How new IFS files get assigned object authorities

Issue

When a PHP script writes a new file to the IFS, the object authorities for the owner of the file may not be what the author of the script expected. Subsequent operations performed on the new file may fail because the owner does not have the expected object authority. This article tells how the owner's object authorities are assigned to new IFS files. This information can help insure that new files have the needed permissions.

> **Note: This article is in reference to object authorities, not data authorities**
>
> Data authorities control Read, Write, and Execute permissions. Object authorities include Exist, Mgt, Alter, and Ref. For a brief description of these, see the help text for the WRKAUT command, excerpted in the Details section below.

Environment

All versions of IBM i.

Resolution

This information was taken from a document published by IBM Support:

Integrated File System Authority Considerations

This is a very good document to read to understand the ins and outs of authorities on the IFS. I recommend it highly. Go read it now, if you can. You can also find much of the same information in this documentation page:

Planning integrated file system security

Data authorities and object authorities are listed below, but just briefly things you cannot do to an object without the proper object authorities include delete, move, and rename. There are more, but these are the kinds of things most folks are surprised they cannot do to an object using the object owner's profile.

The owner of an IFS file is NOT automatically given the object authorities. They have to be granted somehow. When a new file is created, the creator is typically the owner, but the owner's object authorities are assigned in a kind of strange way.

The object authorities given to the owner of a new file are the same as the object authorities given to the owner of the containing folder, even if that is a different profile.

This is a little different than what some of us (myself included) might imagine. We might imagine that whatever object authorities the new file owner has in the containing folder would apply to the new object. We could not be more wrong.

Consider this example, basically copied from the above IBM document, but altered a bit to be more applicable (we use QTMHHTTP instead of BOB).

Here are the object authorities for a directory called /rjzeller:

```
Object . . . . . . . . . . . . . . . :    /rjzeller
       Owner . . . . . . . . . . . . :     RJZELLER
       Authorization List. . . . . . :     *NONE

                         User:     Object Auth:
Object authorities - *PUBLIC    *NONE
                     QTMHHTTP   *ALL
```

You will recognize the QTMHHTTP profile as the default Apache profile that often is the profile running PHP scripts. Imagine a PHP script writes a PDF file called mydoc.pdf into this directory. PHP was running as QTMHHTTP when the PDF was created, so QTMHHTTP owns /rjzeller/mydoc.pdf. A subsequent script attempts to delete /rjzeller/mydoc.pdf, but gets an error. Why?

This is because RJZELLER, the owner of /rjzeller, does not have any object authorities to /rjzeller. They were never granted. So, the owner of any file created in /rjzeller will not have any object authorities to that file. In this case, the owner of /rjzeller/mydoc.pdf is QTMHHTTP, and so QTMHHTTP has no object authorities to the file, despite being the owner. Even though QTMHHTTP has all object authorities to the containing folder /rjzeller, it does not matter. It only matters what object authorities RJZELLER has to /rjzeller, because RJZELLER is the owner of the folder.

So, with no Exist authority, QTMHHTTP is not allowed to delete the /rjzeller/mydoc.pdf file, and the PHP script fails.

To avoid this situation, the owner of /rjzeller could be given all object authorities to the /rjzeller folder:

```
CHGAUT OBJ('/rjzeller') USER(RJZELLER) OBJAUT(*ALL)
```

Any new files created in this folder will have all object authorities granted to their owners. So, if QTMHHTTP creates a file in this folder, QTMHHTTP will be able to delete, rename, or move the file later. This is the typical expected behavior.

Details

Some basic descriptions of the authorities, copied out of the help text for the WRKAUT command:

```
Data Authority
         The data authority that the user has to an object.  Several
         different system-defined data authority levels may be assigned to
         users.  The levels are:
         *RWX
             Allows all operations on the object except those that are
             limited to the owner or controlled by the object rights.
         *RX
             Allows access to the object attributes and use of the object.
             The user cannot change the object.
         *RW
             Allows access to the object attributes and allows the object to
             be changed.  The user cannot use the object.
         *WX
             Allows use of the object and allows the object to be changed.
             The user cannot access the object attributes.
         *R
             Allows access to the object attributes.
         *W
             Allows the object to be changed.
         *X
             Allows the use of the object.
         *EXCLUDE
             All operations on the object are prohibited.
         *NONE
             Displayed by the system when the user does not have any data
             authorities.
         USER DEF
             Displayed by the system when the specific data authorities do
             not match any of the predefined data authority levels above.
             You can see the specific data authorities by using the "display
             detail" function key.
         The value *AUTL is also valid when public authorities are being
         defined.  It indicates that the public authority specifications in
         the authorization list used by this object should be used to
         determine public authority.

     Object Authorities
         The object authorities that the user has to the object.  An "X" in
         the column indicates that the user has the specified object
         authority to the object named.  The specific object authorities
         are:
         Exist
             Object existence authority provides authority to control the
             object's existence and ownership.
         Mgt
             Object management authority provides authority to specify
             security, to move or rename the object, and to add members if
             the object is a database file.
         Alter
             Object alter authority provides authority to change the
             attributes of an object, such as adding or removing triggers
             and adding members for a database file.
         Ref
             Object reference authority provides authority to specify the
             object as the first level in a referential constraint.
```