**SmartReindexing V3.0**  
Universal Re-indexing script supporting SQL 2005 onwards, ensures minimum locks, no production outage due to Blockings or disk out of space…   
[Full Video explaining script: (this is for V 1.0)](https://www.youtube.com/watch?v=rTJhJy51BIE&feature=youtu.be)

**Files:**

1. SmartReindexing.ps1 - Script to run from local or network computer
2. SmartReindexing\_TestJob.sql - Test job to schedule reindexing job

**License:** The SQL Server SmartReindexing Solution is free.

**Distinctive Features:**

* Re-index only fragmented tables
* Takes log backups in between to ensure log file do not get filled up
* Timeout automatically based on table size to avoid long blockings created by Index jobs
* Automatically update fill factor to 90 if not specified for tables with fragmentation
* Detailed execution status is logged into msdb..tbl\_indexRebuild\_Log and html log file

**Parameters:**

* **lServerName**: SQL Server Name
* **SqlUser**: SQL user name (in case domain authentication not preferred)
* **SqlPassword**: If password is provided SQL authentication will be used to perform re-indexing job
* **DoNotIgnoreReadOnlyDatabase**: default option will ignore read only db's for reindexing
* **FragCollTimeout**: default option will set 1200 second (20 min) max timeout for collecting fragmentation information for each db
* **BackupAfterChangePagesGB**: Log backup to be executed after GB worth of data is changed due to indexing, default is 1 GB
* **IgnoreTableForReindexing**: specify table names separated by comma if re-indexes should be ignored
* **LogBackupJobName**: Pass backup job name, default it will search for any of the above jobs to see if they exists:
  + DBA:Backup All Tlogs
  + DBA\_BackupDB.LogBackup
  + DBA\_BackupDB.Logsbackup
* **IgnoreTableForReindexing**: specify table names separated by comma if re-indexes should be ignored

**Examples:**

1. #To run on remote server using sql authentication

$erroractionpreference = "Stop"  
$ScriptDir = split-path -parent $MyInvocation.MyCommand.Path  
$lServerName="testserver.contoso.com"  
$SqlUser="validateSql"  
$SqlPassword="testpass"   
Z:\Scripts\SmartReindexing\SmartReindexing.ps1 -lServerName $lServerName -SqlUser $SqlUser -SqlPassword $SqlPassword

1. #To run on localhost and ignore read only db using windows authentication

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1

1. #To run on localhost and include read only db

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1 -DoNotIgnoreReadOnlyDatabase

1. #To execute using sql authentication

$erroractionpreference = "Stop"  
$SqlUser="validateSql"  
$SqlPassword="testpass"   
D:\DBABIN\SmartReindexing.ps1 -SqlUser $SqlUser -SqlPassword $SqlPassword

1. #To ignore certain tables from reindexing

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1 -IgnoreTableForReindexing "PerformanceSecurity,AdvPortfolioTransaction"

1. #To execute "DBA\_BackupDB.LogBackup" backup job

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1 -LogBackupJobName "DBA\_BackupDB.LogBackup"

1. #To execute with backup taken after 10 GB

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1 -BackupAfterChangePagesGB 10

1. #To execute with waiting upto 30 minute before ignore collecting fragmentation data (in seconds)

$erroractionpreference = "Stop"  
D:\DBABIN\SmartReindexing.ps1 -FragCollTimeout 1800

**Version 3 changelog:**

* Custom timeout for collecting fragmentation information
* Custom variable for specifying log backup job
* Wait for Log backup to finish
* Allow excluding tables from re-indexing to handle space issues
* Custom variable for specifying log backup job trigger after certain GB of data change
* Error handling to report all errors as summary in the end of html log file