User community support forum for Apache OpenOffice, LibreOffice and all the OpenOffice.org derivatives

[Wizard] Create a new 'split' HSQL 2.x database

Post Reply Search this topic... 9 posts • Page **1** of **1**

[Wizard] Create a new 'split' HSQL 2.x database

by DACM » Thu Apr 18, 2013 9:33 pm

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

Edit: Various versions will be posted here as development progresses...

Download: Split_HSQLDB_2.3.2_Wizard_v3d.odb ...includes HSQLDB 2.3.2 (User Guide 2.3.2.pdf / Utilities Guide.pdf)

- **OVERVIEW:** This Base template (.odb) generates a 'split database' as necessary for proper data-durability and other considerations. It serves as a replacement for the 'New database' wizard in Base. **Simply download the template to an empty folder and open it in Base with macros enabled** (or follow step-by-step instructions below).

  The template combines a preset Base document (.odb) with the popular HSQLDB 2.x engine -- collectively providing one of the most powerful and flexible, cross-platform, database options available for the desktop today. Key features include: a new database creation wizard; drag-&-drop database portability; seamless HSQLDB 2.3.x integration; and a fix for the AutoValue setup bug in the Base GUI. Existing split-database users can add the macro (below) to enhance the portability of their own project. Likewise, a legacy 'embedded database' file may be converted to a 'split database' as necessary to avoid rampant data-corruption with Base.

  A split-database involves multiple files, but the template maintains everything within a dedicated *folder* for portability and backup purposes. This folder may be renamed or moved as desired, or distributed as a single zip-archive file. The macro-automation ensures seamless database access on any computer (Windows, OS X, Linux) while uniquely allowing these 'split databases' to be run along-side legacy 'embedded databases' as desired. These databases work with AOO / LibO / portable on any computer with a proper Java Runtime Environment (JRE / portable) installed -- as also necessary with Base for legacy 'embedded database' file support. This template is configured for single-user access, suitable for non-concurrent database access, but the database itself may be hosted in 'server mode' for *concurrent* read/write access through Base or other front-ends on a LAN or web-server in multi-user scenarios. Needless to say, **this database solution is much more powerful, flexible and robust than the default Base implementation**.

  However, initial table development is slightly *easier* with the default 'embedded-database' configuration due to better GUI support in Base. Specifically, the Table Designer (GUI) in Base exhibits some limitations with a 'split' database. Technically speaking, it comes down to the database connection driver; the Base Table Designer is better-tuned for the built-in SDBC driver used to connect Base with the bundled HSQL 1.8 database engine -- as opposed to other data-connectivity options including other SDBC, JDBC or ODBC drivers. Since HSQLDB 2.x requires 'JDBC' connectivity with Base, the Table Designer proves more limited. I say 'more limited' because the Base Table Designer proves inadequate in any case ('split' or 'embedded'), so workarounds are often necessary when managing table structures with this GUI tool. In any case, the Table Designer can be used to create, cut, copy, paste, insert *or delete* tables and fields (columns) in either configuration ('split' or 'embedded'). And as expected, we can specify individual field 'properties' (field name, length, data type, etc.) when creating a field. However, after saving the field to the table structure, it is no longer possible to *modify* these field-properties in a 'split' database, using the Table Designer GUI. In fact, *existing properties* will appear frozen (grayed-out) in the GUI when connected to a 'split' database. Thankfully we do have some [superior] alternatives. A GUI Table Designer such as we find in Base is simply a graphical tool used to generate and send the necessary SQL (DDL) commands to the 'back-end' database engine. These SQL (DDL) commands can be created manually and sent directly to the database engine using the Base SQL Console (Tools > SQL...). Novice database designers should also consider the excellent commercial GUI options (30-day free trial) available for 'split' database management. Otherwise, when relying solely on the Base Table Designer (GUI) to 'manage' table structures in a 'split' database, it becomes necessary to create a new field with all the necessary properties, while eventually deleting the old field. Of course, once we've added user-data, relations or constraints to a table, it can be more difficult by any means (including SQL/DDL) to modify or delete a field within that table. In particular, it may be necessary to suspend or remove associated table relations before *deleting* a field. In any case, it's a simple matter to copy data between compatible fields (similar data-type, adequate length, etc.) using SQL. All of that to say, we have several options including SQL, commercial GUI alternatives, and/or the Base GUI workarounds should it becomes necessary to 'modify' existing table structures in a split database, *but* some users might find it easier to begin with an 'embedded database' file for table-structure development, while converting to a 'split database' configuration *before* adding critical data to the tables.

  In any case, this 'split HSQL database' solution is more than adequate and much easier to setup and use with Base than similar options including PostgreSQL, MySQL or Firebird. Thanks goes out to Villeroy for this critical tip because it's paving the way for some major breakthroughs in Base user-community support, such as this.

Instructions for use:

(1) Ensure global macro security in *Office is set to Medium (or Low):

- **Tools (*Office) > Options (Preferences) > *Office > Security > Macro Security > Medium**

(2) Remove any global Class Path to *hsqldb.jar* that you may have setup manually in *Office:

- **Tools (*Office) > Options (Preferences) > *Office > Java/Advanced > Class Path > Remove**

(3) Create a new folder on your computer to serve as a *dedicated* split-database folder (one per database)

(4) Download the above template and place this (.odb) file in the newly created folder.

(5) Then click to open the template in Base while selecting **Enable Macros** if prompted.

(6) Click the **Tables** icon in Base to begin (or optionally utilize SQL).

- The integral macro-wizard runs automatically. The macro initially prompts the user for a database name. It then adds a copy of the HSQLDB engine to the split-database folder ('driver' subfolder) and configures the template for single-user database access based on the current folder path. Upon clicking the **Tables** icon in Base, a connection is established to HSQLDB. The engine automatically creates a new database within the split-database folder ('database' subfolder). Thus, your entire Base project (including the database, engine, and Base front-end) is neatly contained within the dedicated folder you created in step 3 above. Feel free add standalone forms or additional subfolders (such as a portable 'images' subfolder) to your split-database folder.

  This entire split-database folder is portable. With Base shutdown, you can move or rename this dedicated folder as desired (drag-&-drop the entire folder). Or you can zip the entire folder for archiving or electronic distribution purposes as a single-file. The template serves as your Base (.odb) front-end file used for storing your Queries, Forms, Reports and Macros. You can rename this ODB file, but it must remain inside the dedicated folder to support database portability. The integrated macro runs quietly, for a split-second, each time you open the ODB file in Base, in order to ensure database connectivity based on the current location (folder path). If you never plan to move or rename your split-database folder (or files), you can safely bypass/disable the macro, since the current settings are stored within the template (.odb). You could theoretically disable the macro and then move the ODB file independent of a fixed database location (since the path is saved), but this can affect your ease-of-backup as well. So it's best to leave everything within the dedicated folder, and simply create a desktop-shortcut to this ODB file and/or folder as desired. You can also register the ODB file in *Office as a 'data source' (Tools > Options > Base > Databases) for seamless access (F4) from Writer or Calc documents, including standalone forms.

  Consider moving your entire split-database folder into a cloud folder (such as Dropbox, Google Drive or SpiderOak) to gain automated session backup with file-history and personal client-side encryption . A cloud folder can also provide *non-concurrent* internet access to the database.

Additional Information on the split-database concept, history, features, and limitations with Base:

- **BACKGROUND:** When 'Creating a new database' using the built-in Base wizard, a special Base (.odb) file is produced which encapsulates both front-end and back-end components of your database application including your queries, forms, reports, macros *and* tables. In other words, everything is stored within a single file (.odb). This all-in-one 'embedded database' concept is nice in theory, but the particular Base implementation is risky, and has proven highly-susceptible to data corruption. Fortunately, Base offers an alternative wizard dubbed '**Connect to an existing database**.' This wizard generates a more traditional Base (.odb) file containing only the connection parameters and other *front-end* components (limited to queries, forms, reports and macros). The *back-end* database tables and user-data are maintained as separate files in this configuration, as necessary for data-reliability. This configuration is quite flexible, supporting both flat-file and relational databases, as well as multi-user environments. **But users can get lost in the myriad of options and manual settings presented by the necessary connection wizard**. Actually, it's quite simple to create *flat-file* database-tables through Base by selecting the 'dBase' option in the Base wizard (dubbed 'Connect to an Existing Database'). But unbridling the potential of a *relational* database is another matter. Relational databases traditionally involve separate DBMS installation, setup, and management. These tasks include database creation and schema development largely using hand-coded SQL (actually DDL), in addition to daily chores such as engine start-up, shutdown, data-backup, etc. And that's to say nothing of the cryptic connection settings involved with Base connectivity.

  In response, **the Base user-community has turned to pre-configured Base (.odb) templates, allowing users to bypass the cryptic wizards**, while historically leveraging the bundled HSQLDB 1.8 engine. After all, this engine is reliable, seamless, portable, and enjoys tight integration and wide community support with Base. It generates new databases as necessary, through Base. It meets the database needs of most users, while providing an easy upgrade path to HSQLDB 2.x which enjoys a feature-set that rivals anything on the desktop today in both speed and function -- including billions of records, ANSI SQL, built-in functions, LOBs, encryption, and procedural SQL with triggers -- all for free. The HSQLDB engine scales modestly in multi-user environments complete with access control and hot backups. And as a bonus, HSQL databases are portable and cross-platform (Windows, Mac, Linux). This overall flexibility and SQL-power in an end-user product is unmatched today, even by MS Access (although the ease/appeal/depth of MS Access as a front-end is otherwise undisputed). The manual setup steps have been depreciated (in single-user configuration) because **some issues remain with a macro-free approach**. For instance, with a manual setup, database-folder portability requires tedious modification of the 'data source URL' associated with the Base (.odb) file. And while savvy users upgrade to the latest HSQLDB engine, this step can lead to unintended consequences for your legacy 'embedded database' files. Also, managing the required Class Path settings *globally* while ensuring HSQLDB version-compatibility across multiple computers, databases and *Office updates becomes a nightmare in practice. That is, until now...

  Enter an era of macro-enhanced Base templates featuring:

- 1. automated 'split HSQL database' creation wizard
  2. portable database-folders (automatic Class Path and Data-Source URL handling for seamless portability)
  3. minimal setup (limited to manually enabling macros in *Office)
  4. simultaneous support for all HSQL databases and versions (dedicated engine per database-folder; run along-side 'embedded databases')
  5. drop-in support of existing 'split' HSQL databases (add the template to an existing database folder or copy the macro to an existing .odb)
  6. built-in, light-weight, database management GUI with HSQLDB 2.x (click hsqldb.jar to start)
  7. built-in, light-weight, command-line, database access tool (sqltool.jar; see the SqlTool documentation for use)
  [future] 8. automated template begins as an 'embedded database' for initial table development, then expands on-command into a split-database for production-use as necessary for data-reliability
  [future] 9. easy switching between single-user and multi-user access modes
  [future] 10. templates optimized for large, multi-million record databases/result-sets (such as large outer-joins)
  [on hold] 11. "Pack & Go" user-selection for automated portability of 'split' databases as a single-file

LIMITATIONS: While this macro-enhanced template attempts to overcome some glaring deficiencies with Base, it scarcely addresses the holes in the Base GUI with respect to DDL tasks (database schema development) when connected through standard ODBC/JDBC drivers. As mentioned in the overview above, the Base Table Designer exhibits an anomaly when connected to HSQLDB in a split-database configuration. The Table Designer can be used to create Tables and Fields in the normal fashion. But **after saving the results, existing field-attributes appear frozen in the Table Designer**. So existing attributes can't be modified using this graphical tool. The Table Designer also fails to setup **Default field values** for most data-types. These deficiencies are limited to the table-design phase, when field-attribute modifications are essential but rare. So when modifying existing table-structures with a split-database of any kind, even novice users should resort to SQL, perhaps utilizing the Base SQL console (Tools > SQL...) or an alternative GUI database manager. Consider these interactive SQL tutorials: link1, link2. However, some GUI workarounds do exist:

> **Base limitations as a database manager:**
>
> - The Frozen 'Field Property' Workaround (adjusting existing Table structures with Base including **AutoValue** Primary Keys and **Default** values)
>   Upgrade to HSQLDB 2.3.x (a fix for the **AutoValue** setup issue in Base)
>   Alternative SQL database managers (free, cross-platform, cross-database, administration tools)

Additional Information for legacy 'split HSQL database' users:

- Existing 'split HSQL database' users should add the following macro to their .odb while removing the **global Class Path** to hsqldb.jar (as outlined in step 2 above). In this case, you'll also need to add a copy of your current HSQLDB engine to your split-database folder (choose between the default hsqldb.jar 1.8.0.10 -or- upgraded hsqldb.jar 2.3.2). You may also be interested in a copy of sqltool.jar for command-line SQL support (or server-mode management automation), as included with the template above. These steps will preserve your database-compatibility in all cases, including portable environments, while returning Base to its originally installed configuration. This has become important for a few reasons. **First**, a global Class Path setup is no longer necessary since we now rely on a portable 'session' Class Path (in single-user environments) derived by the macro; this not only enhances portability but also allows simultaneous support for legacy 'embedded databases.' **Second**, and most critical, a global Class Path now risks inadvertent upgrade of all your HSQL databases accessed with OpenOffice/LibreOffice, including legacy 'embedded HSQL databases.' With the debut of HSQLDB 2.3, Base no longer recognizes the distinction between its built-in HSQLDB 1.8.0.10 version and the newest HSQLDB releases. This is particularly dangerous because inadvertent upgrade of an *embedded* database renders the ODB file unusable. So it's time to end the practice of setting a global Class Path manually in OpenOffice/LibreOffice in support of HSQLDB in single-user environments (*multi-user* server-mode HSQLDB access still requires *global* Class Path setup...for now).

  Here's the source code (macro) found in the original **mydb_wizard.odb**. This macro is provided exclusively for existing 'split HSQL database' users in order to add the macro-enhanced features to their own .odb file while eliminating global Class Path use as outlined in step 2 of the 'Instructions for Use' above:

  CODE: SELECT ALL

```
REM  *****  BASIC  *****

'Instructions for use:
' (1) Macro security must be set to Medium (or Low) in: Tools > Options > *Office > Security > Macro Sec
' (2) Remove any Class Path to hsqldb.jar if setup in: Tools > Options > *Office > Java/Advanced > Class
' (3) Place this .odb file in a dedicated (empty) database folder
' (4) Add a copy of the HSQLDB engine (hsqldb.jar) to the database folder
'
' NOTE: If copying this code to your own .odb, press Alt-F11,
'              then select your .odb in Basic Macros (DO NOT add this macro to the global 'My Macros' l
'              then click: New (button) > New Module, then paste this code into the Basic code window,
'              then set to auto-run: Tools > Customize > Events > Open Document > Assign Macro > "Setup
```

  [/color]

Last edited by **DACM** on Tue Apr 26, 2016 10:19 am, edited 309 times in total.

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL database

by **Greengiant224** » Thu Apr 18, 2013 9:44 pm

Greengiant224

Posts: 284
Joined: Wed Jun 09, 2010 3:50 pm
Location: All Over The World

Hi DACM:

I have been looking at your post and would like to throw this into the mix. Looking through my 'old' batch files I find that we need the separate **sqltool.jar** to shutdown the hsqldb v 2.x server utilising the **--inlineRc** switch or a separate **.rc** file.

The **sqltool.jar** was separated from the **hsqldb.jar** for version 2.x and can be used as a standalone database/sql program.

Another macro maybe needed here that is bound to **Options>>Customize>> Document Close** we could then shutdown, and make the necessary backups at that time.

Kind regards,

Greengiant224

Win 7, Portable AOO 4.1.14, LibreOffice 6.2, 7.4.5 & 7.6.1 (Java 1.7.81 & 1.8.0_281) utilising HSQL 1.8.10 & 2.4.*, MySQL, PostgreSQL, SQLite + Blood, Sweat and Tears (Application, Determination and Perseverance)

Re: [Wizard] Create a new 'split' HSQL database

by **DACM** » Thu Apr 18, 2013 10:24 pm

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

Yes, thank you for all your inputs. And thank you for your own efforts in support of these configurations with HSQLDB.

This discussion is precisely why I stopped short of server-mode support in the current **mydb_wizard.odb** template.

- I'm aware of the requirements associated with server-mode support and HSQLDB 2.x. But remember that separate server-mode shutdown is only necessary as we deploy into multi-user environments. In single-user environments, the "shutdown=true" switch suffices at server start-up since this causes server shutdown upon closing the last connection (closing the .odb). Otherwise, I'm not even convinced that we need to shutdown the HSQLDB server using the .odb template as we actually deploy into multi-user environments, since automatic shutdown (through "shutdown=true") is the first thing we disable in those environments. We've historically supported server-mode shutdown in multi-user environments through a separate script (STOP.vbs, server.stop.bat, or your tray applet) because server shutdown must be very deliberate in those environments (as opposed to inadvertent due to over-automation). Proper automation might be as simple as prompting the user upon closing the front-end (.odb) file. But we would need to know whether the .odb is running on a client machine or on the database host, since that would affect the whole shutdown decision matrix.

However, even if we don't support server-mode *shutdown*, it seems useful to pursue some support for server-mode:

- Specifically, the ability to flip the .odb file between file- and server-modes seems useful. After prompting the user for server-mode data (host computer's network name or IP address; or Localhost for a single-user), we could even write a text file to the database-folder in order to eliminate, or at least pre-fill, these popups for subsequent mode switching.

  Also, it would still be nice to *start* the HSQL server using a macro, but only if hsqldb.jar supports a switch that checks for a running hsqldb-server instance before start-up. Otherwise, macro-code designed to perform this vital check wouldn't be cross-platform due to differing SHELL commands across platforms (Windows, Mac, Linux) [AFAIK]. As a minimum, we can probably check the platform with a macro, and offer to write the necessary batch/script files for server start-up and shutdown to the database-folder, while checking/prompting for sqltool.jar in the folder. I have some batch code for Windows, but not for Mac or Linux...?

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL database

by **DACM** » Sat Apr 20, 2013 12:49 am

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

Update: version 2 adds support for spaces and other escaped-characters in the database-folder name.

Download the updated **mydb_wizard.odb** file from the first post above. This is a drop-in replacement for the original in terms of database access. However, if you've begun development with a previous version -- specifically development of embedded front-end components such as Queries, Forms, Reports or your own Macros -- then consider replacing the old code with this new code as re-published in the first post above; simply copy-&-paste over the existing 'Setup' subroutine. This will allow you to take immediate advantage of the updated code.

And again, for those of you with an *existing* 'split' HSQL database that *pre-dates* the development of mydb_wizard.odb, you may also add this code to your current Base (.odb) front-end file. This will make your existing database folder portable, since the code updates the connection-settings as necessary. Simply copy-&-paste the code to a module within your named .odb container using the the macro development environment (Alt-F11 ; do not save to the global 'My Macros' container). Then auto-run this macro by name using: *Tools > Customize > Events (tab) > Open Document > Assign: Macro > "Setup"*

Enjoy!

Last edited by **DACM** on Sun May 05, 2013 11:54 am, edited 2 times in total.

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL database

by **Greengiant224** » Sat Apr 20, 2013 10:28 pm

Greengiant224

Posts: 284
Joined: Wed Jun 09, 2010 3:50 pm
Location: All Over The World

DACM:
Many thanks for your kind words, they are very much appreciated.

Just a small point I have found and it may confuse 😕 some users.

By using the Villeroy macro and the one as updated by yourself in the above post; any user, whilst using the mydb_wizard.odb or their own derivative goes to:

File>>Database>>Properties or right clicks in the bottom pane and selects

Database>>Properties

and then clicks on the **Test Class** and/or **Test Connection** buttons they will receive an error message

The JDBC Driver could not be loaded
and/or
The driver class org.hsqldb.jdbcDriver could not be loaded

This is because the **Tools>>Options>>Java>ClassPaths** have been disabled.

So far your macro looks good

Kind regards

Greengiant224

Win 7, Portable AOO 4.1.14, LibreOffice 6.2, 7.4.5 & 7.6.1 (Java 1.7.81 & 1.8.0_281) utilising HSQL 1.8.10 & 2.4.*, MySQL, PostgreSQL, SQLite
+ Blood, Sweat and Tears (Application, Determination and Perseverance)

Re: [Wizard] Create a new 'split' HSQL database

by **DACM** » Sat Apr 20, 2013 10:41 pm

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

> **Greengiant224 wrote:**
> **Test Class** and/or **Test Connection** buttons they will receive an error message

Yes, good point. If the error message is caused by the global Class Path setting in the *Office installation, then I'm afraid there's not much we can/should do about it. We could add a note somewhere to inform the user that those 'test' buttons in **File > Database > Properties** are no longer useful or valid. Maybe we could disable access to those properties at the menu level, just as we see now with an 'embedded database' file (.odb).

I can't help but think that we're doing -- with a macro-driven .odb file -- precisely what the devs should have done internally by now to support JDBC+HSQLDB -- perhaps as the default in Base. They could easily iron-out minor issues such as this, while also fixing the Base GUI to eliminate the need for workarounds with the built-in engine (JDBC+HSQLDB).

> **Greengiant224 wrote:**
> So far your macro looks good

Thank you so much for testing!

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL database

by **DACM** » Sun Apr 28, 2013 4:53 pm

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

Update: version 3a bundles HSQLDB 2.2.8 with the added support of Villeroy's portable FreeHSQL library.

Update: version 3b bundles HSQLDB 2.3.0 snapshot beta to preview the AutoValue fix for the Base GUI.

Download the new **mydb_wizard_2.2.8.odb** and **mydb_wizard_2.3.0.odb** files from the first post in this topic. These .odb templates incorporate the portable variant of Villeroy's FreeHSQLDB library. This embedded (portable) library adds support for the conversion of legacy 'embedded databases" into the reliable 'split database' folder concept. In addition, it supports a dedicated 'driver' per database-folder, in order to eliminate issues with database compatibility in portable environments involving multiple computers and *Office installations (appropriate Java 6 or later installation is assumed for database engine function). In particular, we're taking advantage of the dedicated driver support with these and future editions of the mydb_wizard.odb. At the moment, there's no added menu commands or other popups for user-level access to the legacy embedded-database extraction capabilities, but I would anticipate these features relatively soon in a future release of this mydb_wizard.odb template series.

The FreeHSQLDB library incorporates subfolders named "database" and "driver" respectively. These become integral to the main database-folder established by the location of mydb_wizard.odb. Accordingly, the "database" subfolder will contain your back-end HSQL data files, while "driver" will contain a dedicated *hsqldb.jar* driver/engine for each database. You'll also notice *sqltool.jar* in the "driver" subfolder as well, which becomes a player in multi-user environments or as a command-line tool.

- NOTE: If you have an *existing* 'split' HSQL database that you've also enhanced (using the code in the first post above) or created anew using mydb_wizard.odb, then please standby for future enhancements. With your macro-enhanced .odb, you can now rename or move your database-folder, as a unit through drag-&-drop, or as a single zip-archive file through: right-click: Send To > zipped folder (on Windows). ~~However, the plan is to add automated "Pack & Go" support using Villeroy's FreeHSQLDB library as these features are developed and tested~~.

NOTE: If you're still running a legacy 'embedded database' then stay tuned for user-level support in the form of a popup dialog offering to "*Unpack an embedded database*" in support of initial conversion, ~~and also "*Pack & Go*" to support portability of the resulting 'split database.~~' Unfortunately these user-level options/access are not yet fully implemented.

NOTE: **mydb_wizard_2.3.0.odb** includes a snapshot beta version of HSQLDB (hsqldb-20130412.214310-43.jar) which uniquely supports **AutoValue primary key setup using the Base GUI**. This has been a log-standing bug in the Base GUI, requiring SQL to workaround the issue. This .odb template implements the associated settings as recommended by **Fred Toussi** of the HSQLDB project. These Base settings are unique to HSQLDB 2.3.x, so they're not compatible with earlier versions of HSQLDB. Thanks goes out to Fred for incorporating this fix, in the absence of proper bug fixes in Base.

Enjoy!

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL database

by **DACM** » Sat Jul 13, 2013 10:05 am

DACM
Volunteer

Posts: 1138
Joined: Tue Nov 03, 2009 7:24 am

Update: version 3c bundles HSQLDB 2.3.0 including AutoValue setup through the Base GUI.

Update: version 3d bundles HSQLDB 2.3.2 including all previous features.

Download the latest template from the first post in this topic.

AOO 4.1.x; LO 4.2.x; Windows 7/8 64-bit
Warning: Avoid embedded databases --> Solution: Adopt a portable 'split database' folder
Soli Deo gloria

Re: [Wizard] Create a new 'split' HSQL 2.x database

by **DKO75** » Thu Nov 18, 2021 6:09 pm

DKO75

Posts: 2
Joined: Thu Nov 18, 2021 6:02 pm

Hi, this tool is very useful to me but I don't know if it is still maintained.

I have a problem when I place the folder on a network share, I get the following message:

Error 1: An exception occured :
Type: com.sun.star.ucb.InteractiveAugmentedIOException
Message: the name contained invalid characters. (line : 17)

Then I can't access to the database. I need to hardcode the path to the folder on the network share.

Does anyone know how I could fix this and automatically update the path to the folder even if I place it on a network share?

thank you for your help!

laurent

Libre Office Version: 6.2.7.1.lin1 (x64)
Build ID: ac167a92e33a5447f0bf604564addc465dbb4b35
Windows 10 Pro

Post Reply

9 posts • Page **1** of **1**

Return to "Base"

Jump to

Board index Contact us Delete cookies All times are UTC+02:00

Powered by phpBB® Forum Software © phpBB Limited
Privacy | Terms

Get OpenOffice