

# What does it mean when the Advanced Security Settings dialog says that an ACE was inherited from "Parent Object" without naming the specific parent?

 [devblogs.microsoft.com/oldnewthing/20150505-00](http://devblogs.microsoft.com/oldnewthing/20150505-00)

May 5, 2015



Raymond Chen

The Advanced Security Settings dialog shows the ACEs in an object's ACL, and one of the pieces of information is a column labeled *Inherited from* which identifies whether the ACE was inherited, and if so, from where. A customer observed that when they opened the Advanced Security Settings dialog, one of their objects had an ACE that showed *Parent Object* as the *Inherited from*.

Name: C:\dir1\dir2\dir3\dir4\file

Type	Principal	Access	Inherited from	Applies to
Allow	Administrators	Full control	None	This folder only
Allow	Administrators	Full control	C:\dir1\dir2\	This folder, subfolders
Allow	SYSTEM	Full control	C:\dir1\dir2\	This folder, subfolders
Allow	CREATOR OWNER	Full control	C:\dir1\dir2\	Subfolders and files on
Allow	Users	Read & execute	C:\dir1\dir2\	This folder, subfolders
Allow	Users	Special	C:\dir1\dir2\	This folder and subfold
Allow	Authenticated Users	Full control	Parent Object ←	This folder, subfolders

However, when they went to the parent object `C:\dir1\dir2\dir3\dir4`, that ACE is nowhere to be found.

Name: C:\dir1\dir2\dir3\dir4

Type	Principal	Access	Inherited from	Applies to
Allow	Administrators	Full control	None	This folder only
Allow	Administrators	Full control	C:\dir1\dir2\	This folder, subfolders and
Allow	SYSTEM	Full control	C:\dir1\dir2\	This folder, subfolders and
Allow	CREATOR OWNER	Full control	C:\dir1\dir2\	Subfolders and files only
Allow	Users	Read & execute	C:\dir1\dir2\	This folder, subfolders and
Allow	Users	Special	C:\dir1\dir2\	This folder and subfolders
Allow	Everyone	Full control	C:\dir1\dir2\	This folder, subfolders and

How can an ACE be inherited from its parent, when it doesn't exist in the parent? The Advanced Security Settings dialog is trying to be helpful, but in doing so, it implies a greater level of confidence than it actually offers. ACEs do not specify where they were inherited from. There is merely a bit in the ACE called `INHERITED_ACE` which means, "This ACE was created via inheritance." Not only does this bit not tell you where the ACE was inherited from, but the bit might even be wrong! Anybody can go in and toggle the bit, and bingo, you now have forged the "I was created via inheritance" flag. Another way this flag could be out of sync is if the user started an ACL update operation and then canceled it partway through. The Advanced Security Settings dialog uses the `GetInheritanceSource` function to determine the source of each ACE. That function walks up the parent chain looking for matching inheritable ACEs. If a match is found, then the Advanced Security Settings dialog shows that parent as the *Inherited from*. Otherwise, it shrugs its shoulders and says *Parent Object*. The string *Parent Object* means "This ACE claims to have been inherited from somewhere, but I can't figure out where, so I'm just going to be vague and say that it came from some parent object somewhere." Perhaps a less confusing string would have been *Ancestor Object* or even simply *Unknown*.

The Advanced Security Settings dialog figured that it would go the extra mile and instead of merely saying *Inherited = Yes*, it would try to find a parent object that was the most likely source of the inheritance. But by doing that, you came to expect it, and then you got upset when it wasn't able to come through for you. No good deed goes unpunished.

[Raymond Chen](#)

**Follow**

