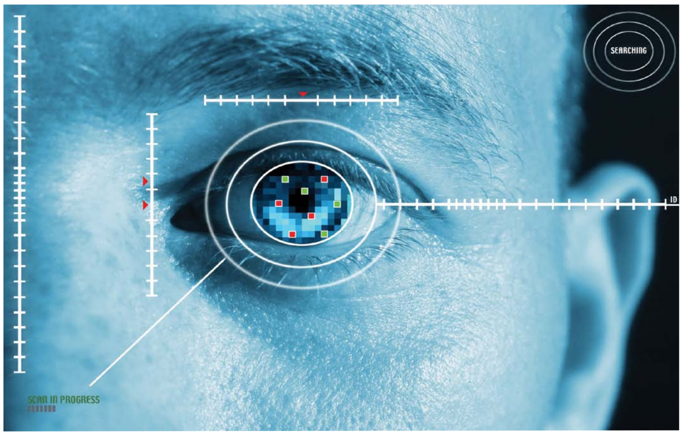
DB Security That You Have Always Dreamed About

Security is, or should be, a major consideration for any database administrator. Depending on the requirements of each company, this could be as simple as requiring a user name and password to connect to the system, or as complex as different permissions for different types of SQL operations on various objects in the database for each individual user.

Standard permissions have been available in BBj® since version 1.0. These permissions give the administrator the ability to assign read-only or read/write access to an entire database. However, standard permissions did not permit assigning different permissions to different objects (tables, views, etc) in the database – it is all or nothing. While this is very easy to manage, it limits the control that the administrator has over access to sensitive data. This in turn could limit the ability for users to have easy desktop data query access to subsets of the company production database. Workarounds are possible but they require far more maintenance and structuring of the database by the database administrator.



BBj 11.0 now provides a complete feature set for managing user permissions in a powerful new feature called "Object Level Permissions." This new feature will, for many customers, unlock meaningful desktop data query access to corporate data for all the users within a company with appropriate restrictions that are easy to structure and maintain.

Object Level Permissions

The reference to "objects" in a database refers to tables, views, and stored procedures. Object Level Permissions gives the administrator the power to assign different permissions to different users or groups of users on varying objects in the database.

For example, a role or group of users called HUMAN_RESOURCES contains a list of all users in the HR department. This group of users might have access to information such as salaries, performance reviews, etc., that the company does not want everyone to access. The HR-specific tables would only grant permission to this role and any other individual users who might need the information. Anyone not explicitly granted permission or made a part of this role would be unable to access the information in these tables. This puts the security at the database level, instead of the application level, making it impossible to circumvent by accessing the database outside the application.

Standard Permissions vs. Object Level Permissions

Here is a quick look at the two different ways of managing permissions:

	Standard/Legacy Permissions	Object Level Permissions
User permission types	Read or read/write permissions	SELECT, UPDATE, INSERT, DELETE in any combination
Table, view, stored procedure permissions	The same permission for all	Different permissions for each
Permission setting locations	From the BBj Enterprise Manager or the Admin API Admin	From the BBj Enterprise Manager, API or using standard SQL GRANT/REVOKE statements
User group options	Cannot group users with similar permissions	Can group users using ROLES

> >

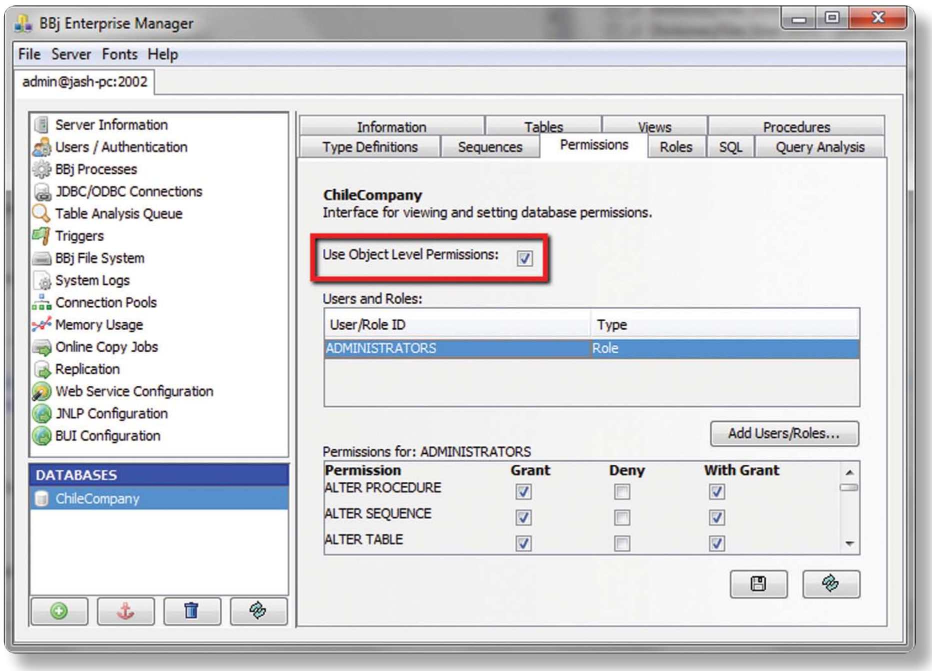
By Jeff Ash
Software Engineer

Using Object Level Permissions

The default setting for databases is Standard Permissions. With a click in a single checkbox, the administrator can switch to using Object Level Permissions or back to Standard at any time. To enable Object Level Permissions:

1. While logged into the Enterprise Manager as the "admin" user, click on the database to administer.
2. Select the Permissions tab.
3. Click in the checkbox labeled "Use Object Level Permissions" as shown in **Figure 1**.
4. Click the save button to update the database configuration.

After enabling Object Level Permissions, developers can assign permissions at the database level as well as at the object level. The Permissions tab shows a list of the users and roles who have permissions assigned at the database level. These permissions include things such as CREATE TABLE, ALTER TABLE, CREATE PROCEDURE, ALTER PROCEDURE, etc.



ChileCompany
Interface for viewing and setting database permissions.

Use Object Level Permissions: ✓

Users and Roles:

User/Role ID	Type
ADMINISTRATORS	Role

Permissions for: ADMINISTRATORS

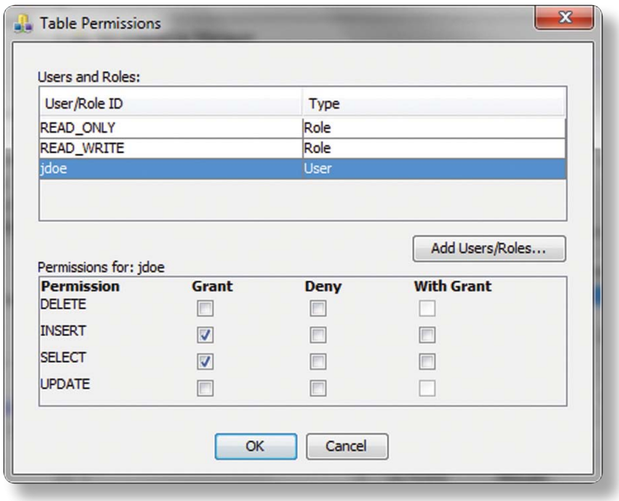
Permission	Grant	Deny	With Grant
ALTER PROCEDURE	✓		✓
ALTER SEQUENCE	✓		✓
ALTER TABLE	✓		✓

Figure 1. Database Permissions tab

Assigning Table, View, and SPROC Permissions

To assign table, view, and SPROC permissions, select the appropriate tab for the type of object to work with. Right-click on any table, view, or SPROC to show a popup menu with the "Permissions" option available. Then select multiple items at one time to assign permissions to all or some objects. **Figure 2** shows the permissions assignment dialog for tables.

Granting a permission gives the user the ability to perform that operation on the database. Adding the "With Grant" option gives the user the ability to grant that permission to other users on that object. Selecting "Deny" overrides any other permissions that may be set for the user if they should belong to a role that has that particular permission.



Users and Roles:

User/Role ID	Type
READ_ONLY	Role
READ_WRITE	Role
jdoe	User

Permissions for: jdoe

Permission	Grant	Deny	With Grant
DELETE			
INSERT	✓		
SELECT	✓		
UPDATE			

Figure 2. Table Permissions dialog

Using Roles

Roles are a powerful feature used to make the management of permissions much easier. A "role" is simply a list of users who will have the same permissions for certain objects in the database such as in the human resources example cited earlier in this article. Let's take a look at how to create roles and manage membership of those roles.

The Roles tab shows a list of roles currently defined for the database. These roles are specific to only the database for which they are defined. To view the members of a role, simply select the role from the list as shown in **Figure 3**.

After defining roles, developers can assign permissions to roles instead of individual users. This is the recommended way to assign permissions as it is highly unlikely that each user will need completely different permissions on database objects. The example in **Figure 3** shows a role for ADMINISTRATORS that would be used to assign database level permissions to various users who need to > >

perform administrative tasks such as CREATE and DROP TABLE. The READ_ONLY role will be used for users who need only READ_ONLY access, while READ_WRITE would be used for users who need to perform read/write operations from SQL.

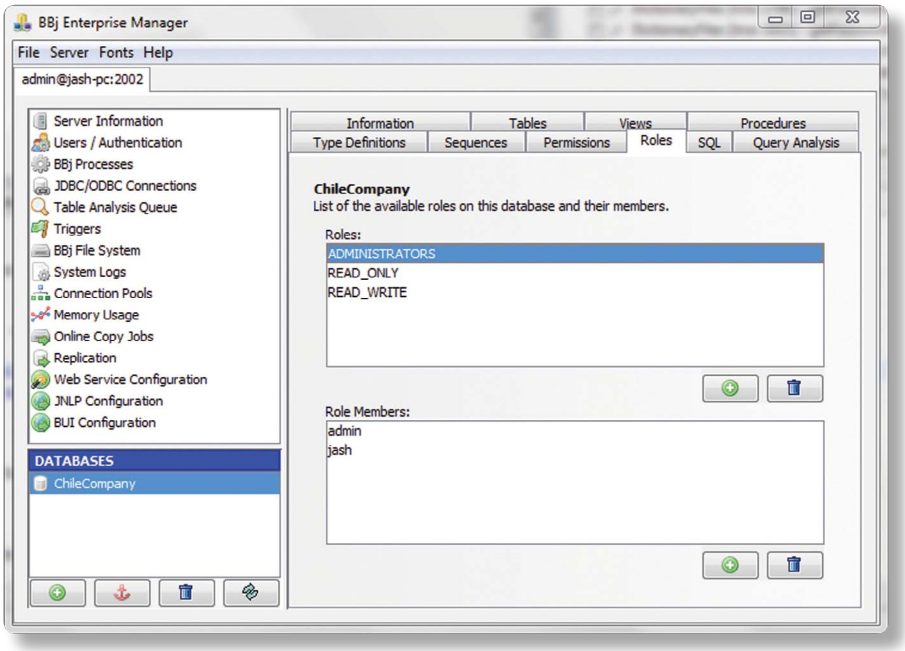
Should it become necessary to change the type of access a user needs, simply add them to or remove them from a role, and those permissions instantly change for that user on any objects that have permissions defined for that role.

Using Other Ways to Manage Permissions

There are three ways to manage permissions: 1) using the BBj Enterprise Manager, 2) programmatically using the Admin API, or 3) using SQL GRANT/REVOKE.

SQL Statements Using GRANT/REVOKE

Use GRANT, REVOKE, CREATE ROLE, and DROP ROLE to manage permissions with standard SQL statements. Here are a few examples of how you would use these statements:



ChileCompany
List of the available roles on this database and their members.

Roles:
ADMINISTRATORS
READ_ONLY
READ_WRITE

Role Members:
admin
jash

Figure 3. Database "Roles" management tab

Create a new role:

```
CREATE ROLE MY_NEW_ROLE
```

Add a user to a role:

```
GRANT MY_NEW_ROLE TO 'jash'
```

Add SELECT permission to a role on a table:

```
GRANT SELECT ON MY_TABLE TO MY_NEW_ROLE
```

Revoke UPDATE permission from two users on a table:

```
REVOKE UPDATE ON MY_TABLE FROM 'jash','jdoe'
```

Add UPDATE and INSERT permission to two users on a table and allow them to do the same for other users:

```
GRANT UPDATE,INSERT ON MY_TABLE TO 'jash','jdoe' WITH GRANT OPTION
```

BBj Admin API

Discussion of the use of the Admin API is beyond the scope of this article, however, for those interested, check out BBjAdminDatabase.grant(), BBjAdminDatabase.revoke(), BBjAdminDatabase.createRole(), and BBjAdminDatabase.dropRole() in the online BASIS Javadocs documentation for BBjAdminDatabase.

Summary

Security is a vital consideration in any organization and therefore, it is equally as important to BASIS who provides the framework and foundation for any application. BBj 11.0 provides another powerful security feature enabling administrators to better manage access to their precious data with Object Level Permissions. Using Object Level Permissions, administrators now have the power to control SQL access to their databases down to the individual table, view and stored procedure level with ease. Furthermore, the addition of roles allows administrators to group users together for even more flexibility, power, and efficiency. Allied with the ability to offload realtime queries to a replicated database (see Anatomy of a Replication Job on page 32), there is no longer any reason for database administrators to not open up access to live corporate data via a user's desktop data query tool. ■

See *Managing Database* and *Database Permissions* in the online documentation