissions in this document, please email [marketing@dotnetnuke.com](mailto:marketing@dotnetnuke.com). Please provide the title of the document, the page number of the error and the corrected content along with any additional information that will help us in correcting the error.

## Appendix A: Document History

Version	Last Update	Author(s)	Changes
1.0.0	Jan 23, 2006	Charles Nurse	<ul style="list-style-type: none"><li>• First Draft</li></ul>