README - Multi Database Restore Tool

A PowerShell-based GUI tool to restore multiple SQL Server databases from backup files (.bak). Designed for efficiency and bulk operations, it allows you to restore databases with optional renaming, validate server connectivity, and manage existing databases.

Features

* **Graphical User Interface (GUI)** for ease of use.
* **Multi-Database Restore**: Restore multiple databases from backup files in a single operation.
* **Database Renaming**: Optionally rename databases during restoration.
* **Server Connectivity Check**: Verify SQL Server connection before proceeding.
* **Progress Tracking**: Real-time progress bar and status log.
* **Bulk Actions**: Select/deselect all databases, drop existing databases, and auto-detect backup files.
* **File Path Customization**: Specify custom locations for backup files, MDF (data), and LDF (log) files.

Prerequisites

1. **PowerShell 5.1+** (typically pre-installed on Windows).
2. **SQL Server Instance**: Access to the SQL Server where databases will be restored.
3. **Permissions**:
   * Read access to the backup folder.
   * sysadmin or equivalent SQL Server permissions to restore/drop databases.
4. **.NET Framework 4.5+** (for Windows Forms components).

Installation

1. Save the script as MultiDatabaseRestoreTool.ps1.
2. Ensure the SQL Server instance is running and accessible.

Usage

Running the Script

1. Open PowerShell with administrative privileges (if required by your SQL Server permissions).
2. Run the script:

powershell

Copy

Download

Set-ExecutionPolicy RemoteSigned -Scope Process -Force *# If script execution is blocked*

.\MultiDatabaseRestoreTool.ps1

Interface Overview

*(Example UI layout)*

1. **Server Connection**:
   * Enter the SQL Server name (default: (local)).
   * Click **Check Connectivity** to validate the connection.
2. **Folder Paths**:
   * **Backup Location**: Path to .bak files.
   * **MDF/LDF Locations**: Target folders for database/log files.
   * Use **Browse** buttons to select folders.
3. **Database Grid**:
   * Click **Show DB Names** to auto-populate databases from backup files.
   * Check databases to restore/drop.
   * Optional: Provide new names for databases in the **New DB Name** column.
4. **Actions**:
   * **Select All**/**Deselect All**: Toggle all checkboxes.
   * **Restore Databases**: Start the restore process for selected databases.
   * **Drop Selected DBs**: Delete selected databases (irreversible!).
5. **Progress & Logging**:
   * Progress bar shows current operation status.
   * Status log displays detailed messages (success/errors).

Parameters & Options

* **SQL Server Name**: Defaults to (local). Use .\SQLEXPRESS for named instances.
* **New DB Name**: Leave blank to retain the original name.
* **REPLACE Option**: Automatically replaces existing databases during restore.
* **File Paths**:
  + Backup files must have .bak extension.
  + MDF/LDF paths must exist before restoring.

Examples

Restoring Databases

1. Enter server name and validate connectivity.
2. Set backup location to C:\Backups.
3. Set MDF/LDF paths (e.g., C:\SQLData and C:\SQLLogs).
4. Click **Show DB Names** to list databases from .bak files.
5. Check databases to restore, optionally rename them.
6. Click **Restore Databases**.

Dropping Databases

1. Select databases in the grid.
2. Click **Drop Selected DBs** and confirm.

Notes

* **Backup Files**: The tool scans all .bak files in the specified folder. Ensure backups are valid.
* **File Paths**: SQL Server service account must have write permissions to MDF/LDF folders.
* **Logs**: The status log is cleared each time the script runs. Copy relevant info before closing.

Disclaimer

* Use this tool with caution. Dropping databases or restoring over existing data **WILL CAUSE**

**DATA LOSS**.

* Test in a non-production environment first.
* The author is not responsible for data loss or misuse.