**Understanding the Differences: Users, Roles, Groups, and Permissions**

If you’ve spent any time working with EMC Documentum, you’re aware of how challenging assigning permissions can be. If you’re new to Documentum, then hold on to your hat: Things are about to get interesting.

In Documentum, each *user* has permissions, each *group* has permissions, and each *role* has permissions. Add to this the many applications that make up EMC Documentum xCelerated Composition Platform (xCP), and suddenly you have permissions associated with actions and processes. Then, there are the permissions you can apply to each document. Soon, the whole system starts to seem security crazy.

Happily, it’s possible to parse out the various security settings and design an xCP security system that is effective and does not require constant attention to function. This article discusses the differences between roles, permissions, and individuals.

**Permissions in xCP**

When designing a permissions system in xCP or any of its components (for example TaskSpace, Content Server, Process Builder), be sure to document everything you do. The result should be a living document that you can revise whenever you make changes to permission systems in xCP. This document should record the basic permission sets issued to users as well as group and role permission sets. Life will be much easier in the long run if you keep a security document that is up to date.

In xCP, the general rule is that a user will have the minimum permissions the system allows. For example, say user John Smith is trying to view a document. John belongs to Group A, which allows access to see the document. But John is also a member of Group B, which is explicitly restricted from seeing the document. The result? John Smith will not be able to see the document. Any time a user cannot see an object in xCP, it’s because a permission setting somewhere is preventing access. There is more information about how permissions are determined in the section “How the xCP Repository Checks Permissions,” later in this article.

When you install an xCP system, you have the option to turn on Repository permissions. Once enabled, each object in xCP will have an associated access control list (ACL) that defines the kind of access all users and processes have to that object. (Remember that everything in xCP is considered an object, including not only documents but also users and processes.) To access any object in the system, the user must either own the object or have permission through an ACL to access the object.

**Permissions for individuals**

Each individual who has access to xCP has an account. That account in turn has an ACL that defines each user’s basic global access. This basic access ACL defines the access a user has to any object in the absence of more specific permission settings. The default global permission level is Read. The basic permission levels of global access are:

* **None.** The user cannot see any object or any object attributes.
* **Browse.** The user can see attributes but no object content.
* **Read.** The user can see the attributes and the content of any object.
* **Relate.** The user can do everything that Read permission allows as well as add annotations.
* **Version.** The user can do everything allowed under Relate permission as well as change the object content, but the user cannot change content           without updating the version and cannot alter attributes.
* **Write.** The user can do everything allowed under Version permission in addition to changing content without updating the version. Write also allows the user to alter object attributes.
* **Delete.** The user can do everything allowed under Write permission as well as delete any object.

You can also assign extended default permissions to individual users. If you grant a user one of these permissions, he or she can perform the function unless restricted by an object-level permission. The extended permissions are:

* **Execute Procedure.** The user can execute external processes.
* **Change Location.** The user can move the location of an object within a repository.
* **Change State.** The user can change the state of a lifecycle object.
* **Change Permission.** The user can modify the permissions of an object.
* **Change Ownership.** The user can change the ownership of an object.
* **Extended Delete.** The user can delete objects but cannot alter the content of object attributes.

There can be some variation in the permissions you assign if a user is a global user with access to multiple repositories in a federation. In such cases, you can assign a user Basic permission by Repository.

**Permissions for groups and roles**

Groups are made up of users, roles, and other groups. They have a permission set that is additive to the user’s individual permission set. Users always have a default group, and in the absence of refining instructions, the permissions defined by that group are applied to objects that a user creates. The group provides its users with access to objects and defines the actions that users can perform on an object. Groups can be either public (created by a superuser or sysadmin) or private (created by a user with Create Group privileges).

A *role* is a type of group that contains users or other groups that are assigned a specific role. Roles—either private or public—provide a means of defining groups that have a particular function within a system. For example, a department name like Technical Writing would be a Group; within that department, each user might have one or more roles, such as writer, content reviewer, editor, publisher, or manager. In addition, each user might belong to one or more other groups, like SOP Team or Change Control Board.

*Dynamic groups* and *module roles* are lists of potential users who need to interact with xCP on a session-by-session basis only—for example, users who are external to the system, such as contractors who require limited access for a project. These users are not granted the Module Role or Dynamic Group permissions until requested by the application.

**Special roles**

Three special roles in xCP allow privileges that no other role conveys: an object owner, Superuser, and Sysadmin. The owner of any object has permissions to change that object. This is true of all types of objects, including documents, users, groups, and roles. He or she always has at least Read permission to any object the owner has created.

Superuser and Sysadmin are default roles in xCP, and both belong to the group *admingroup.* These two roles have specific permissions that no other users in the system share: They grant the maximum possible permissions to objects, and even with restrictive permissions, these users always have at least Read access. Other special privileges include creating and modifying types, users, groups, roles, and permission sets—even if they are marked private; the ability to read annotation objects; registering a type for indexing; creating a site-publishing configuration; and creating and accessing Administrator Access sets without being bound by those rules. However, Superuser and Sysadmin users cannot assign extended privileges to their own accounts or reset their own state.

**Object permission sets**

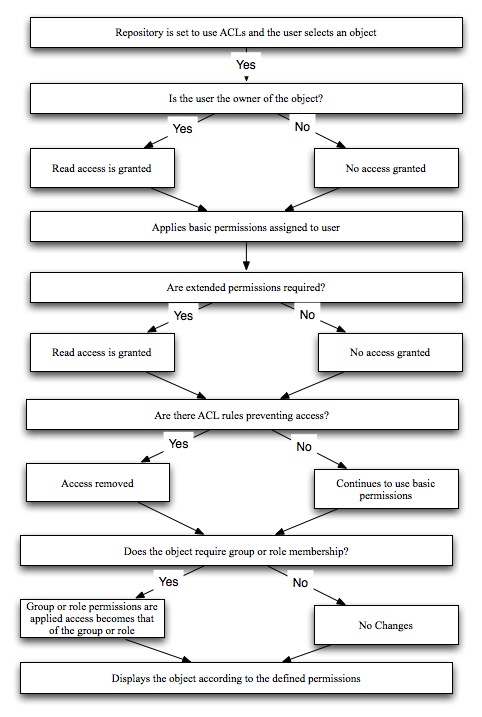
A *permission set* is defined as all the various permissions that apply to a single object. Each object in an xCP system has a permission set, and by default, any object inherits its creator’s permission set. This means that if a user has Write access to a set of documents that no other users can see (that is, permission is set to None), anything that that user creates will have this permission set. In addition to Basic permission functions like Read and Write, actions such as importing, copying, linking files, and starting workflows can be saved in a permission set.

The owner of a document can change the default permissions as long as the owner has permission to make permission changes. When a user changes a permission set or creates a new permission set that is saved as a private permission set (that is, the Class attribute is set to Regular), it can be applied by that user to other objects in xCP.Any repository user can create and use a public permission set. However, public permission sets can be deleted or modified only by the permission set owner, the repository owner, or a user with the Superuser or Sysadmin role.

**How the xCP repository checks permissions**

It is important to understand how the system checks for object access. Permissions to individual objects are detected in the following order (see Figure 1):

1. Is the user the owner? If so, the user is granted default Read access to the object, and xCP then escalates that privilege depending on certain rules.
2. Are basic permissions assigned to the individual? If a user has Write access as a basic permission, then xCP assumes that the user will have Write access to the object.
3. Check extended permissions, if necessary. (xCP checks extended permissions only if the request requires them, such as moving an object.)
4. Do no-access restrictions exist? (xCP looks for anything that explicitly removes access from the user—for example, if that individual or a group to which the user belongs is specifically given the basic permission None. If the user is the document owner, the access will become Read.)
5. Does the object require any group (or role) membership? (xCP checks to see whether the object requires membership in one of more groups. If the user is in the required group(s), his or her access becomes whatever is assigned to the group.)
6. Grant access as defined in the above steps.

[](https://community.emc.com/servlet/JiveServlet/showImage/102-8899-3-19102/permissions+flow+final.jpg)

***Figure 1.*** *Repository permission check flowchart*

**Putting it all together**

This article provided you with a quick reference to how users, groups, roles, and permission sets interact in xCP. When you understand the permission architecture, it becomes increasingly easy to create (and maintain) the fewest possible number of security-related objects. The more streamlined and concise a permission system is, the easier it will be maintain, troubleshoot, and upgrade your environment.

In addition to users, groups, and roles, a variety of additional security and permission setting are available at all levels of xCP. To learn more, be sure to check out the *Content Server Administration Guide and Documentum Content Server Fundamentals Guide.* Also, see the companion article [Basic Documentum xCP Permission Troubleshooting](https://community.emc.com/docs/DOC-9319).