



DBA Utilities

Database Administration



Database Options

Database Administration



Database Options

- Database Options are used to view or modify options for the selected database
- Syntax

```
ALTER DATABASE database_name
SET
{
    <option_spec> [ ,...n ] [ WITH <termination> ]
}
```



Database Option - Restrict Access

- The Restrict Access Database option: Specifies which users may access the database.
- Possible values are:
 - Multiple (MULTI_USER)
 - The normal state for a production database, allows multiple users to access the database at once.
 - Single (SINGLE_USER)
 - Only one user is allowed to access the database at once
 - Used for maintenance actions
 - Restricted (RESTRICTED_USER)
 - Only members of the **db_owner**, **dbcreator**, or **sysadmin** roles can use the database.



Restrict Access - Example

- To take a database to a Single User Mode
 - Using management Studio

Database > Properties > Options > Restrict Access

- Using T-SQL

```
USE MASTER
```

```
GO
```

```
ALTER Database DBName
```

```
SET SINGLE_USER WITH ROLLBACK IMMEDIATE
```

```
ALTER Database DBName
```

```
SET MULTI_USER WITH NO_WAIT
```



Database Option - Auto Shrink

- The AUTO_SHRINK option shrinks files when more than 25 percent of the file contains unused space.
- It shrinks the file to one of two sizes (whichever is larger):
 - The size at which 25 percent of the file is unused space
 - The size of the file when it was created
- You can't shrink a read-only database.
- You can determine this option's status by
 - examining the is_auto_shrink_on column in the sys.databases catalog view
 - examining the IsAutoShrink property of the DATABASEPROPERTYEX function.



Auto Shrink – (cont.)

```
ALTER DATABASE          databaseName  
SET AUTO_SHRINK         { ON | OFF }
```

- ON: Specifies that the database files are candidates for periodic shrinking.
- OFF: The database files are not automatically shrunk during periodic checks for unused space.
- Both data files and log files can be automatically shrunk
- AUTO_SHRINK reduces the size of the transaction log only if you set the database to SIMPLE recovery model or if you back up the log.



Database Option - Database State

```
ALTER DATABASE          databaseName  
SET    {READ_ONLY | READ_WRITE }
```

- Changing the state of a database or file group to READ_ONLY or READ_WRITE requires exclusive access to the database
- To change the state of a database
 - ☐ Set the database to SINGLE_USER mode
 - ☐ Change the state of the database
 - ☐ Then return access to the database to all users



Database State - Example

```
USE master;  
GO  
ALTER DATABASE      databaseName  
SET      SINGLE_USER  
WITH     ROLLBACK IMMEDIATE;  
GO
```

```
ALTER DATABASE      databaseName  
SET READ_ONLY  
GO
```

```
ALTER DATABASE      databaseName  
SET MULTI_USER
```



The DATABASEPROPERTYEX Function



DATABASEPROPERTYEX

- The DATABASEPROPERTYEX function returns the current setting of the specified database option or property.
- Syntax
DATABASEPROPERTYEX (database , property)
 - Where Property is an expression specifying the name of the database property to return



Database Properties

- **Collation:** Default collation name for the database
- **IsAutoClose:** Database shuts down cleanly and frees resources after the last user exits.
- **IsAutoShrink:** Database files are candidates for automatic periodic shrinking.
- **Recovery:** Database recovery model
- **UserAccess:** Indicates which users can access the database
 - SINGLE_USER: Only one db_owner, dbcreator, or sysadmin user at a time
 - RESTRICTED_USER: Only members of db_owner, dbcreator, or sysadmin roles
 - MULTI_USER: All users



Database Properties (cont.)

■ Status: Database status.

- ONLINE: Database is available for query
- OFFLINE: Database was explicitly taken offline
- RESTORING: Database restore has started
- RECOVERING: Database recovery has started and the database is not yet ready for queries
- SUSPECT: Database did not recover
 - Also known as **Recovery Pending** databases
- EMERGENCY
 - Database is in an emergency, read-only state.
 - Access is restricted to sysadmin members



Examples

- Retrieving the status of the AUTO_SHRINK option

```
SELECT DATABASEPROPERTYEX('myDB', 'IsAutoShrink');
```

□ 0: indicates that AUTO_SHRINK is off

- Retrieving the default COLLATION for a database

```
SELECT DATABASEPROPERTYEX(myDB ', 'Collation') AS Collation
```



Using the DBCC Function



DBCC Function

- DBCC stands for Database Consistency Checker.
- DBCC commands are used to check the consistency of SQL server database.
- They are used for maintenance of database, tables, filegroups, indexes
- Help on DBCC commands
 - List the commands
DBCC HELP ('?');
 - Get a description of a particular command
DBCC HELP (CHECKDB);



DBCC Function - Examples

■ DBCC CLEAN TABLE

- Reclaims space from dropped variable-length columns in tables

```
DBCC CLEAN TABLE (myDB , 'myTable', 0)  
WITH NO_INFOMSGS;  
GO
```



DBCC Function - Examples

■ DBCC DBREINDEX

- Rebuilds one or more indexes for a table in the specified database

■ Rebuilding an index

- Rebuild the Employee_EmployeeID clustered index with a fill factor of 80 on the Employee table in the myDB database.

```
USE myDB;
```

```
GO
```

```
DBCC DBREINDEX ('Employee', PK_Employee_BusinessEntityID,80);
```

```
GO
```



DBCC Function - Examples

- Rebuilding all indexes

- Rebuild all indexes on the Employee table in myDB by using a fill factor value of 70.

```
USE myDB;  
GO  
DBCC DBREINDEX ('Employee', ' ', 70);  
GO
```



DBCC Function - Examples

■ DBCC SHRINKDATABASE

- Shrinks the size of the data and log files in the specifiedDB
- Shrinking a database and specifying a percentage of free space

■ Examples

- Reduce the size of the data and log files in the UserDB user database to allow for 10 percent free space in the database.

```
DBCC SHRINKDATABASE (UserDB, 10);  
GO
```

■ Truncating a database

- Shrink the data and log files in the myDB database to the last assigned extent.

```
DBCC SHRINKDATABASE (myDB, TRUNCATEONLY);
```



DBCC Function - Examples

■ DBCC SHRINKFILE

- Shrinks the current database's specified data or log file size
- Shrinking a data file to a specified target size

■ Example: Shrink the size of a data file named DataFile1 in the UserDB user database to 7 MB.

```
USE UserDB;  
GO  
DBCC SHRINKFILE (DataFile1, 7);  
GO
```



Maintenance Plan

Database Administration



Maintenance Plan

- Maintenance Plan is a defined workflow that runs a set of preconfigured maintenance tasks
 - such as backing up databases
 - Rebuilding indexes or
 - Running SQL Server Agent jobs.
- Administrators can run maintenance plans manually or schedule them to run at specified intervals
- A Maintenance Plan can be created using the Maintenance Plan Wizard in Management Studio



Maintenance Plan (cont.)

- Using the Management Studio

- Create a Maintenance Plan that backs up all user databases

1. Management > Maintenance Plan
2. New Maintenance Plan
3. Toolbox – Drag the “Backup Database Task”
4. DB Click on the “Backup Database Task” box and configure
5. General TAB



Maintenance Plan (cont.)

5. General TAB

- ☐ Select databases
- ☐ To Disk

6. Destination

- ☐ Set the destination
- ☐ Subfolders for each database

7. Options

- ☐ Encryption
- ☐ Expiry
- ☐ Verify backup integrity

8. Save ALL



Maintenance Plan (cont.)

- Once a Maintenance Plan is created, it will create an entry in SQL Server Agent
- The Maintenance Plan can be executed manually from
 - ☐ SQL Server Agent
 - ☐ Maintenance Plans
- The Maintenance Plan can also be scheduled to run automatically



Policy Based Management

Database Administration



Policy Based Management

- Modern relational systems are all about **proactive management** of your data
 - Proactively manage the objects and rules applied to the server
- You establish a set of rules to govern your server
- There is a wide list of items that can have policies attached to them



Policy Based Management (cont.)

- How exactly these Policies (rules) are enforced is set by rules — treatment options include:
 - On Demand: Only check for violations of policy when an administrator has specifically requested the policy audit.
 - Scheduled: Run an audit of policy violations according to some schedule
 - On Change: Prevent: This proactively prevents the policy violation from being allowed.
 - On Change – Log: This notes the violation, but simply logs it



Examples

- Want to enforce that all stored procedures begin with “sp”
- Steps to create the policy
 - Management > Policy Management
 - New CONDITION
 - Name: spNameCondition
 - Field: @name
 - Operator: =
 - Value: 'sp_%'
 - New Policy
 - Name: spNamePolicy
 - Check condition: spNameCondition
 - Against Target: Every Stored Precedure
 - Evaluate Mode: On demand



Jobs and Tasks

Database Administration



Tasks

- A task is a logical unit of work
 - single processes that are to be executed or
 - a batch of commands that are to be run
- In SQL Server, TASK cannot be created on their own
 - TASKs can only exist as members of JOBs.
- Example of tasks
 - backing up a database
 - running a T-SQL script to meet a specific need
 - Write the results of a JOB to a file (for auditing)
- Jobs and Tasks are created using SQL Server Agent service



Jobs

- A JOB is a grouping of one or more tasks
 - The TASKs in a JOB can be defined
 - Run sequentially or
 - Run conditionally
- Dependencies and branching can be defined among the TASKs
 - depending on the success or failure of individual tasks
- For example
 - Task A can be defined to run if the previous task succeeds
 - Task B can be defined to run if the previous task fails



Jobs

- Jobs can be scheduled based on the following criteria:
 - ☐ Daily, weekly, or monthly basis
 - ☐ A specific time of the day
 - ☐ A specific frequency (say, every 10 minutes, or every hour)
 - ☐ When the SQL Server Agent starts
 - ☐ In response to an alert



Jobs

- A job can contain several tasks
 - this is no guarantee that every task in a job will run.
- Tasks are run as part of a job and based on the branching rules you define for your job.
- In addition to branching you can perform the following:
 - Provide notification of the success or failure of a job to an operator.
 - Write the information to the event log.
- A TASK can be defined to either run or not run depending on the success or failure of other tasks in the job
 - A task can run in response to the success or failure of another task
 - For example, you might cancel the remainder of the job if one of the tasks fails.



Creating Jobs - Example

- SQL Server Agent > Jobs > New Job > Name the job
- Steps > New
 - ☐ Step Name , Type , Command
 - ☐ Repeat and add steps to the job
- Set On Success / On Failure options
- Setup schedule
- Add Alert, Notifications, etc.



Operator

- The **SQL Server Agent** service supports the notification of administrators through operators
 - Operators enable notification and monitoring capabilities of SQL Server Agent
- An Operator is an alias for people or groups that can receive electronic notification when jobs have completed or alerts have been raised



Scheduling

- Jobs (collection of tasks) can be scheduled
- Scheduling jobs allows you to run tasks that place a load on the system at off-peak hours
- It also ensures that you don't forget to take care of things
- Example: Index rebuilding and performing backups



Scheduling (cont.)

- Scheduling means defining the condition or conditions that cause the job to begin running without user interaction
- You can schedule a job to run automatically by
 - creating a new schedule for the job, or
 - attaching an **existing schedule** to the job
 - After a schedule has been created, you can attach that schedule to multiple jobs
- A schedule can be based upon time or an event
- A schedule defines the Recurrence and frequency of a job



Help on Jobs and Scheduling

- `dbo.sysjobactivity` - shows the current information of the jobs
- `dbo.sysjobhistory` - shows the execution result of the jobs.
- `dbo.sysjobs` - shows the information of the jobs
- `dbo.sysjobsschedules` - shows the job schedule information like the next run date and time of the jobs.
- `dbo.sysjobsteps` - shows the job steps
- `dbo.sysjobsteplogs` - let you see the logs of the steps configured to display the output in a table



Topics to Explore

- Database Tuning
 - Execution Plan
-