**Grant Execute Permissions to SQL Server Stored Procedures using DDL Triggers**

|  |  |  |
| --- | --- | --- |
| http://www.mssqltips.com/images/Ranga_Narasimhan.JPG | By: [Ranga Narasimhan](http://www.mssqltips.com/sqlserverauthor/27/ranga-narasimhan/) | [Read Comments (4)](http://www.mssqltips.com/sqlservertip/1876/grant-execute-permissions-to-sql-server-stored-procedures-using-ddl-triggers/#comments) | [Print](http://www.mssqltips.com/tipprint.asp?tip=1876)   Ranga has been in IT for over 10 years working on Siebel, Visual Basic, Crystal Reports and SQL Server.  Related Tips: [More](http://www.mssqltips.com/sql_server_developer_tips.asp) | [http://www.mssqltips.com/images/MSSQLTipsTurnsSix_Giveaway.gif](http://www.mssqltips.com/giveaway.asp) |

**Problem**In your development environment if you have locked down permissions for developers, but still need them to execute stored procedures you will need to grant execute rights each time a new stored procedure is generated. In this tip I will show you a way of doing this automatically each time a new stored procedure is created without granting your developers additional permissions.

**Solution**The simple way that this can be accomplished is by using DDL triggers which are available in SQL Server 2005 and later versions.

Data Definition Language (DDL) triggers are fired when changes are made to database objects. For example, DDL triggers can be created to execute when a new table is created or a new stored procedure is created etc. For a whole list of events you can run this query.

|  |
| --- |
| select \* from sys.trigger\_event\_types |

**Create a DDL trigger on Create Procedure event**

In my environment there is a database role called DevUserRole. All developers are members of this role. This role is a member of db\_datareader and db\_datawriter database roles. I had executed a script to grant execute permissions on all the existing stored procedures, but I wanted a way to do this automatically for all new stored procedures.

Here is the script that I created to do this. This uses a DDL trigger that is fired for the "Create\_Procedure" event. In this script I will grant execute rights to the role DevUserRole whenever a new stored procedure is created. I am also limiting this to only do this when the object is part of the "dbo" schema.

|  |
| --- |
| /\*  To get a list of all available events:  select \* from sys.trigger\_event\_types  This proc will grant execute permissions for any new procedure that is created.  Change the Rolename in this script for your environment.  \*/  IF EXISTS (SELECT \* FROM sys.triggers WHERE parent\_class\_desc = 'DATABASE' AND name = N'DDLTRG\_StoredProcedureCreate')  DROP TRIGGER [DDLTRG\_StoredProcedureCreate] ON DATABASE  GO  CREATE TRIGGER DDLTRG\_StoredProcedureCreate  ON DATABASE  FOR Create\_Procedure  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Purpose: Grant execute permissions to DevUserRole for all new stored procedures created.  \*  \* MODIFICATIONS  \* 11-03-09 Ranga Narasimhan NEW  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  AS  DECLARE @data XML  DECLARE @objectname VARCHAR(255)  DECLARE @DatabaseName VARCHAR(255)  DECLARE @SchemaName VARCHAR(255)  DECLARE @strsql VARCHAR(500)  SET @data = EVENTDATA()  SET @objectname = @data.value('(/EVENT\_INSTANCE/ObjectName)[1]', 'varchar(256)')  SET @DatabaseName = @data.value('(/EVENT\_INSTANCE/DatabaseName)[1]', 'varchar(256)')  SET @SchemaName = @data.value('(/EVENT\_INSTANCE/SchemaName)[1]', 'varchar(256)')  IF @SchemaName = 'DBO' --Grant execute permissions for stored procedures owned by dbo schema.  BEGIN  SET @strsql = 'grant execute on '+@DatabaseName+'.'+@SchemaName +'.'+@objectname+' to DevUserRole'  EXECUTE (@strsql)  END  GO |

As you can see in the above code I am using EVENTDATA which returns a value of type xml. By default, the schema definition for all events is installed in the following directory: C:\Program Files\Microsoft SQL Server\100\Tools\Binn\schemas\sqlserver\2006\11\events\events.xsd.

Here is a list of the information that is available in the EVENTDATA() function for the Create\_Procedure event.

|  |
| --- |
| <xs:complexType name="EVENT\_INSTANCE\_CREATE\_PROCEDURE">  <xs:sequence>  <xs:element name="EventType" type="SSWNAMEType"/>  <xs:element name="PostTime" type="xs:string"/>  <xs:element name="SPID" type="xs:int"/>  <xs:element name="ServerName" type="PathType"/>  <xs:element name="LoginName" type="SSWNAMEType"/>  <xs:element name="UserName" type="SSWNAMEType"/>  <xs:element name="DatabaseName" type="SSWNAMEType" />  <xs:element name="SchemaName" type="SSWNAMEType" />  <xs:element name="ObjectName" type="SSWNAMEType" />  <xs:element name="ObjectType" type="SSWNAMEType" />  <xs:element name="TSQLCommand" type="EventTag\_TSQLCommand"/>  xs:sequence>  xs:complexType> |

As you can see this is a pretty easy process to put in place each time a new procedure is created.

**Next Steps**

* DDL Triggers are very powerful and can help the DBA automate manual processes and be used in database auditing
* Take the time to tweak this to meet your needs
* Read these related tips:
  + [Script to check SQL Server connection pooling](http://www.mssqltips.com/sqlservertip/1830/script-to-check-sql-server-connection-pooling/)
  + [Connecting to SQL Server with a Bad Logon Trigger](http://www.mssqltips.com/sqlservertip/1631/connecting-to-sql-server-with-a-bad-logon-trigger/)
  + [SQL Server 2005 SysAdmin Login Auditing](http://www.mssqltips.com/sqlservertip/1627/sql-server-2005-sysadmin-login-auditing/)
  + [Using the EventData() Function with DDL triggers in SQL Server 2005](http://www.mssqltips.com/sqlservertip/1571/using-the-eventdata-function-with-ddl-triggers-in-sql-server-2005/)
  + [Auditing DDL (Create, Alter, Drop) Commands in SQL Server 2005](http://www.mssqltips.com/sqlservertip/1006/auditing-ddl-create-alter-drop-commands-in-sql-server-2005/)