Improved Security Controls Open Door to DB2 for i Tool Usage

System iNEWS Magazine

Scott Forstie
Scott Forstie

Sat, 2011-01-01 (All day)

The DB2 for i product includes industry leading tools for database administration and performance tuning. Given the sensitive nature of data and administration tasks, many of the useful and important DB2 for i tools have been restricted to certain users. Until recently, the security officer had to grant *JOBCTL (job control user special authority) to allow users to use these tools. Since *JOBCTL authority allows a user to change many important system settings using work management controls, the security officer may not grant the required authority for database administrators, performance analysts, or application development teams. Now, those same security officers have an alternative to consider in the form of DB2 for i function usage. In basic terms, QIBM_DB_SQLADM function usage is a granular alternative to *JOBCT. If you have permission to use QIBM_DB_SQLADM function usage, you are allowed to use those same database tools. The other DB2 for i function usage identifier is QIBM_DB_SYSMON, which can be used to grant access to a small subset of the functions guarded by QIBM_DB_SQLADM.

This article provides an in-depth description of security function usage and the database tools which honor function usage as an alternative security control.

The addition of security function usage IDs on DB2 for i give the security officer a low risk alternative for letting database users use database tools and techniques, without giving them the master key to the machine.

The DB2 for i security function usage support described in this article was shipped in the base IBM i 7.1 operating system and System i Navigator 7.1 client. The host side support has been provided via a PTF back to IBM i 6.1. By providing support in IBM i 6.1, a large IBM i install base can immediately take advantage of the new controls.

Security Function Usage

Function usage is a powerful approach to implement a granular security model. Granularity is better for the users because they’re allowed to use things that are consistent with their job or role in the organization. A granular approach to implementing security might sound bothersome for the security officer, but rest assured that IBM i’s security function usage is easy to understand and implement.

Function usage support on IBM i has been in place for many releases, providing an alternative security control for Job Watcher, Tivoli administration, Backup Recovery and Media Services for AS/400 (BRMS), and other components. If function usage is a new topic for you, take a couple of minutes to review the non-DB2 for i possibilities.

The Work with Function Usage (WRKFCNUSG) and Change Function Usage (CHGFCNUSG) are the green screen commands that are used to see the function usage choices and the provided access to specific functions. A naming scheme is used to provide a logical grouping of functions. To use these commands, the user needs to have Security administrator (*SECADM) user special authority. Figure 1 shows the first screen that appears for the WRKFCNUSG command.

Database related function names begin with QIBM_DB_. The first two DB2 for i functions are

QIBM_DB_SQLADM and QIBM_DB_SYSMON and they roughly correspond Database Administration and Database Information tasks. As you'll see later in this article, QIBM_DB_SQLADM-enabled users can do many things, while QIBM_DB_SYSMON-enabled users can only observe the database. If you've never heard of function usage before, take a moment to scroll through the different controls.

By default, any user with All object (*ALLOBJ) user special authority will be able to use any DB2 for i feature protected by QIBM_DB_SQLADM and QIBM_DB_SYSMON. The CHGFCNUSG (Change Function Usage) command can be used to remove *ALLOBJ as an avenue to function usage capability.

New Authority Options for SQL Analysis and Tuning

DB2 for i contains a rich set of commands, stored procedures, APIs and tools for analysis and tuning of the performance aspects of database applications. Figures 2 and 3 give a brief glimpse of where to find the tools. Figure 2 shows how an SQL Plan Cache snapshot can be mined for information via the Show Statements action. Figure 3 shows how to launch into the SQL Details for Jobs dialog, which can be used to "look into" active jobs and understand useful SQL information. Many of the features relevant to this article are found within the SQL Performance Monitors and SQL Plan Cache (Snapshots & Event Monitors) portion of System i Navigator.

Prior to IBM i 7.1, a system security officer would need to grant *JOBCTL user special authority to enable database analysts and database administrators to use the database tools. Since *JOBCTL authority allows a user to change many system critical settings that are unrelated to database activity, it was not an easy decision for security officers to grant this authority. In many cases, the request for *JOBCTL was not granted to database analysts, thus prohibiting the use of the full set of database tools.

New in 7.1, the security officer has additional capability to authorize access to database analysis tools and the SQL Plan Cache. DB2 for i will now take advantage of the function usage capability available in the operating system. A new function usage group called QIBM_DB has been created. In 7.1, there are two function IDs in the QIBM_DB group:

- QIBM_DB_SQLADM (IBM i Database Administrator tasks)
- QIBM_DB_SYSMON (IBM i Database Information tasks)

The security officer now has flexibility to grant authorities by either; granting *JOBCTL special authority or authorizing a user or group to the IBM i Database Administrator Function through Application Administration in System i Navigator of IBM Systems Director Navigator for i. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_DB_SQLADM, can also be used to change the list of users that are allowed to perform Database Administration operations. The function usage controls allow groups or specific users to be specifically allowed or denied authority. The CHGFCNUSG command also provides a parameter which can be used to grant function usage authority to any user that has *ALLOBJ user special authority (e.g. ALLOBJAUT(*USED)).

The Database Administrator function is needed whenever a user is analyzing and viewing SQL performance data. Some of the more common functions are displaying statements from the SQL Plan Cache, analyzing SQL Performance Monitors and SQL Plan Cache Snapshots, and displaying the SQL details of a job other than your own.

The database administrator function usage is an alternative to granting *JOBCTL, but it doesn't replace the requirement of having the correct object authority. To enable database administrator tasks which are unrelated to performance analysis, refer to the specific task for details on the authorization requirements. For example, to allow an administrator to reorganize a table, they will need to have object authorities granted, which are not covered by QIBM_DB_SQLADM.

The Database Information function provides much less authority than Database Administrator. The primary use is to allow a user to examine high-level database properties. For example, a user that does not have

*JOBCTL or QIBM_DB_SQLADM, could be allowed to view the SQL Plan Cache properties if granted authority to QIBM_DB_SYSMON. Given the limited amount of functionality that is available via QIBM_DB_SYSMON, its primary purpose could be related to someone who is being trained to use the full DB2 for i complement of tools, but isn't prepared to have full access.

Database Function Usage Feature List

Table 1 (part 1, 2, and 3) lists the DB2 for i features which have function usage authorization. This table includes an indication of which SQLnnnn or CPFnnnn error message identifier is returned when the user lacks sufficient authorization. In some cases, the messages differ slightly between 6.1 and 7.1. Many of the features can be used directly via a command or SQL procedure or graphically through the System i Navigator client. The 'User Action' column indicates the System i Navigator user action first and follows with the matching non-Navigator interface (e.g. API).

Note the Multitudes

A special case exists when user profiles appear under the function usage more than once. Since a user can belong to one or more group profiles, the function usage specification could refer to the user in multiple ways for a single function usage.

First, if multiple group profiles, for which a user belongs to, are referenced under the function usage and the user profile is NOT referenced under the function usage, the user will be granted function usage if ANY of the group profiles listed have USAGE(*ALLOWED).

In our example as shown in the sidebar, "Security Function Usage Configuration Example," user MARKA belongs to two groups, PERFTEAM and DBATEAM. If one group profile was configured with usage *DENIED while the other group profile has *ALLOWED, MARKA would be granted function usage.

All a user needs is for one of their group profiles to have permission to use the function.

Second, if one or more group profiles, for which a user belongs to, are referenced under the function usage and the user profile IS explicitly referenced under the function usage, the explicit user reference will determine whether the user will be granted function usage.

Again, user MARKA belongs to two groups, PERFTEAM and DBATEAM. If both groups were configured with usage *ALLOWED and user profile MARKA was had function usage *DENIED, MARKA would not be granted function usage.

Individual function usage references take precedence over group profile references.

Configuring function usage with System i Navigator

To work with QIBM_DB database group function usage from System i Navigator, follow these steps:

- Launch Application Administration as shown in Figure 2.
- Expand the i5/OS and Database folders under the Host Applications tab as shown in Figure 3.
- Customize the Database Administrator (QIBM_DB_SQLADM) function usage as shown in Figure 4.

In this example, the security officer determined that they wanted to use two group profiles, named Dbateam and Perfteam, to manage the user profiles who should be allowed to use the DB2 for i tools. Also they explicitly wanted to deny access to Tester1. Tester1 is a member of the Perfteam group, but should not be allowed to use the tools. As you can see, the security officer has a convenient and easily monitored place to view and authorize users to these functions. Figure 5 shows an example authorization failure. Figure 6 shows the QIBM_DB_SQLADM function usage settings, while Figure 7 shows an authorization failure.

Database Security (Object and Action)

Database security is largely object based. Every object has its own object access authorization setting. The authorization settings can include user, groups and special values such as *PUBLIC. Function usage has no intersection or impact on object authorization. Function usage is usually tied to functions and features. A specific function may have both function and object authorization requirements. A good example of the interplay between object and function usage authorization would be SQL Performance Monitor (aka Database Monitor).

To start an SQL Performance Monitor over multiple jobs requires *JOBCTL or QIBM_DB_SQLADM authority. The STRDBMON command refers to an output file location. The command requires that the user have *CHANGE authority to the library (DBMONLIB). If the file (SLFMON1) exist, the user also needs *ALL authority to the file.

Even if the user possesses *JOBCTL / QIBM_DB_SQLADM authority, they will observe an authorization failure if they don't have the appropriate object level authorities.

```
STRDBMON OUTFILE(DBMONLIB/SLFMON1) JOB(*ALL/*ALL/QRWTSRVR) TYPE(*DETAIL) FTRUSER(SCOTT)
```

For the reasons explained above, I think it's a best practice to use a dedicated and authorization controlled library for all SQL Performance Monitors and SQL Plan Cache dumps and snapshots.

Here is one way to set up a library for this purpose, where PERFTEAM and DBATEAM are group profiles setup and maintained by the security officer:

```
CRTLIB DBMONLIB
RVKOBJAUT OBJ(DBMONLIB) OBJTYPE(*LIB) USER(*PUBLIC)
GRTOBJAUT OBJ(DBMONLIB) OBJTYPE(*LIB) USER(PERFTEAM) AUT(*CHANGE)
GRTOBJAUT OBJ(DBMONLIB) OBJTYPE(*LIB) USER(DBATEAM) AUT(*CHANGE)
```

Audit Records

Audit records are an important component of any security model. An audit record contains details of instances where a user action (Command, API, etc) was denied due to lack of sufficient authority. There are two types of audit records that relate to security function usage:

- Changes to the function usage configuration. (Successful CHGFCNUSG command use)
- Function usage authorization failures. (Failed attempts to use functions protected by a specific function usage identifier)

Here's an example: User SCOTTF has *SECOFR authority. User SLFUSER has *PGMR authority. User SCOTTF does the following:

```
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(SLFUSER) USAGE(*ALLOWED)
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(SLFUSER) USAGE(*DENIED)
```

(Both operations succeed because SCOTTF has authority to change function usage.)

User SLFUSER does the following:

```
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(SLFUSER) USAGE(*ALLOWED)
STRSQL and set current degree = '2'
```

(Both operations fail because SLFUSER does not have *SECOFR authority and has been denied QIBM_DB_SQLADM function usage.)

The failure to use CHGFCNUSG by user SLFUSER is not logged to an audit entry. The first two audit entries show that SLFUSER has usage authority to QIBM_DB_SQLADM allowed at first, and then changed to deny use. The audit entries with *CHGUSAGE indicate the successful use of CHGNFCNUSG.

The third audit entry shows that SLFUSER was denied use of a function which required QIBM_DB_SQLADM function usage, as indicated by *USAGEFAILURE in the journal entry.

In both cases, the General purpose audit record (Entry Type - 'GR') format is used.

Adopted Authority

Adopted authority does not affect the outcome of a security function usage authorization check. Programs and service programs that are built with USRPRF(*OWNER) run with the authorizations of program owner. However, function usage is not included when adopted authority is used. The result of the function usage authorization check will be based solely on the current user profile of the job or thread.

Service Level Detail

To fully utilize DB2 for i security function usage, the System i Navigator client needs to be upgraded to 7.1. This is true even if the server side is running with IBM i 6.1 because pre-7.1 versions of System i Navigator include some client-side enforcement of *JOBCTL user special authority.

The enabling IBM i 6.1 PTFs are included with DB Group PTF SF99601 (Version #15) which was released in October, 2010. IBM i 7.1 users have this support enabled in the base release of the operating system. (IBM encourages database customers to always apply the most recent Database Group PTF.)

More Power Tools on Tap, Safe and Secure

If you act as the security officer, deploy DB2 for i function usage to enable users to take full advantage of IBM i features. If you aren't the security officer, use this article to lobby your security officer to implement function usage controls to improve IBM i efficiency and utilization, without introducing security risks.

Sidebar: Security Function Usage configuration example

To better understand how function usage authorization could be configured, an example is provided below. Some of these user and profile names are used elsewhere in this article. Let's see how to configure DB2 for i function usage. Notice that none of the user's have *JOBCTL user special authority.

Create the profiles as before, but none of them are given *JOBCTL user special authority.

```
CRTUSRPRF USRPRF(SCOTT) PASSWORD(PASSW0RD) USRCLS(*PGMR) SPCAUT(*SERVICE)
CRTUSRPRF USRPRF(MARKA) PASSWORD(PASSW0RD) USRCLS(*PGMR) SPCAUT(*SERVICE)
CRTUSRPRF USRPRF(JIMF) PASSWORD(PASSW0RD) USRCLS(*USER) SPCAUT(*ALLOBJ *SAVSYS)
CRTUSRPRF USRPRF(TESTER1) PASSWORD(PASSW0RD) USRCLS(*USER) SPCAUT(*USRCLS)
```

Create two group profiles for the Admin and Performance teams.

```
CRTUSRPRF USRPRF(PERFTEAM)
CRTUSRPRF USRPRF(DBATEAM)
```

By default, we decided to grant QIBM_DB_SYSMON usage to all users.

```
CHGFCNUSG FCNID(QIBM_DB_SYSMON) DEFAULT(*ALLOWED)
```

Associate user's with group profiles.

-- Scott is on the performance team

```
CHGUSRPRF USRPRF(SCOTT) GRPPRF(PERFTEAM)
```

-- Mark works on both performance and DB admin

```
CHGUSRPRF USRPRF(MARKA) GRPPRF(PERFTEAM) SUPGRPPRF(DBATEAM)
```

-- The tester handles performance benchmarks

```
CHGUSRPRF USRPRF(TESTER1) GRPPRF(PERFTEAM)
```

-- Jim is the primary DB administrator

```
CHGUSRPRF USRPRF(JIMF) GRPPRF(DBATEAM)
```

Grant and revoke authority to the DB2 for i Administrator function usage.

```
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(PERFTEAM) USAGE(*ALLOWED)
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(DBATEAM) USAGE(*ALLOWED)
CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(TESTER1) USAGE(*DENIED)
```

At this point, the security officer is done. As people move on/off of the PERFTEAM or DBATEAM, all the security officer has to do is change their user profile. The function usage settings can remain intact.

Scott Forstie *is a senior software engineer at IBM, and he is the SQL development leader for DB2 for IBM i in Rochester, Minnesota. Before working on DB2, he worked on UNIX(R) enablement for the AS/400(R) and S/390(R) systems.*

Source URL: http://iprodeveloper.com/database/improved-security-controls-open-door-db2-i-tool-usage