Sunny Solutions

MS Access Repair & Compact: User Guide

(v1.084, 26-Jul-2022, 03:00 AM)

Disclaimer: ***The author of this product accepts no responsibility for damages resulting from the use of this product and makes no guarantees, warranty, or representation, either express or implied, including but not limited to, any implied warranty of merchantability or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assumes all risks when using it.***

1 Introduction

MS Access DB's tend to get corrupted by either user error or through other unforeseen circumstances like unreliable or intermittent network connections. Regular database backup is essential and occasionally performing a **Repair & Compact** (**RC**) is also a good practice. MS Access does provide a **Compact & Repair Database** option in its user interface, but here are some issues administrators and users encounter, while using **MS Access**' RC function:

- Quite cumbersome to get to with passwords and other DB protective measures.
- If the DB is distributed as an **MS Access Application**, then the **Repair & Compact Database** menu item may not be available to end-users.
- It is a laborious process to RC, if the **Shift** key is disabled and a DB and/or user passwords are assigned.
- Frequently, this function fails, leaving behind the dreaded **db1.mdb** file!
- Users are required to be cognizant of exiting the DB, when they are done with an application within it. Application.Quit should be called when no user input is detected for an extended period of time.
- Leaving a DB open overnight is a sure way to corrupt it when backups and other security policies are in place!

Note: ***There are several very powerful MS Access DB repairing applications available on the market. These have a very specific purpose and that is to restore a DB that will not respond to the MS Access Compact & Repair Database command. MSARC does not replace these sophisticated applications! It is a simple utility that attempts to keep an Access DB in optimum shape on a daily basis, if used appropriately, hence not requiring those expensive applications!***

2 Key Features

So, having an external application that can perform the RC operation is of considerable advantage to the DB administrator. This is where **MS Access Repair & Compact** (**MSARC**) utility comes into play. Here are some of its key features:

- Once set up, it does not require having to open the Access DB.
- Provides a simple user interface to RC the DB.
- Provides a console application that can be run each night through the **Windows Task Scheduler** or some other mechanism without user intervention.
- Provides backup of original and RC'd files.
- Provides zipping after the RC.
- Allows running under Cybele Software's VirtualUI Server so it can be run through a HTML5 -compatible browser, essentially enabling it in a cloud server-type environment.

Note: ***The author has taken every precaution to ensure that the data is sufficiently backed up before and after the Repair/Compact process. The code called within the application is a single call to Microsoft JRO's (Jet Replication & Objects) CompactDatabase() method, so the responsibility is on them for the actual Repair/Compact of the DB. If you see problems with a db1.mdb file lying around, then it is a Microsoft Access issue, not with this application! If you do find any problems, please do not hesitate to contact the author at your earliest convenience!***

3 MS Access Repair & Compact

MSARC consists of two applications. One is a Windows graphical user interface (GUI) application that allows a user to manually repair/compact (RC) the DB, and the other is a Windows Console application that runs automatically when launched with the provided command line. It can also be run unattended nightly using a scheduler such as the **Windows Task Scheduler**.

3.1 Source File

MSARC will RC any MS Access DB files of versions: **95/97**, **2000**, **2002/2003**, **2007, 2010, 2013, 2016**. The file extensions in the table below determine which DB engine is used to perform the RC:

File Extension	Resulting RC Operation
***.accdb**	The **Microsoft Office 15.0 Access Database Engine Object Library**, using **Data Access Objects (DAO)** from the installed copy of Access 2007 is used to perform the RC. The Access file format contained in the file itself may be of any version; 95/97, 2000, 2002/2003, 2007, 2010, 2013, 2016. The DAO engine determines if there are any issues. ***Note: Use this extension for all Access 2007 and higher files formats.***
***.mdb**	The **Microsoft Jet and Replication & Objects 2.6 Library (JRO)** is used to perform the RC. Only Access file formats of 95/97, 2000, and 2002/2003 are supported. The Access 2007 file format is not supported with this extension. MSARC automatically determines what the MDB's file version is and uses the appropriate parameters to perform the RC. If the file is too corrupted to get this information, MSARC will not continue. ***Note: Use this extension for all version prior to Access 2007 file formats.***

3.2 Backup Files

Basically, **MSARC** uses a MS Access DB source file and a backup folder. When you start the process, the original source file is copied to the backup folder with **_YYYMMDD####** appended to the filename. This is done, so that if problems are encountered with RC, the file can easily be restored. The DB is then RC'd and this file is also backed up with an extension of **sss**, unless **Create backup file after RC** option is disabled. So, for each run either one or two backup files are created. E.g., if the DB is named **MyFile02.mdb**, then the backup files for **15-Jan-2008** would be **MyFile02_200801150001.mdb** and **MyFile02_200801150001.sss**.

The **####** is a four-digit counter that increments for each backup file from **0001** to **9999**.

In addition to this, the **Create ZIP file after RC** option creates a zip of the **mdb** file after RC. During the development cycle of an MS Access application, copying before and zipping after to a backup folder is recommended. In this case, turning **OFF** the **Create backup file after RC** option and turning **ON** the **Create ZIP file after RC** option creates pairs of files that are easy to keep track of. E.g., **MyFile02_200801150001.mdb** can be very large after a long day of multi-user activity, and **MyFile02_200801150001.zip** ends up being much smaller after RC and ZIP, and therefore, appropriate for backing up.

So, for any given run of the application, if both options are checked, then the resulting files would be:

Source File:	**MyFile02.mdb**
Original Backup:	**MyFile02_200801150001.mdb**
RC'd Backup:	**MyFile02_200801150001.sss**
Zip File:	**MyFile02_200801150001.zip**

The option **Prefix backup date** prefixes the date and counter information onto the filename. This is very handy to keep 2 or more DBs backed up and in sequential order. If the data/counter were prefixed, then the files would not be easily locatable in a directory. E.g.

Source Files:	**MyFile02.accdb, MyFile02_BE.accdb**
Frontend Backup 1:	**MyFile02_201912150001.accdb**
RC'd Frontend Backup 1:	**MyFile02_201912150001.sss**
Zip File Frontend 1:	**MyFile02_201912150001.zip**
Frontend Backup 2:	**MyFile02_201912150002.accdb**
RC'd Frontend Backup 2:	**MyFile02_201912150002.sss**
Zip File Frontend 2:	**MyFile02_201912150002.zip**
Backend Backup 1:	**MyFile02_BE_201912150001.accdb**
RC'd Backend Backup 1:	**MyFile02_BE_201912150001.sss**
Zip File Backend 1:	**MyFile02_BE_201912150001.zip**
Backend Backup 2:	**MyFile02_BE_201912150002.accdb**
RC'd Backend Backup 2:	**MyFile02_BE_201912150002.sss**
Zip File Backend 2:	**MyFile02_BE_201912150002.zip**

As you can see in the directory listing above, the files are ordered by name and backups from the same date/run are separated; imagine hundreds of days of backups. With a large number of backups, these files would be hard to locate. Enabling the prefixing of the backup date produces a more usable result:

Frontend Backup 1:	**201912150001_MyFile02.accdb**
RC'd Frontend Backup 1:	**201912150001_MyFile02.sss**
Zip File Frontend 1:	**201912150001_MyFile02.zip**
Backend Backup 1:	**201912150001_MyFile02_BE.accdb**
RC'd Backend Backup 1:	**201912150001_MyFile02_BE.sss**
Zip File Backend 1:	**201912150001_MyFile02_BE.zip**
Frontend Backup 2:	**201912150002_MyFile02.accdb**
RC'd Frontend Backup 2:	**201912150002_MyFile02.sss**
Zip File Frontend 2:	**201912150002_MyFile02.zip**
Backend Backup 2:	**201912150002_MyFile02_BE.accdb**
RC'd Backend Backup 2:	**201912150002_MyFile02_BE.sss**
Zip File Backend 2:	**201912150002_MyFile02_BE.zip**

Here you can see that the files for a given date/run are together and so easier to locate and work with.

From testing, the best settings would be as follows.
Enable: **Try to delete lock file** and **Create ZIP file after RC**.
Enable: **Prefix backup date**, if you have more than DB (FE/BE) being backed up in the same backup folder.

3.3 Log Files

Two log files are provided, where the results of errors are recorded. They are called **MsAccessRepairCompact.log** and **MsAccessRepairCompactConsole.log** and are located in the ProgramData folder:
C:\ProgramData\Sunny Solutions\MS Access Repair & Compact.

Note: These log files are not removed on program uninstall.

3.4 INI Files

INI files are used to store the RC settings which appear in the GUI. Multiple INI files are supported, so any number of Access DBs can be RC'd without much effort. Use the following command line to specify the INI filename in shortcut(s):

MsAccessRepairCompact.exe /ini="My Ini Filename.ini"
MsAccessRepairCompactConsole.exe /ini="My Ini Filename.ini"

All INI files ***must*** be located in the ProgramData directory where **MSARC** looks for them:

C:\ProgramData\Sunny Solutions\MS Access Repair & Compact

If no INI filename is provided using the command line, the default of **MsAccessRepairCompact.ini** is used. If this doesn't exist, then the GUI application will inform you, or the console application will fail.

Note: These INI files are not removed on program uninstall!

3.5 Alternate Filename Prefix

If a single DB file is used whose name changes for each revision (e.g., **MyFile01.mdb**, **MyFile02.mdb**, **Myfile03.mdb**, etc.), then **MSARC** can use the **Alternate Filename Prefix** to locate the next file in order in the same folder on a subsequent run. This feature was added to facilitate the backup of a file, if the administrator forgot to update the INI file after deploying a new version of a DB.

E.g., if **MyFile02.mdb** is currently in use and the next version is **MyFile03.mdb**, and **MyFile02.mdb** is not present in the folder, then **MSARC** will locate and use **MyFile03.mdb** on the next run, when you enter **MyFile** as the prefix. Leave blank to ignore this feature.

Note: **MSARC *will only get the name of first file it finds alphabetically in the source file's folder, so it is best to only have a single DB file in the application folder starting with the provided prefix, if you want to use this feature!***

3.6 DB Passwords

MSARC supports the presence of DB security. There can be three ***different*** types of security in an MS Access DB, and are described below:

Database:	The **DB Password** protects the entire DB. There is no username associated with it. If set in the DB, all users need to use this password to access it and will also be required by **MSARC**. Usually used in the Backend data DB in a split DB implementation. ***Note: It can be present in conjunction with* Workgroup Security *in non-Access 2007 versions.***
Workgroup Security:	When **Workgroup Security** is being used, each user usually has their own user ID and password assigned to them. The **Workgroup Information File** (***.mdw**) needs to be present and accessible by **MSARC**. ***Note: It can be present in conjunction with the* DB Password*. Ignored for Access 2007 DBs.***
Visual Basic for Applications:	The VBA password protects the code modules and copying of other DB objects. ***This is NOT used by* MSARC.**

Note: Although, both passwords and the workgroup usernames are encrypted in the INI files, no level of security is implied by the author!

Note: Passwords are per machine, so if you move an INI file to a different machine, you will need to enter the password and save the INI again, otherwise the decrypted password will be of no value when the INI file is read.

3.7 Workgroup Security

MSARC supports the presence of **MS Access Workgroup Security**. If you are using workgroup security in your DB, check **Use Workgroup Security**, and enter a **Username**, a **User Password**, and the location of the **Workgroup Information File**. This password is ***not*** the DB password, but the one associated with the workgroup user account. If the DB also has a DB Password, then provide this as explained in ***Section 3.6 DB Password*** as well.

Note: The workgroup user must have Database Administer rights, not just on other DB objects.

3.8 Presence of LDB/LACCDB File

Before running its process, **MSARC** checks for the presence of the DB's lock file (***.LDB** for **Access 95/97**, **2000**, **2002/2003**, and ***.LACCDB** for **Access 2007 and higher**) in the same folder. Originally, if the file existed, the entire process was aborted, the user was informed, and a log file entry made. This applied to both the GUI and console applications.

In testing, it was discovered that some third-party applications that launch an **Access** DB, may leave the lock file behind, even though **Access** has exited gracefully. Option **Try to delete lock file** has been added to assist in this scenario, whereby an attempt is made to delete it. If the lock file is not locked (all users have exited gracefully), attempting a delete is appropriate to continue with RC. If the file is indeed locked, meaning the DB is still open by one or more users on the server/network, and the delete fails, the process aborts as described above. But, if it can be deleted, RC continues as normal.

3.9 Running the Applications

Instructions on running the applications follow.

3.9.1 MSARC Application

This is the Windows GUI application which allows manually running the RC process.

1. Launch the application from **Start Menu | All Programs | Sunny Solutions | MS Access Repair & Compact | MSARC Application**. The main application window will appear (***Figure 1***).

Note: ***The first time RC is run and no settings have been assigned values, the message in Figure 2 appears. Once valid targets have been selected, the errors will not appear, unless the targets are invalid in the future, like a changed backup folder.***

Note: ***The first time RC is run and Access 2007 is not installed, the message in Figure 3 will appear. The CheckAccess2007 setting is then set to OFF in the INI file, and the check is not made again. Each new INI file will reset this setting.***

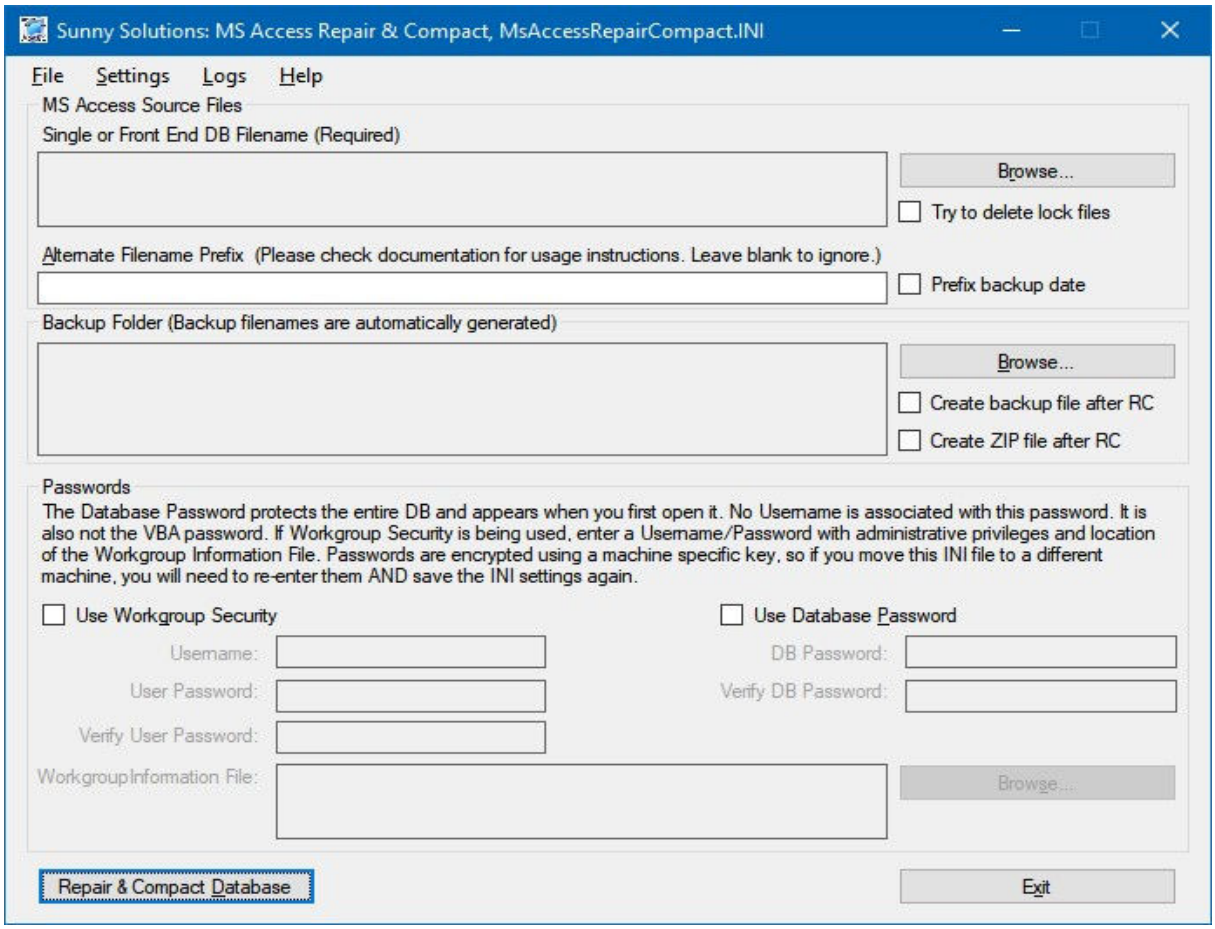

Figure 1: MSARC application window.

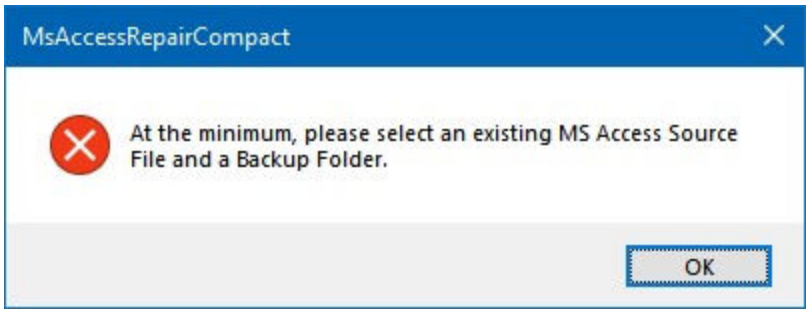

Figure 2: Error when no file or folder has been selected.

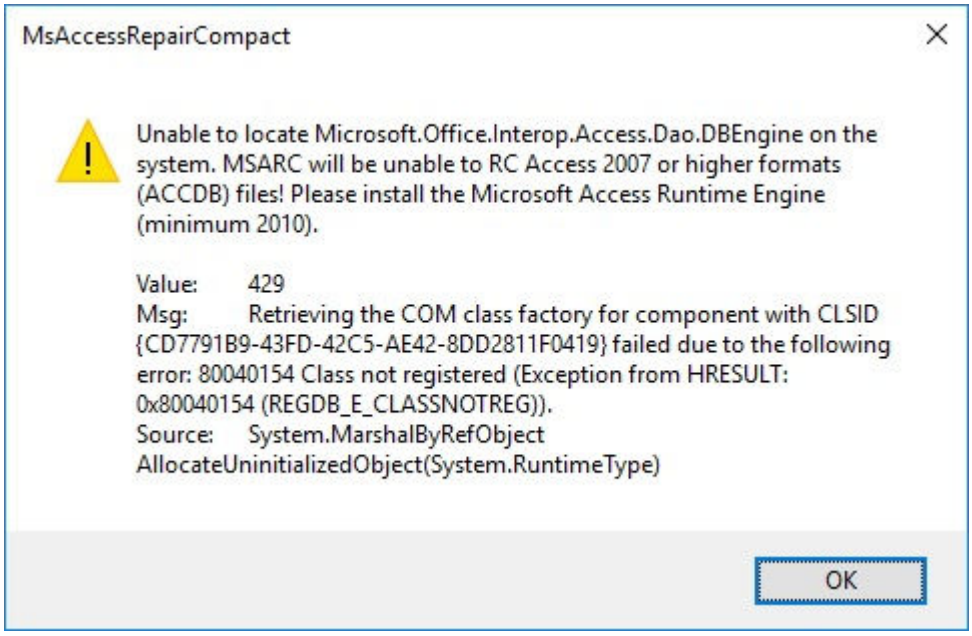

Figure 3: Error 429 when Access 2007 or higher is not installed.

2. For **MS Access Source File**, click **Browse...** on the right. You will be provided with an **Open** file window (***Figure 3***):

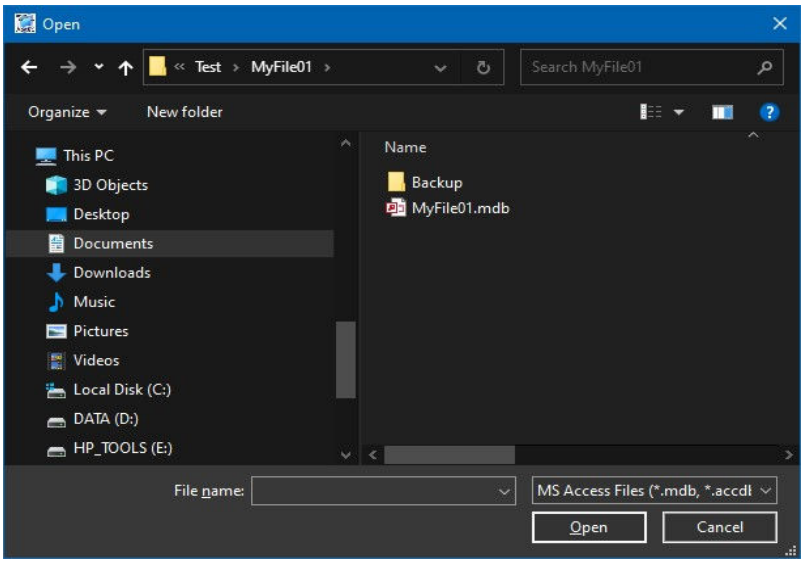

Figure 4: Open File window.

3. Locate your **MS Access DB** and click **Open**.
4. For **Alternate Filename Prefix**, enter the beginning of your DB's filename, so that if the original one does not exist anymore, it may try to locate a similar filename. See ***Section 3.5 Alternate Filename Prefix***.
5. For **Backup Folder**, click **Browse...** on the right. You will be provided with a **Browse for Folder** window (***Figure 4***).

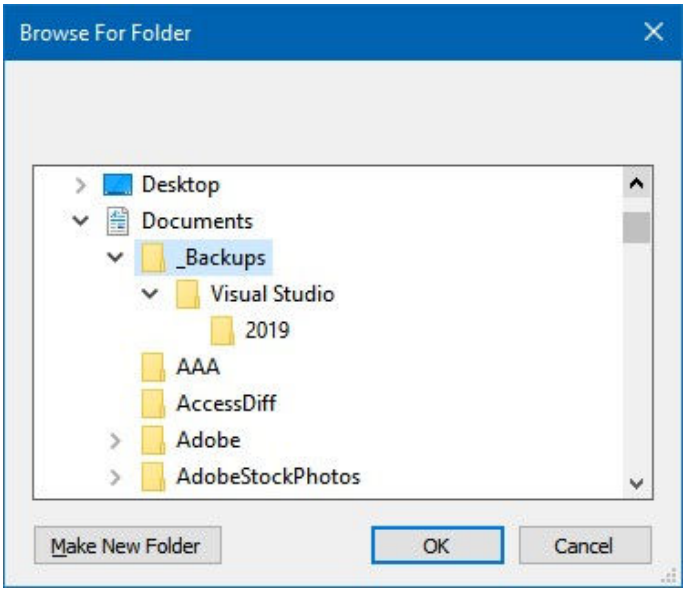

Figure 5: Browse For Folder window.

6. Locate the destination folder you would like to back your files up to and click **OK**.
7. Turn on/off the option to **Create backup file after RC**.
8. Turn on/off the option to **Create ZIP file after RC**.
9. If you are using workgroup security (non-Access 2007 DBs), then check **Use Workgroup Security**, and enter a **Username**, a **User Password**, and **Workgroup Information File** (***.mdw**) using **Browse** to the right. Enter the password twice for verification. See ***Section 0 :*** Passwords are *per machine*, so if you move an INI file to a different machine, you will need to enter the password and save the INI again, otherwise the decrypted password will be of no value when the INI file is read.
10. Workgroup Security.
11. If you are using a DB password, then check **Use Database Password** and enter the password twice for verification. See ***Section 3.6 DB Passwords***.
12. Select **Settings | Save INI File** to save the current selections to an INI file.
13. Click **Repair & Compact Database** to run the process. A message is provided at the end of the process indicating the names of the backup files.

3.9.1.1 File Menu

3.9.1.1.1 Repair Compact Database

Runs the RC process. Same as the **Repair & Compact Database** button.

3.9.1.1.2 Exit

Exits **MSARC**. Prompts the user to save changes, if changes have been made to the INI settings.

3.9.1.2 Settings Menu

3.9.1.2.1 Clear All

Clears all the settings on the screen to the defaults.

3.9.1.2.2 Open INI

Opens an INI file from the application folder.

Note: Please use the ProgramData folder only. INI files in other folders are not supported!

3.9.1.2.3 Save INI

Saves the onscreen settings to an INI file.

Note: Please use the ProgramData folder only. INI files in other folders are not supported!

3.9.1.2.4 Save INI As

Saves the onscreen settings to a user selectable INI file.

Note: Please use the ProgramData folder only. INI files in other folders are not supported!

3.9.1.2.5 Create Desktop Shortcuts

Use this to create desktop shortcuts, so they are easy to execute, or setup keyboard shortcuts for.

3.9.1.2.6 Copy Console Command Line to Clipboard

Use this to copy the command line necessary to run the console application with the currently selected INI file. Usually used for creating a task in Windows Task Scheduler.

3.9.1.2.7 Create Task Scheduler Import File

Use this to create an XML file that you can import into Windows Task Scheduler easily.

3.9.1.3 Logs

3.9.1.3.1 View Application Log

Open the **MSARC** GUI application's log file in your default text editor (e.g., **Notepad**).

3.9.1.3.2 View Console Log

Opens the **MSARC** Console application's log file in your default text editor (e.g., **Notepad**).

3.9.1.4 Help

3.9.1.4.1 User Guide

Opens this file in **Adobe Acrobat**.

3.9.1.4.2 Release Notes

Opens the latest **Release Notes** in **Adobe Acrobat**.

3.9.1.4.3 Check for Access 2007

Provides a manual check to ensure that **Access 2007 or higher** is correctly installed and that MSARC has access to the **Microsoft Access 15.0 Access Database Engine Object Library** (Microsoft.Office.Interop.Access.Dao.dll).

3.9.1.4.4 About

Provides version information for this application.

3.9.2 MSARC Console Application

This is a Windows console application that allows you to automatically run the process without human intervention. You can provide the INI filename in the command line as specified in ***Section 3.4 INI Files***, so you can RC multiple Access DBs without much effort. There is no user interface to this application, so errors and results are only written to the console application log file. The application can be run in two ways:

1. Launch the application from **Start Menu | All Programs | Sunny Solutions | MS Access Repair & Compact | MSARC Console Application**. There will be no user interface associated with it!
2. Set the application up in the **Windows Task Scheduler**. Recommendations:
   - The scheduled task is setup on the same computer as the hard drive that the DB is located on for file integrity and performance.
   - The username/password provided for the task is an administrative level user on the same computer.
   - Ensure that users always close the DB file before the task runs [each night] as the presence of an LDB/LACCDB file will cause the process to abort completely. See ***Section 3.8 of LDB/LACCDB File***.

If you want to view the console application's log file, you can view it from the GUI application, or you can launch it from **Start Menu | All Programs | Sunny Solutions | MS Access Repair & Compact | MSARC Console Log**.

4 Errors

Some specific errors are provided below. All other error descriptions are provided by the applications and in the error logs.

Error #	Description
5	***Access Denied:*** This error usually occurs if the DB file is read-only. Only RC files that have read/write access to the user the applications are running under.
429	***COM class factory error:*** This error usually occurs, if Access 2007 is not installed. If you are trying to RC an Access 2007 format DB, then you need to install Office/Access 2007 or the Access 2010 (or higher) Runtime Engine. You will get this error if you try to run RC on the DB as well. Please follow instructions in ***Section 5 Installation***.

5 Installation

Please read this section entirely before installation!

5.1 Install

Installation of **MSARC** used to be quite simple when last released in 2010, but due to the following, things have actually gotten more complicated:

- **.NET Framework** dependencies.
- Which version of**.NET Framework** pre-exists on ***different*** versions of **Windows**.
- Which installer is used to deliver **Office** products.
- Different versions of **Office**/**Access** are incompatible and not recommended to be installed at the same time.
- **Windows** increased default security.

Originally, you were provided a ZIP file (e.g. **MsAccessRepairCompact_2_03_000.zip**), which you extracted to a folder on your hard drive and launched **Setup.exe**. You only needed to click **Next** on each step and then **Close** at the end. This still stands, but a few more things need to be taken into consideration and the rest of this section provides some guidelines to success.

Note: ***A newer version will force you to uninstall the older version before allowing the process to continue.***

Note: ***The setup now installs .NET Framework v4.8 and so requires an internet connection (especially on a new Windows Server 2016 installation. If it is already installed, it will still go through a download, but actually skips it in the background. This is a Microsoft anomaly.***

MSARC uses **.NET Framework**'s **Microsoft.Office.Interop.DLL**, which contains **Microsoft.Office.Interop.Access.Dao**. This component is not installed (or installed correctly) by **MS Office 365 (2016) Click-to-Run** installations, and **MSARC** flags **Error 429** and cannot proceed. Therefore, this and other components need to be installed using a different installer from MS.

If you have **Microsoft Office 365** installed, make sure you only have 32-bit versions of it. Do not install any 64-bit components. If you get errors telling you that an installation cannot go further due to a 64-bit installation being present, then you need to follow the instructions in this link to remove them first as they don't necessarily appear in **Control Panel | Add/Remove Programs** or **Settings | Apps & Features**.

https://answers.microsoft.com/en-us/msoffice/forum/msoffice_install-mso_win10/office-16-click-to-run-extensibility-component-64/e79ee5bd-f119-4808-9bb2-289dd815b76a

Note: ***On a Windows 10 Pro machine, installation errors were encountered indicating 64-bit components present on an Office 365 (32-bit) install. Originally, Office 365 (64-bit) was installed briefly along with Office Updates. This was then uninstalled to accommodate third-party Access application requirements and Office 365 (32-bit) was installed. The 64-bit uninstall did not remove components added during the updates, which also did not appear in the installed software list. The above procedure allowed their manual removal.***

Even though you have the latest **Office 365 (32-bit)** installed, you will still need to install **MS Access 2016 Runtime** due to the missing components. Unfortunately, **Office 365** uses the newer **Click-to-Run** installer and will not allow you to install this runtime, which uses the traditional **Windows Installer (MSI)** and they are NOT compatible; thank you, Microsoft! In this case, you need to install the **MS Access Database Engine 2016 Redistributable** in this link. This is smaller at 25MB, does not appear to have too many checks for what's already present or the type of installations, and has the components **MSARC** requires.

https://www.microsoft.com/en-us/download/details.aspx?id=54920

If you have an earlier version of **Access** like 2010, then you can download and install the 32-bit (not 64-bit) version of **MS Access 2010 Runtime** from this link. These are the earlier **Windows Installer (MSI)** packages and are compatible with **Office 2010**'s installations, so should not pose a problem.

https://www.microsoft.com/en-us/download/details.aspx?id=10910

Otherwise, download and install the **Access Database Engine 2010 Redistributable** to work around this:

https://www.microsoft.com/en-us/download/details.aspx?id=13255

If you **DO NOT** have any MS Office products installed on your system, then you can download and install the 32-bit (not 64-bit) version of **MS Access 2016 Runtime** (256MB) from the link below. It allows you to open DBs as well:

https://www.microsoft.com/en-us/download/details.aspx?id=50040

5.2 Uninstall

Due to the inconsistencies with **.NET Framework** installations on different versions of **Windows**, the uninstall may remove **.NET Framework v4.8**. If this happens, then please download and install it again from:

https://dotnet.microsoft.com/download

Note: On uninstall, the INI and log files are NOT removed.

6 Cybele Software VirtualUI Server Integration

Cybele Software's VirtualUI Server (**VUIS**) support was integrated in v2.4. This allows MSARC and MSARC Console to be run inside an HTML5-compatible browser page, effectively allowing MSARC to be run in a cloud server environment. This means not having to RDC into the server and perform further actions. A huge time saver. You can also launch MSARC from other applications running through **VUIS**.

6.1 Setup

Setting it up requires one or two entries under the **VUIS | Applications** tab, depending on which apps(s) are required: using **MSARC** and/or **MSARC Console**. End users may need access to both apps and each requires it's own entry based on VIUS' requirements.

Currently, **VUIS** does not have a way to import or export these settings, so they have to be entered manually. Use the **Duplicate** button on the **Application** tab to make it quicker.

The command line settings are similar to a Desktop shortcut or Task Scheduler entry, except for the addition of the **/UseVirtualUI=1** Command Line Parameter (**VUIS' Argument** field). This tells **MSARC** to connect to the web server on launch.

As long as you can get to **VUIS'** landing page, then you should be able to run either **MSARC** or **MSARC Console** with the setup shown below. Use the same **Credentials** as other applications setup in **VUIS**.

Note: ***Please make sure to add the trailing "/" in* Start in the following folder *field as in:***

C:\Program Files (x86)\Sunny Solutions\MS Access Repair & Compact

Note: ***Please make sure to add the "/" at the end of the Virtual Path in the URL, otherwise you will get a resource not found error in the browser. E.g.,***

https://myserver.com/Msarc_App_PTS_7/

Figure 6 shows the correct settings to allow launching of **MSARC** in a browser window through **VUIS**. **MSARC Console** can similarly be set up by changing **Program path and file name** to **MsAccessRepairCompactConsole.exe** instead of **MsAccessRepairCompact.exe**.

The **Name** and **Virtual Path** also need to be changed to reflect the function. E.g., **Msarc_Con_PTS_7**.

If there is a split DB with a backend DB, then one or two more **Application** entries may be required. E.g., **Msarc_App_PTS_7_BE** and/or **Msarc_Con_PTS_7_BE**.

These virtual paths can either be launched through a browser's URL/Address field as in the example above or alternatively, it can be launched from an already running VirtualUI-enabled app. E.g., a frontend DB can have a maintenance option to launch **MSARC** to perform a backup before a major data change operation.

Please refer to the **VUIS** documentation for further clarification of **VUIS'** features and their setup.

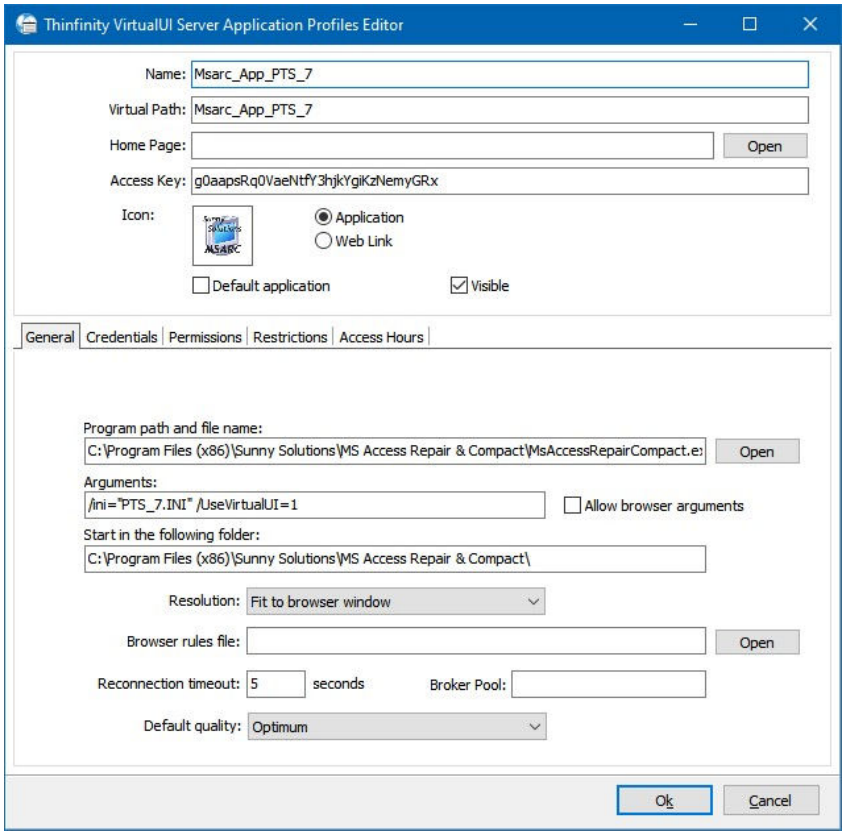

Figure 6: VirtualUI Server's Application Setup Window

7 Technical Support

The best way to get technical support for **MSARC** is to contact **Sunny Solutions**:

Email: jamshes@gmail.com
Website: http://SunnySolutions.Biz
Skype: SunnyJamshedji