grass valley

APPLICATION NOTE

Aurora Edit Security

Controlling asset visibility and access

Patrick Thompson, *Senior Software Engineer*

January 2011

Using the Aurora Edit Security feature, you can control which users and groups have which permissions (read, write, delete, etc.) on which assets in the Aurora Edit bin tree.

www.grassvalley.com

Introduction

With Aurora™ Edit Security from Grass Valley™, you can control the visibility and access for users and groups working within Aurora Edit bins by controlling the file system permissions for the bins and assets. Aurora Edit Security uses the overlapping modes of inheritance, exclusivity, and group membership as implemented by Windows Active Directory (AD) to establish and enforce asset security. These principles apply:

- **Selective access.** You create groups of users, such as Editors, Producers, and Interns, and set permissions for each group.
- **Partial control.** You control access to subbranches of the bin tree for users and groups.
- **Administrative control.** The administrator has exclusive access to a tool in the top-level bin that allows the setting of permissions on the bin-tree root. At all levels of bins (as a feature of Active Directory), you control who can control access.

Technical Background

As part of its open architecture, the Aurora suite of products stores media on Windows-compatible file system volumes, notably the Grass Valley K2 Summit™ shared storage system. These volumes support Windows Active Directory; thus, Aurora Edit is able to leverage Active Directory, particularly in a large, multi-user, domain-controlled environment, to effect fine-grained access control of the Aurora Edit assets, including master clips, subclips, sequences, graphics, and bins. (Subclips and sequences are controlled by their containing bins.) Aurora Edit security is essentially the application of Active Directory controls to Aurora Edit assets.

Example

As an elementary example, suppose that your organization has the bins and groups shown in the table below. The day-named and Investigative bins are sub-bins of the top-level Work in Progress bin.

Read, Write, and Delete permissions are abbreviated to R, W, and D. Permissions are set on top-level bins and are allowed to automatically flow by inheritance (indicated by parentheses) to descendent bins. (Active Directory permissions and inheritances are in fact more nuanced, but they can be effectively discussed as RWD.)

Not listed here are several user members in the groups. In particular, Bob (a member of group Editors) and Alice (a member of group Producers) are working exclusively on a secret investigative report. On the Investigative bin, inheritance is blocked such that no user automatically has access to the bin; permission must be explicitly granted, and it is only for Bob and Alice, who both enjoy full RWD privilege. Note that in other bins, Bob's and Alice's permissions are automatically established by their group membership, such that permissions for these (or any other) individual users need not be explicitly set.

Example (cont.)

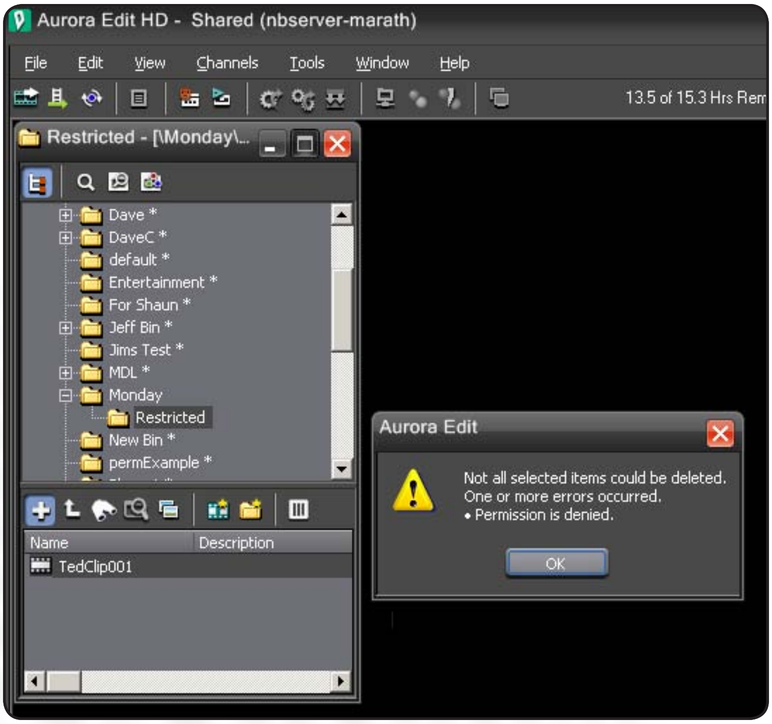
Bin		Group Permissions				User Permissions	
		Editors	Producers	Interns	Archivists	Bob	Alice
Work in Progress		RWD	RW	R	R	(RWD)	(RW)
	Monday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Tuesday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Wednesday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Thursday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Friday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Saturday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Sunday	(RWD)	(RW)	(R)	(R)	(RWD)	(RW)
	Investigative	[none]	[none]	[none]	[none]	RWD	RWD
Feeds		RW	RW	R	RWD		
Hold for Release		RW	RWD	R	RWD	(RW)	(RWD)
Archive		RW	RW	R	RWD	(RW)	(RW)

Some example effects of this schema:

- No one but Bob or Alice may access the Investigative bin.
- Interns may browse everything (except for the Investigative bin) but they cannot record or delete anything.
- Only Editors, Producers, and Archivists can record into Feeds.
- Editors and Producers can push material into the Archive folder, but only Archivists can remove it.
- Assuming inheritance isn't blocked, users who have W permission within a bin can create working sub-bins to an arbitrary depth.

Note that the bin, group, and user designations in this example are not imposed by Aurora Edit; these entities are chosen by the user. Active Directory, and in turn Aurora Edit, support whatever schema your organization requires.

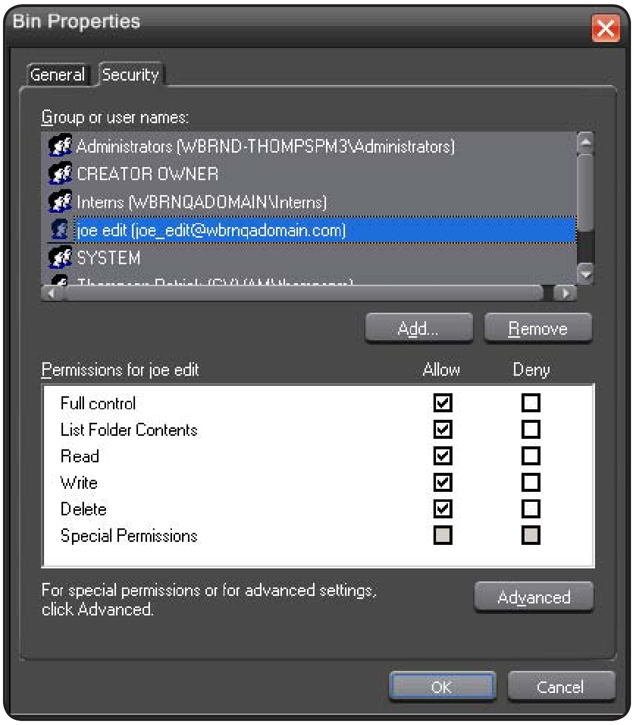
Security in action: In the screen capture below, user Joe Intern, a member of the Interns group, has attempted to delete a sub-clip within the Monday bin and been denied permission.



Not all selected items could be deleted.
One or more errors occurred.
• Permission is denied.

Setting Permissions

The most direct way to set permissions on an object in the bin tree is through Aurora Edit, as shown in the examples below. Aurora Edit passes the Windows Security property sheet (the Security tab you would see for a file in Windows Explorer) to the user within the bin's properties dialog. Here we see that user Joe Edit has been explicitly granted full control on a bin.

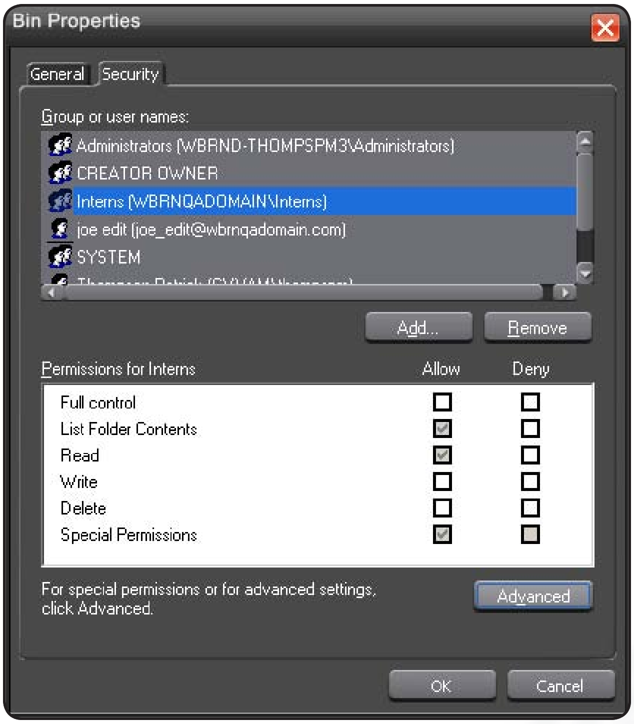


Group or user names:

- Administrators (WBRND-THOMPSPM3\Administrators)
- CREATOR OWNER
- Interns (WBRNQADOMAIN\Interns)
- joe edit (joe_edit@wbrnqadomain.com)
- SYSTEM

Permissions for joe edit	Allow	Deny
Full control	☑	□
List Folder Contents	☑	□
Read	☑	□
Write	☑	□
Delete	☑	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.



Group or user names:

- Administrators (WBRND-THOMPSPM3\Administrators)
- CREATOR OWNER
- Interns (WBRNQADOMAIN\Interns)
- joe edit (joe_edit@wbrnqadomain.com)
- SYSTEM

Permissions for Interns	Allow	Deny
Full control	□	□
List Folder Contents	☑	□
Read	☑	□
Write	□	□
Delete	□	□
Special Permissions	☑	□

For special permissions or for advanced settings, click Advanced.

Another entry on the same bin shows that the group Interns has inherited read-only permission. (Inheritance is indicated by the gray-shaded checkbox.)

The assets in Aurora Edit's bin tree associate with folders in the file system. As an alternative to setting permissions with Aurora Edit, you can set permissions directly on the associated folders using, for example, Windows Explorer, or any administrative application.

Typically, during configuration, organizations establish an inheritance and group hierarchy for assets and users such that permissions do not require constant management; the desired access control happens automatically because of where the asset is located and who is using it, not because every new asset has had its permissions set manually.

Configuration

Careful configuration is the key to care-free enjoyment of Aurora Edit Security. Inheritance and grouping provide nearly automatic access control. The Aurora Edit installation manual details how to set up the security option. Briefly, you would follow these steps:

Design a security schema appropriate for your organization. This is a hierarchy similar to that shown in the example table to describe which users and groups have access to which assets. This can be very simple, with just a few users and groups, or very complex, with hundreds of users and groups. Active Directory can support arbitrarily intricate hierarchies. Before configuring security on Aurora Edit, most organizations already have a domain schema designed and supported by their IT departments.

Establish an Active Directory domain and join the Aurora equipment to the domain. Organizations often already have configured domains that may consist of AD trees or forests, so this step reduces to the straightforward task of joining the Aurora machines to the domain. If the infrastructure does not already exist, a domain and domain controller can be configured for the work cluster that uses Aurora.

Modify service users as necessary for the domain. Depending on how the Aurora equipment was originally deployed, i.e., depending on whether it was born onto a workgroup or a domain, several services, e.g., SmartBins, Advanced Encoder, and Conform Server, may require a user change.

Enable Windows Security on the K2 Summit shared storage system. The default deployment of K2 Summit shared storage is to have Windows Security disabled. With simple changes to two configuration files and a system-wide reboot, the system will reawaken ready to support permissions. (Note: This is not a destructive change—existing assets are preserved.)

Set permissions on the file system according to your schema. This can be done efficiently by exploiting the features of groups and of inheritance. The setting can be executed on Aurora Edit, or by using any Windows tool that can set file system permissions on the shared volume.

The Aurora suite shares media with K2 Summit using a sophisticated system of hard links. In specifying how permissions will be set on file system assets, you must consider the interaction between Aurora and K2 Summit; there are several operational approaches you can take. A common approach establishes monolithic security on the K2 Summit file system branch such that permissions for the K2 Summit aspect are controlled via K2 Summit applications and APIs; then fine-grained control of the Aurora aspect is handled through the Aurora Edit application.

Summary

Several scenarios worry production organizations that use massive, centralized media storage. Many concerns involve security, for example, accidental deletion of a substantial number of assets, or inappropriate access to restricted material. Using the Aurora Edit Security feature, you can militate against these misfortunes. To address the large and small scope of similar concerns, Aurora Edit Security provides fine-grain, scalable, Windows Active Directory-compatible control of Aurora assets.

Please refer to the *Aurora Edit and LD Installation Manual* for more detail about the Security feature and its configuration.

SALES

Local and regional sales contacts can be found by visiting
www.grassvalley.com/sales

SUPPORT

Local and regional support contacts can be found by visiting
www.grassvalley.com/support

grass valley