Global Knowledge®

Expert Reference Series of White Papers

Dynamic Access Control: Beyond Classic NTFS Permissions

Dynamic Access Control: Beyond Classic NTFS Permissions

Glenn Weadock, Global Knowledge Instructor, MCITP, MCSE, MCT, A+

Introduction

Dynamic Access Control in Windows Server 2012 lets you manage access to documents in ways that go beyond classic NTFS file system permissions. For example, if you want to allow Engineering department users in your Denver office read-only access to files relating to the Wind Turbine Project, Dynamic Access Control can do the job.

I. Overview of Dynamic Access Control

Windows Server 2008 R2 introduced the concept of File Classification Infrastructure (FCI): a way to permit organizations to use (and create) file properties, file property rules, and file management operations based on those rules. Administrators gained access to FCI through a new node in the File Server Resource Manager (FSRM) titled "Classification Management." FCI offered a new way of backing up, securing, restoring, archiving, and reporting on files.

FCI intrigued me, and I wrote a few posts about it in my "Network World" blog back in 2009. However, it was one of those technologies that seemed to simmer on "low," and it was rarely a hot topic in the classes I taught for Global Knowledge, or in the trade press. FCI felt a little half-baked when introduced: interesting conceptually, but short on tools and limited in its application.

Today, however, Windows Server 2012 has taken classification management to a new level, focusing on the access control aspect and given it a new acronym in the process: *Dynamic Access Control*. DAC is a hot topic now—you really can't get your Server 2012 certification without understanding it—and in this paper I'm going to illustrate its "nuts and bolts" by walking through a sample scenario: engineers in the Denver office aren't working on my company's Wind Turbine project anymore, so I don't want them to be able to modify project documents, just read them. With DAC, I can accomplish this without using security groups at all!

A. Limitations of NTFS Permissions

NTFS permissions let you restrict access to files by two criteria: user identity (a capability rarely used in practice) and security group membership. As useful as group-based access control is, particularly given the administrative conveniences of nesting groups within other groups ("role groups" within "rule groups"), sometimes organizations would rather control file access on the basis of other criteria: department, location, project, and so forth. In the past, we have tried to fit such criteria into the security group architecture, with results ranging from partial success to epic awkwardness. Share-level permissions don't help much, as these, too, are based on user and group identities.

B. How DAC Provides New Flexibility

DAC provides new flexibility by building on the FCI concept:

- DAC uses FCI to control resource access, instead of just file management operations.
- DAC leverages Active Directory user and computer attributes, called "claims" in the context of DAC.
- DAC lets us create file attributes ("classification properties" in Server 2008 R2 lingo, "resource properties" in DAC-speak) of our own design, or use new built-in ones Microsoft has provided.

- DAC offers the ability to build custom rules, and it uses Group Policy to make those rules available throughout the domain.
- Finally, DAC provides a mechanism for providing helpful information to users who have been denied access based on one or more of those rules.

Note that DAC does not supersede NTFS permissions or share permissions; it just provides another type of access control. Think of DAC as another hurdle the user must jump over in order to access a file.

Also note that you can use DAC with Windows security groups *without* taking advantage of AD user and device attributes; you would simply create central access rules and policies that leverage file resource properties, ignoring user and device claims. This would not be a bad way to ease into using DAC in an organization before taking on the additional complexity of rules and policies that incorporate user and device claims as well as file resource properties. You may be able to reduce the number of security groups you need, or reduce the number of nesting levels.

C. Requirements for DAC

I've seen contradictory information in the trade press regarding the Windows version requirements for DAC, so beware of overgeneralizations that you might read in MCSA test-prep books and even TechNet articles. The precise requirements depend on how you plan to use DAC. For example, you need:

- A Windows Server 2012+ file server with File Server Resource Manager installed
- A Windows Server 2012 schema (*i.e.*, a domain with at least one Server 2012+ domain controller)
- At least one Server 2012+ domain controller per site, if you're using device claims (Windows 8 clients must use Server 2012+ DC's)
- Windows 8+ clients, if you're using device claims OR if you're implementing "access-denied assistance"

You do *not* have to raise the Domain Functional Level (DFL) to Server 2012 or higher, contrary to multiple published references and blog posts! The scenario documented in this white paper was done in a test domain running at the Server 2008 R2 DFL.

II. Setting up Kerberos Claims Support

The standard Kerberos security access token that authorizes users to access network resources has always included the user's Security ID (SID) and the SIDs of all groups to which the user belongs. These SIDs function much like keys on a key chain. However, if you intend to employ user and device claims for Dynamic Access Control, this Kerberos security access token needs to be expanded—we need room on the keychain for new types of keys.

Happily, it's easy to turn on this capability. Fire up the Group Policy Management Console and create a new GPO linked to the Domain Controllers Organizational Unit (or, if you prefer, edit the existing Default Domain Controllers GPO). The policy setting you need to enable is *Computer Configuration > Policies > Administrative Templates > System > KDC > KDC support for claims, compound authentication and Kerberos armoring*.

Two Windows Server 2012 books I've seen stop there, but there's a bit more to it. Create a new GPO linked to whatever OUs contain your Windows 8 *clients* (or edit the existing Default Domain GPO) and enable the policy setting *Computer Configuration > Policies > Administrative Templates > System > Kerberos > Kerberos client support for claims, compound authentication and Kerberos armoring*. Finally, a setting you'll need for both file servers and clients if you're using device claims (as our example does) is *Computer Configuration > Policies > Administrative Templates > System > Kerberos > Support compound authentication*.

III. Creating User and Device Attributes

If you plan to use DAC to create rules involving user and/or device attributes, or "claims," the tool you'll need is the Active Directory Administrative Center (henceforth "ADAC"), accessible via Server Manager or by running *dsac.exe*.

A. User Claim Types ("Engineering Department")

In our scenario, we wish to create a rule that will apply to members of the Engineering department, such as employee Robert Goddard. Mr. Goddard is not in the "Engineering" OU; rather, his identification with the Engineering department is through an attribute of his user account in Active Directory (see Figure 1).

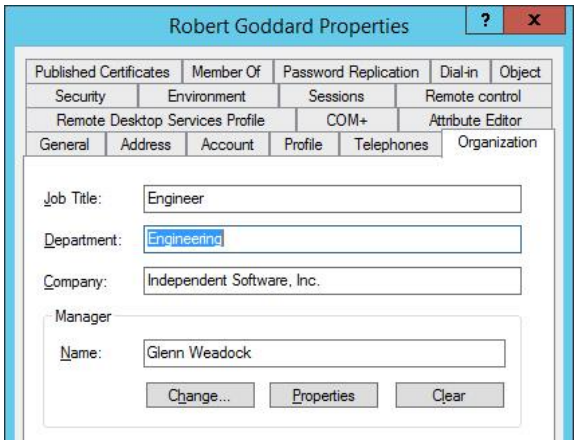

Figure 1: Robert Goddard's user account contains the Department attribute.

In order to make sure that the Department attribute is included in the extended Kerberos security access token, we have to create a "claim" in ADAC. (We wouldn't want every user attribute to be included in the token, as that would make the token unnecessarily huge.) In ADAC's tree view, right-click *Claim Types* and choose *New > Claim Type*. A new property sheet opens up (Figure 2), allowing you to select from a laundry list of AD object attributes in the upper left. For attributes that can apply to both users and computers, you choose the type of claim you want via a checkbox to the right; here, we're using the department attribute of the user object to create the user claim.

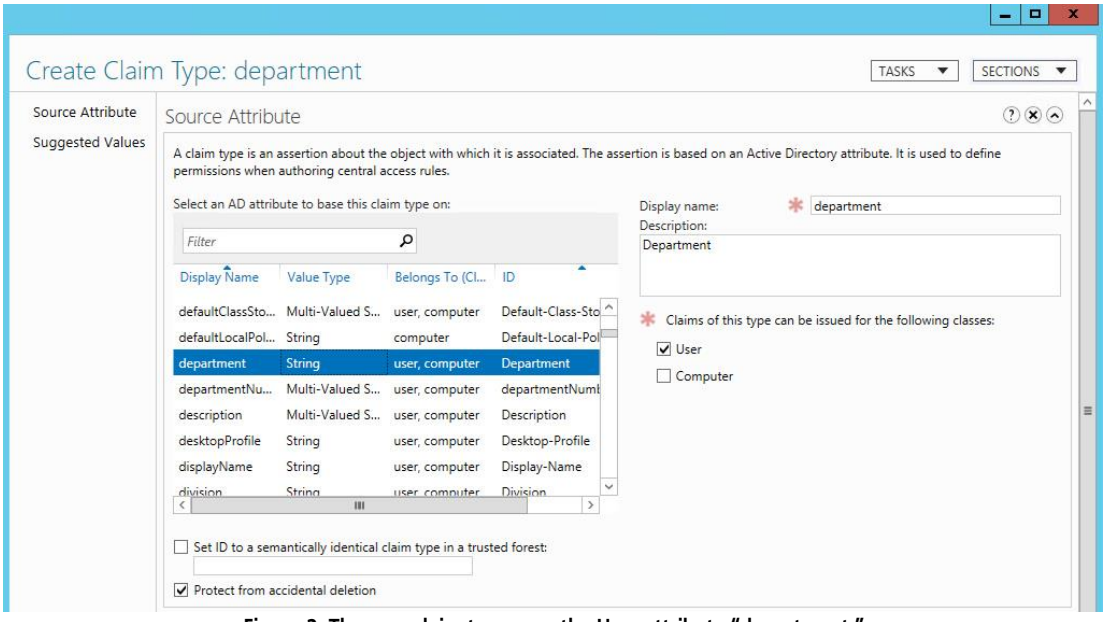

Figure 2: The new claim type uses the User attribute "department."

B. Device Claim Types ("Denver Office")

According to our scenario, engineers in the Denver office may access the files of interest in read-only mode. To specify the Denver office, we need to make the "location" attribute of the computer object in AD into a claim. The computer ISI-T520 has the location attribute set to Denver, as may be seen in Active Directory Users and Computers (Figure 3):

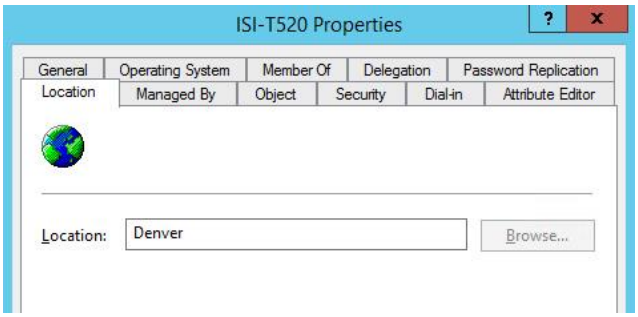

Figure 3: ISI-T520's computer account contains the Location attribute.

We then create a new claim type in Active Directory Administrative Center, much as we did for the user department claim type, but this time specifying the computer object (which is the only choice for the "location" attribute as there is no "location" for a user object in AD).

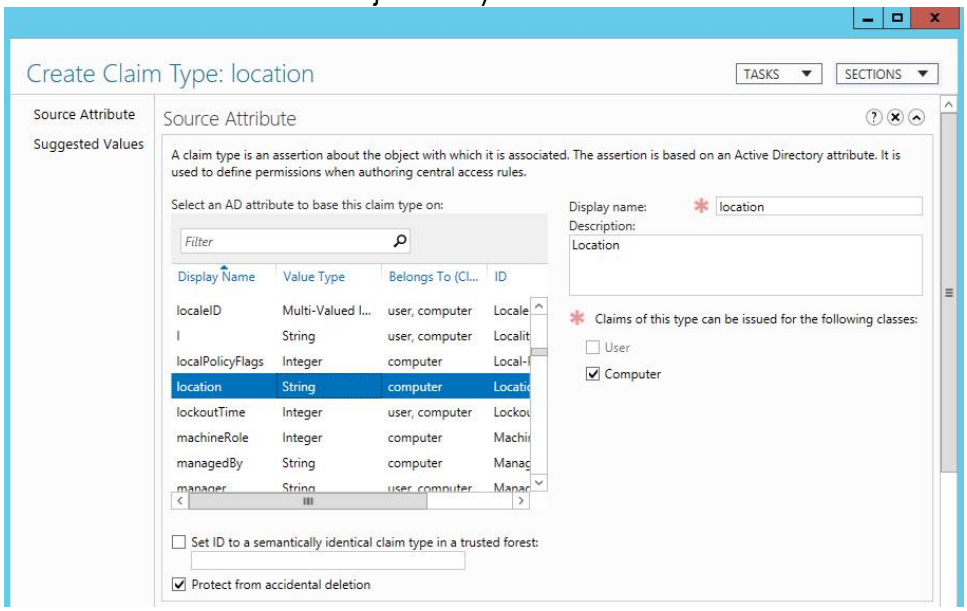

Figure 4: The second new claim type uses the computer "location" attribute.

IV. Creating File Classifications

A. Resource Properties ("Wind Turbine Project")

At this point we have created claims for the user and computer objects, claims that we can use shortly in a central access rule. The other puzzle piece we need before we can build a rule, however, is a way of identifying the files to which we wish to control access. In our example, we specify files that pertain to the "Wind Turbine Project." Unlike with NTFS permissions, the folder hierarchy doesn't matter: we can identify the "Wind Turbine Project" files in a variety of ways. This part of Dynamic Access Control has been around since Server 2008 R2, and is called "file classification."

A "resource property" is basically a file attribute that we can use for access control purposes. Resource properties were called "classification properties" in Server 2008 R2, and you still may see this language in Microsoft documentation. In ADAC, you can access, enable, and create resource properties by clicking the "Resource Properties" node under the "Dynamic Access Control" heading in the navigation pane. Microsoft starts us out with a grab-bag of predefined resource properties: Company, Compliancy, Confidentiality, Department, Discoverability, Folder Usage, Immutable, Impact, Intellectual Property, Personal Use, Personally Identifiable Information, Project, Required Clearance, Retention Period, and Retention Start Date. If none of these work for you, you can create a new resource property in ADAC; in our scenario, we can use the existing Project resource property, so it is only necessary to enable it (Figure 5).

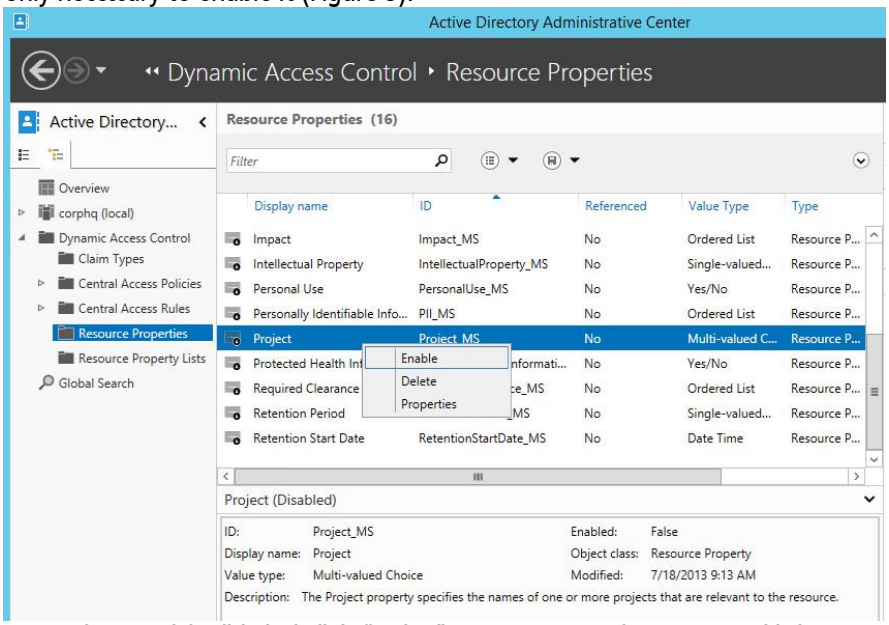

Figure 5: Right-click the built-in "Project" resource property in ADAC to enable it.

However, this will produce an error! Some types of resource property, like Project (which is a "multi-valued choice" type), require that we pre-populate at least one suggested value, to form the basis of a "pick list" should we decide to classify files or folders manually. So, right-click the resource property, choose "Properties" and provide the names of the projects your company is working on, e.g. "Wind Turbine" and "Battery Facility," *then* you'll be able to enable it.

If you create a new resource property, you have to add it to a "resource property list" before using it to classify files. You can simply add it to the "out of the box" list ("Global Resource Property List"), which by default contains all of the built-in resource properties. Later, you can create multiple lists and limit their scope of application if your DAC environment gets large and complex.

B. Updating Active Directory

The newly enabled or newly created resource properties and/or resource property lists now have to get updated in Active Directory. Some references that I've seen suggest that this must be done with the PowerShell cmdlet *Update-FSRMClassificationPropertyDefinition* on a file server that's running the File Server Resource Manager role service (installable via Server Manager). That's not quite true; in my network, restarting the server did the trick. The cmdlet is certainly quicker, though.

TIP: You can view claim types, resource properties, resource property lists, *etc.* in the *Active Directory Sites and Services* console. They reside under the "Services" node, in a container named "Claims Configuration." Alternatively, you can see the same information in ADSI Edit, under the Configuration and Services containers. This information propagates to other domain controllers through normal AD replication—a significant advance over Server 2008 R2 file classifications, which were not part of AD.

C. Classifying Files and Folders

Now things are getting fun. We have effectively created a new file system attribute, "Project," which we can assign to any file or folder that we like, irrespective of the server's folder hierarchy. For example, we might have a folder named "construction permits" that contains documentation for both the Wind Turbine and the Battery Facility projects. It is now a simple matter to view the properties page of each of the construction permit documents in Windows Explorer, and flag them to the correct project via the "Classification" tab's checkboxes, as shown in Figure 6 (I flagged the September and October PDFs as classified with the Wind Turbine project). The classification is also sticky: it stays with the file as long as you keep it on an NTFS volume.

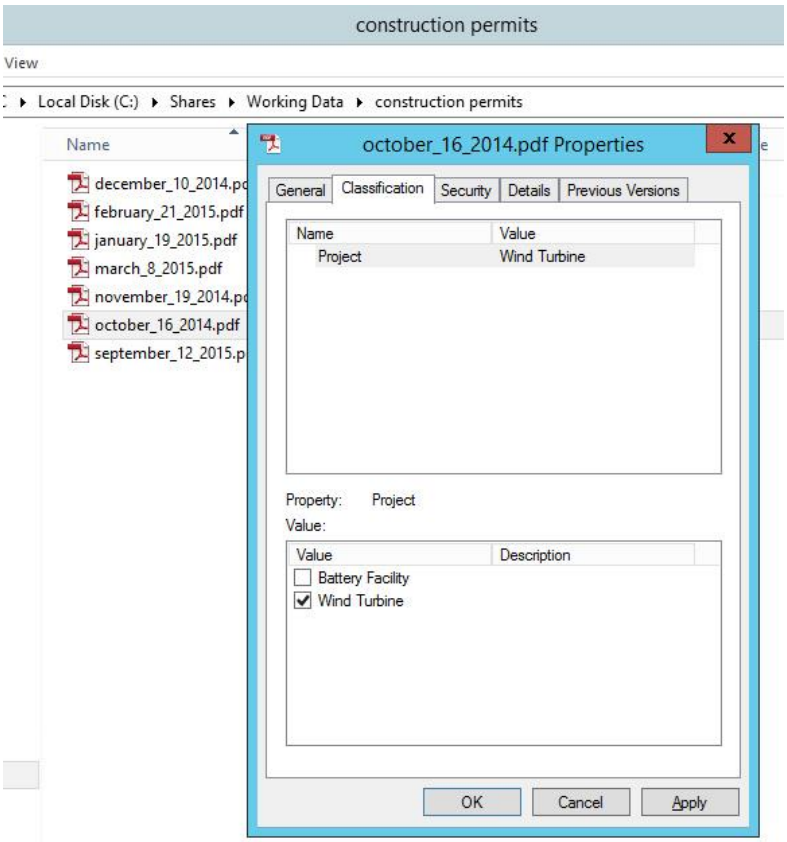

Figure 6: Resource properties appear on a file or folder's Classification tab.

You can also classify all documents within a folder via the folder's Classification tab. However, although manual classification might be fine when you have seven documents, or a lot of documents in a single folder, but it's less fine when you have 700 documents, or documents scattered all across the folder hierarchy. That's why the File Server Resource Manager console provides the facility to create *classification rules* that you can specify to be run once; regularly, on a schedule; or (and this is new for Server 2012) continuously. These classification rules can examine the content of files and auto-assign classifications (for example) based on literal strings or "regular expressions" that appear within the content. We can't examine all the nuances here, but auto-classification has been enhanced since Server 2008 R2; for one example, FSRM can now dynamically clear classification settings that no longer meet relevant classification rules.

V. Create and Deploy a Central Access Policy

Now that we know how to identify the right users (department=Engineering), the right computers (location=Denver), and the relevant files (project=Wind Turbine), all that remains is to build the central access rule, a policy based on that rule, and a GPO that makes the policy available domain-wide.

A. Create a Central Access Rule

To create the rule for our scenario, we go back to the ADAC console and choose *New > Central Access Rule*. There are two steps: targeting the resources for the rule (that is, specifying which files it should apply to), and specifying the permissions (what user and device claim conditions should produce a specified level of access). First we define the target resources with a rule condition, as shown in Figure 7. In the syntax of DAC, the resource-property condition reads *(Resource.Project Any of ("Wind Turbine"))*.

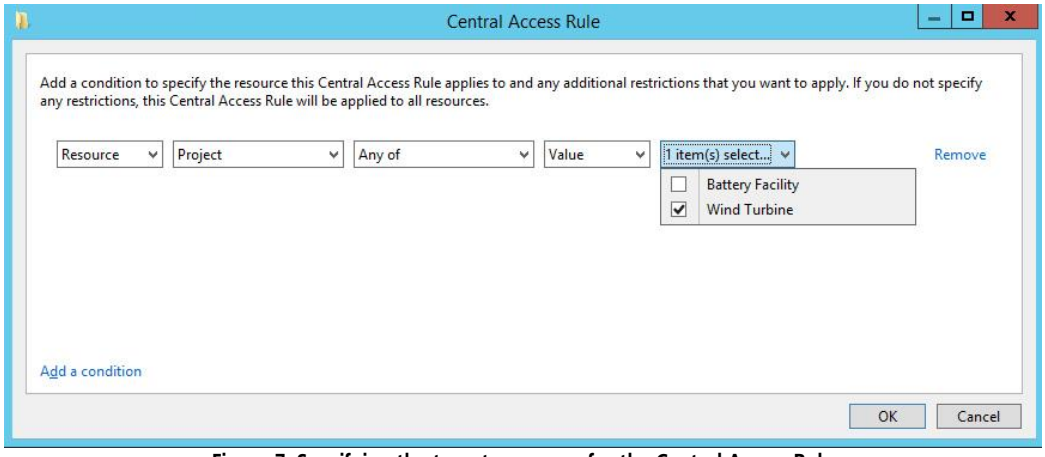

Figure 7: Specifying the target resources for the Central Access Rule

Next, we define the permissions. You can choose "proposed permissions" if you only want to audit the access events, or "current permissions" if you want them to be enforced. In our example, we choose to set the Read permission for Authenticated Users, and specify two conditions: the user department is Engineering, and the device location is Denver. In the syntax of DAC, the claims-based condition reads *((User.department Equals "Engineering") And (Device.location Equals "Denver"))*. Adding these conditions uses drop-down menus similar to those in Figure 7. The results appear in the completed dialog box, shown in Figure 8.

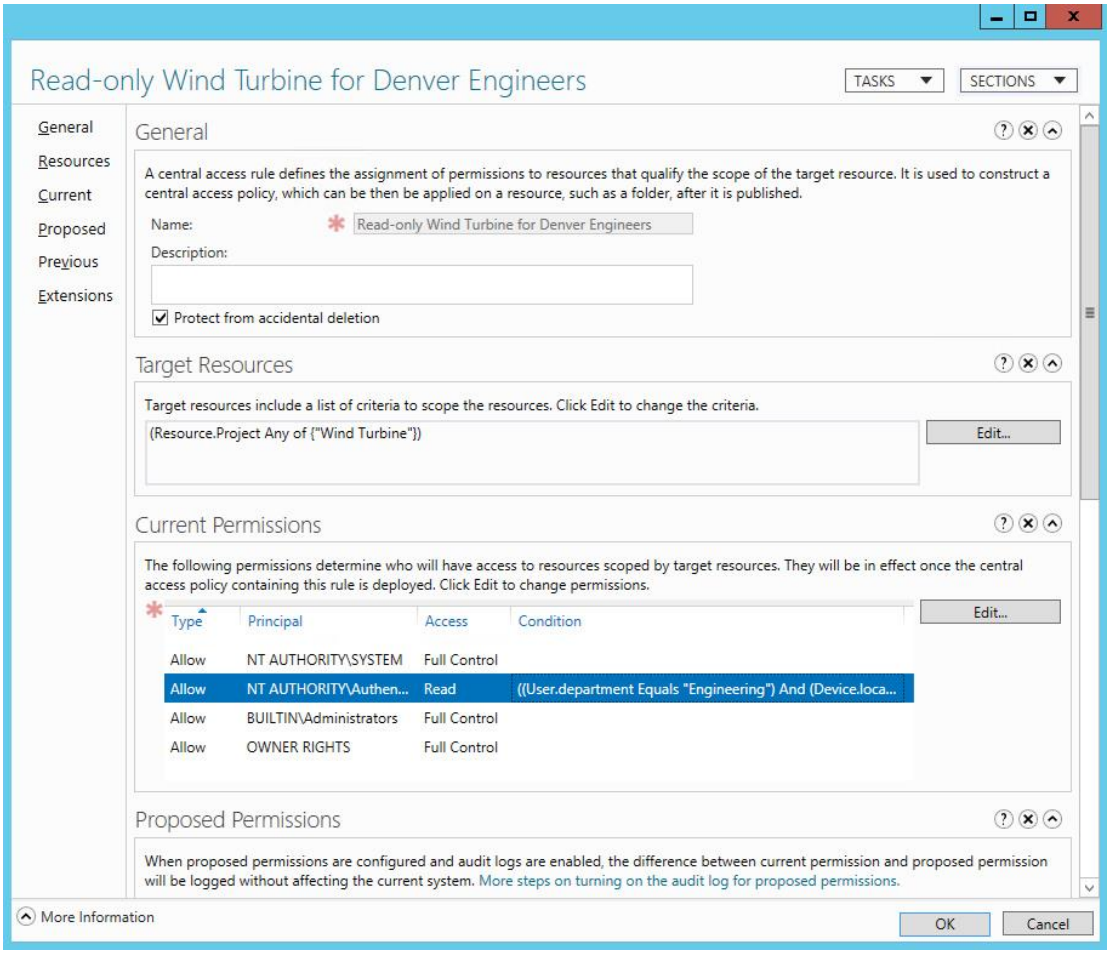

Figure 8: The completed rule, with targeting and permissions configured

B. Create a Policy

In our example, the Central Access Policy (CAP) will contain only one rule. The procedure for adding a CAP is simple: right-click Central Access Policy in ADAC's tree view, choose *New*, then provide a name and add the rule.

C. Configure Group Policy

To make the new Central Access Policy available to file servers, create a GPO and link it to the OU containing your file servers (or to the domain, if you have no such OU; you can use security filtering to limit the GPO to specific file server computers). The relevant GPO setting is *Computer Configuration > Policies > Windows Settings > Security Settings > File System > Central Access Policy*. Right-click *Central Access Policy*, choose *Manage Central Access Policies*, highlight the policy in the left column and click the *Add* button.

D. Apply the Policy

The GPO makes the Central Access Policy available to your file servers, but doesn't actually enforce it. So now you have to open Windows Explorer on your file server, navigate to the properties page of the relevant share, click the *Security* tab, click *Advanced*, click the new *Central Policy* tab, and configure the policy, as shown in Figure 9.

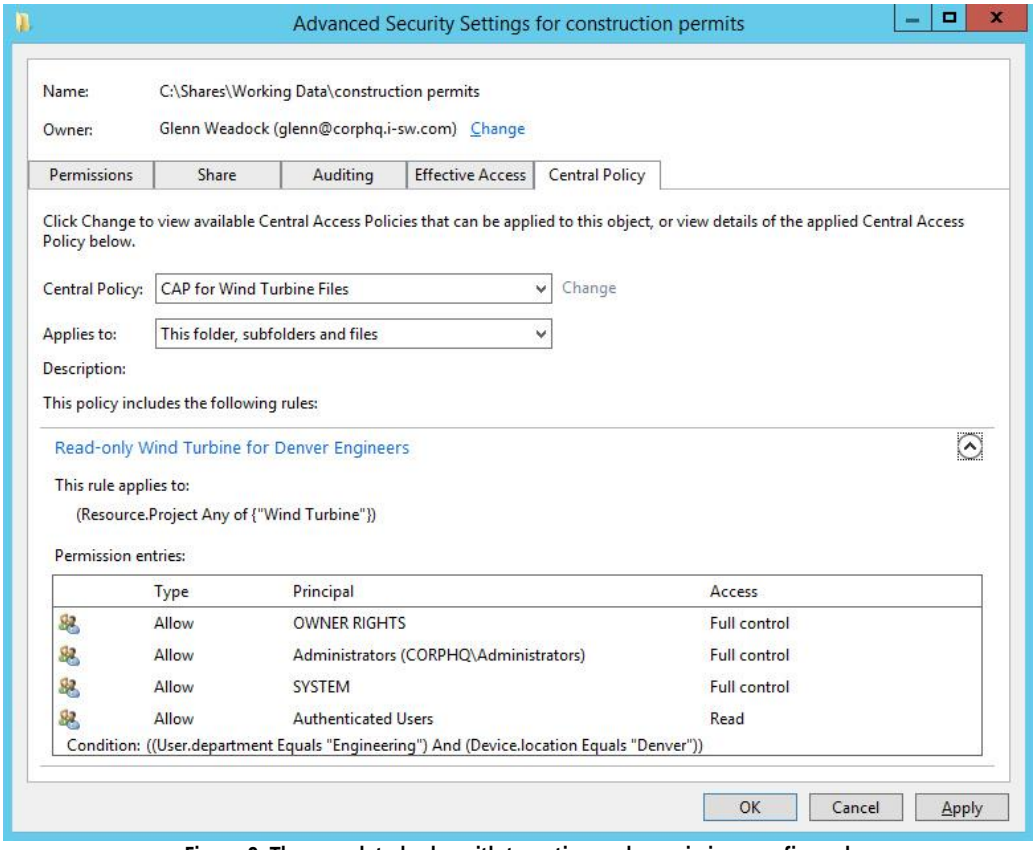

Figure 9: The completed rule, with targeting and permissions configured

The NTFS permissions for the "construction permits" folder give Full Control to authenticated users. So the central access policies, being more restrictive, will control access to the tagged files, based on the user and device claims. Let's see how this would work:

- If you log on as Robert Goddard (whose user attribute "Department" equals "Engineering") to a computer in the Denver location, you will have read-only access to the September and October construction permit documents, as you can see in the Effective Access tab (Figure 10—note the "Access limited by" column).
- If Robert Goddard uses a computer whose location attribute specifies *Golden* instead of Denver, he will be denied all access to the tagged Wind Turbine files; that might be handy, for example, if the Golden office is less secure than the Denver office.
- William Gates, whose user account's department attribute is "Tech Support," can log on to a Denver computer but he won't be able to even read the Wind Turbine files, because his user claim doesn't match the central access policy.

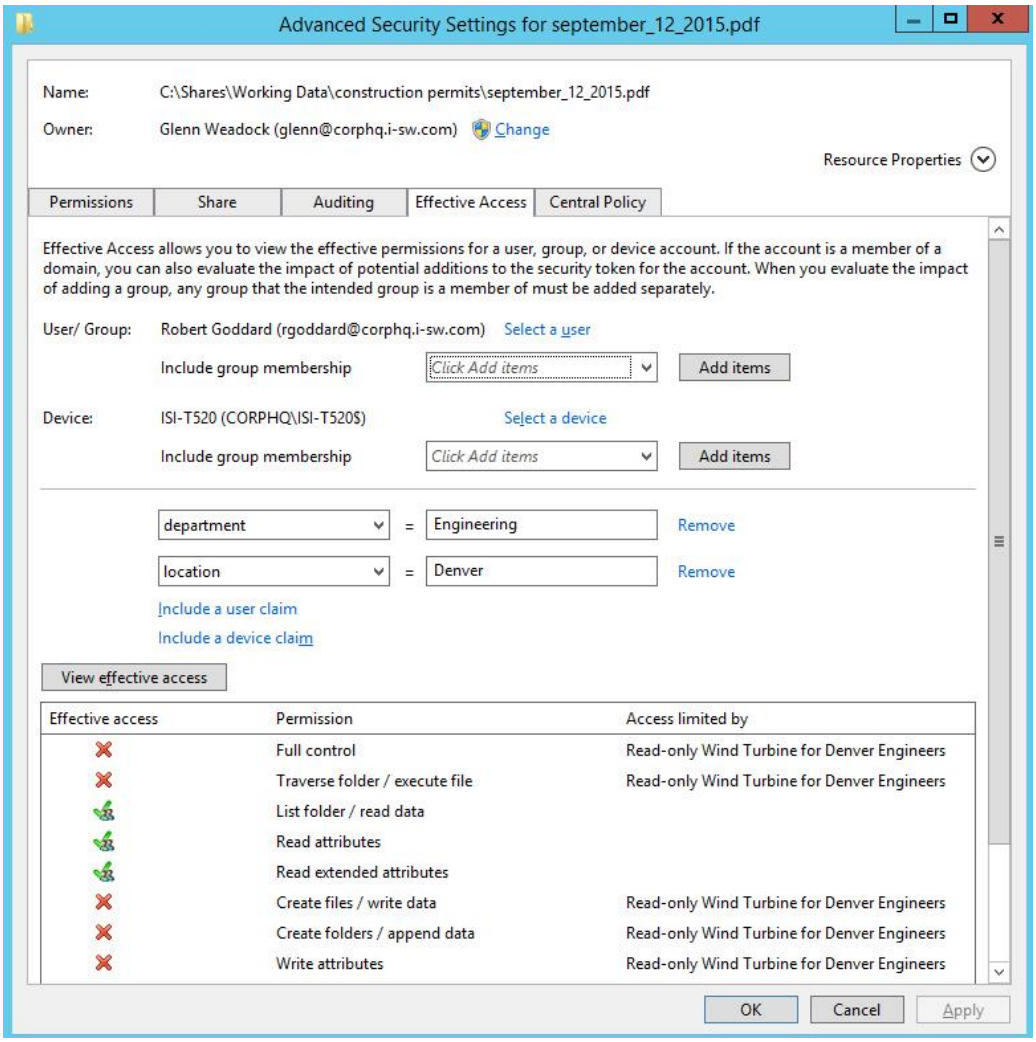

Figure 10: Robert Goddard's access limited by a Central Access Policy

To flesh out this scenario, an administrator could add rules so that engineers in the Phoenix office, which took over the Wind Turbine project, would have full control over the tagged documents. She could also tag the documents pertaining to the Battery Facility project, create rules for those, and so on.

Conclusion

If you've ever found yourself jumping through a series of ever-smaller hoops in order to design security groups and folder hierarchies that let you control file system access the way you want, Dynamic Access Control could be an eminently useful feature. The built-in Active Directory attributes and resource properties are a great start, and eventually you can create your own and build access control rules that match your needs exactly. The promise of the File Classification Infrastructure that got our attention back in 2009 has evolved into a practical reality for building almost any access control scheme you can dream up.

For Further Reading

Here's a sampling of documents and software to take you deeper into this topic:

- Microsoft's "Data Classification Toolkit" (updated in 2014 for Server 2012 R2), which includes tools to assist you in performing file classification and central access policy deployment: *http://www.microsoft.com/en-us/download/details.aspx?id=27123*
- Using Central Access Policies for auditing: *http://technet.microsoft.com/en-us/library/hh831476.aspx*
- Performing file classification with PowerShell: *http://blogs.technet.com/b/filecab/archive/2009/09/14/how-to-do-content-classification-of-files-using-windows-powershell-scripts.aspx*
- Setting up access-denied assistance: *http://technet.microsoft.com/en-us/library/jj574182.aspx*

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

Administering Windows Server 2012 (M20411)

Installing and Configuring Windows Server 2012 (M20410)

MCSA: Windows Server 2012 Boot Camp

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Glenn Weadock is a longtime instructor for Global Knowledge and teaches Windows 8, Windows 7, Vista, Server 2012, Server 2008, and Active Directory. He co-developed with Mark Wilkins two advanced Server 2008 classes in the Microsoft Official Curriculum. Glenn also consults and provides expert witness services in patent infringement cases through his Colorado-based company Independent Software, Inc. (www.i-sw.com).