

4DM Personal Database Verification

OVERVIEW

Local databases on a server or workstation can be verified to ensure they are functioning optimally and there are no database issues. If issues are found during the verification process, database files can be repaired and compacted to reclaim unused space within the database. The verification tool can be scheduled to run in the background through Microsoft Task Scheduler on the workstation.

Database Verification is a console application that is handled by the executable CBDatabaseVerfication.exe and is designed to be run through the command-line interface. The results from the database verification process are recorded in the user-specified text file that is designated while scheduling the verification task through Microsoft Task Scheduler.

HOW TO GUIDE

DATABASE VERIFICATION

Database Verification is most easily performed by scheduling it to run in the background through Microsoft Task Scheduler. The table below displays the required and optional command-line arguments to be designated while scheduling through Microsoft Task Scheduler.

Argument	Required	Description
-db	Yes	Specifies the database ID of the database to verify. Users that wish to apply different arguments to multiple, but not all, databases should create individual tasks for each database. <i>This argument requires a database ID descriptive parameter after it with a space separating the two (e.g., -db default). If there are any spaces in the name of the database, the entire name must be surrounded by quotation marks (e.g., -db "default database").</i>
-log	Yes	Specifies the file where the results should be written out. If the file already exists it will be overwritten. <i>This argument requires a file location descriptive parameter after it with a space separating the two (e.g., -log C:\results.txt). If there are any spaces in the file path, the entire path must be surrounded by quotation marks (e.g., -log "C:\Documents and Settings\results.txt").</i>
-fix	No	Indicates that fixable issues discovered during verification should be fixed. <i>IMPORTANT: This command allows the databases to be modified by adding, and/or deleting rows.</i>
-compact	No	Reclaims unused space from the databases.

Scheduling through Microsoft Task Scheduler:

1. Navigate to Administrative Tools and be aware that Administrative Privileges will be required:
 - a. For Server 2012, 2016, 2019 and Windows 10 & up (see 1 Figure 1):
 - Select the **Start** button > Type **Windows Administrative Tools**
 - Choose **Run as Administrator**.
2. In the *Administrative Tools* window, double left-click **Task Scheduler**.
3. Within the *Task Scheduler* window, click the **Action** tab and select **Create Task...** from the dropdown (Figure 2).
4. Enter a name for this task in the **Name** text box (e.g., Database Verification and a description for this task in the **Description** text box (e.g., Automatic Database Verification).

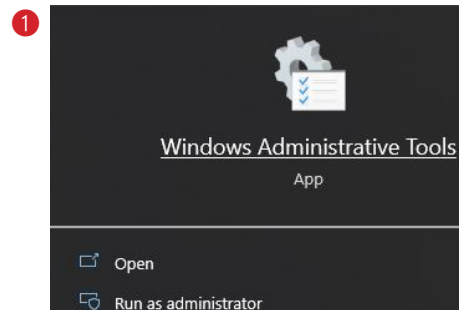


Figure 1. Navigate to Administrative Tools

- Under **Security Options**, select **Run whether user is logged on or not**

The remaining **Security options** within the **General** tab should be configured according to user preference.

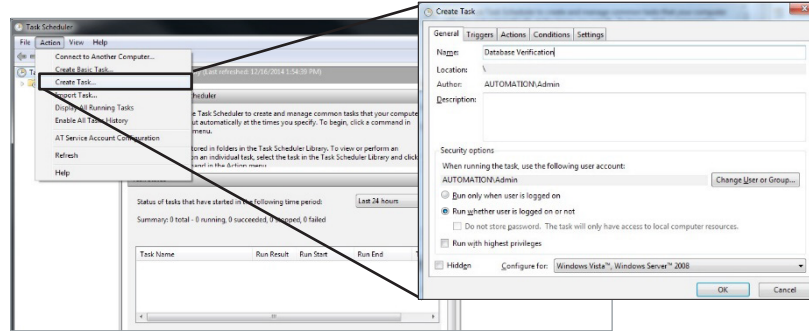


Figure 2. Select Create Task from the Task Scheduler window

- Select the **Triggers** tab from within the **Create Task** window and click the **New** button to open the **New Trigger** window (Figure 3)
- The options in the **New Trigger** window should be configured according to user preference and will control when Database Verification will run in the background. INVIA recommends scheduling this task at a time when the workstation is not in use (e.g., Sundays at 2:00 AM).
- Once the triggers have been configured, click the **OK** button.
- Next, create the action of running the Database Verification. Begin by selecting **Actions** from within the **Create Task** window and then clicking the **New** button to open the **New Action** Window (Figure 4).

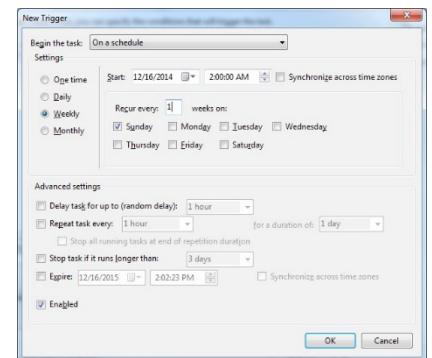


Figure 3. New Trigger Window

- Select **Start a Program** in the Action drop down.
- Locate the Database Verification executable to modify. Click the **Browse** button and browse to the **CBDDatabaseVerification.exe** in the Program Script field (default directory is C:\Program Files\INVIA\CorridorBrowser\CBDDatabaseVerification.exe).

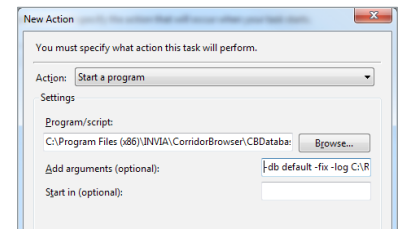


Figure 4. New Action Window

The default directory will be *Program Files* for a 64-bit OS and *Program Files (x86)* for a 32-bit OS.

- In the **Add Arguments (optional)** field, add the required and optional arguments from the table on page one.
 - At minimum, type in the required Database ID and the Results File arguments. Each argument and descriptive parameter should be separated with a space (e.g., `-db default -log C:\Results.txt`).
 - The `-db *` argument can be used to specify that all known databases are verified. Users that wish to apply different arguments to multiple, but not all, databases should create individual tasks for each database.
 - The arguments can be written in any order, but spaces must be inserted between the each argument and each descriptive parameter. Spaces within the descriptive parameters require the entire parameter be enclosed with quotation marks (e.g., `-db default -log "C:\INVIA_CBD\results.txt"`).
- Click the **OK** button.

10. The **Conditions** and **Settings** tabs provide additional configuration options for the created task and can be configured according to user preference.
11. Click the **OK** button within the *Create Task window*.
12. Click the **Task Scheduler Library** link to verify the new Database Verification task has been successfully created (Figure 5).

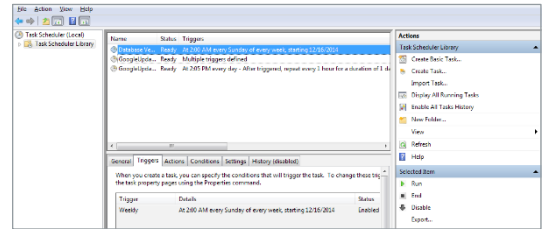


Figure 5. Verify the Database Verification added

VERIFICATION RESULTS

After the database verification utility runs, the results are automatically written to the text file that was designated in the command-line arguments while scheduling Database Verification through Microsoft Task Scheduler (Figure 6).

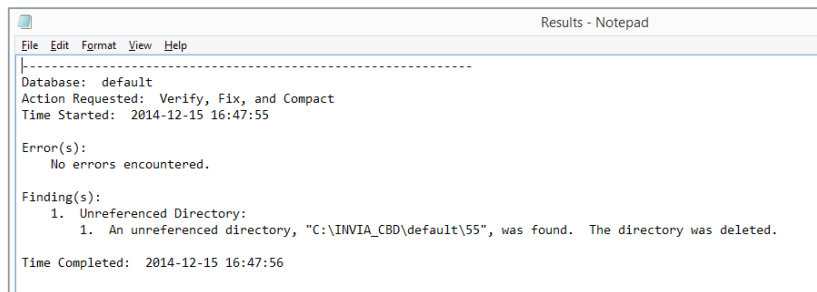


Figure 6. Verification Results

To access these results, locate the filename and directory selected in Step 9c above (e.g., C:\Results.txt). The results will display the following information for each database verified:

- The actions required on the database – verification, fixing, and compacting.
- The errors found – errors typically halt further verification and are not fixable.
- The findings found – these are fixable issues and the actions taken to fix the problem will be indicated if the fix argument was requested in the command-line.
- Indicates the beginning and end times for the Database Verification process.

The table below represents a list of possible errors and findings discovered through the Database Verification process and provides additional details regarding each verification event.

Verification Event	Type	Details
Invalid Database	Error	Indicates that the database version is not the same version specified in the configuration.
Not Supported Database	Error	Indicates that the database version is not supported by the verification process.
Database Not Found	Error	Indicates that the specified database file can't be found.
Remote Database	Error	Indicates that the specified database is on a remote network. Only local databases can be verified.
Open Database	Error	Indicates that the specified database file could not be opened.
Lock Database	Error	Indicates that the specified database file could not be locked.
Unlock Database	Error	Indicates that the specified database file could not be unlocked.
Unreferenced Directory	Finding	Indicates a directory exists in the database directory that is not referenced by the database. Specifying the fix command deletes the unreferenced directory.
Unreferenced DICOM File	Finding	Indicates that a valid DICOM file exists in the database directory but is not referenced by the database. Specifying the fix command imports the file into the database.
Unreferenced Non-DICOM File	Finding	Indicates that a non-DICOM file exists in the database directory. Non-DICOM files are never added to the database. Specifying the fix command moves the file into the "Non-DICOM" directory. The "Non-DICOM" directory is within the database directory.
Patient Not Found	Finding	Indicates that a patient is referenced in the database but the files for the patient can't be found in the database directory. Specifying the fix command removes the patient from the database.
Instance File Not Found	Finding	Indicates that an instance/image file is referenced in the database but the file can't be found in the database directory. Specifying the fix command removes the instance/image file from the database.