

SQLBrowser | Manual

Installation	5
Prerequisites	5
Attention: Installing a New Version Of The Service	6
Step 1 - Sqlbrowser License Server Installation	6
Step 2 - Send Server Information To Establish The License Key – To do just Once	6
Step 3 - Copy The Received License Key Files In The Server Bin Directory	7
Step 4 - Verify Sqlbrowser Client In The Server Bin Directory	10
Step 5 – Verify License On SQLBrowser Client	10
Renewal of the license key	10
Case 1 : license key is compatible with your installed sqlbrowser version	10
Case 2 : license key requires an upgraded sqlbrowser version	10
Reminder	11
SQLBrowser Client Installation (Windows ONLY)	11
Shared Drive Installation Of Sqlbrowser Client	14
License Key Input in SQLBrowser Client	14
Command Line Example	14
Sybase Ini File	15
Arguments Xml File	15
The Four SQLBrowser Directories to know	17
Installation Directory	17
User directory	17
Output Directory	17
Input Directory or URL	17
Getting Started	18
Quick Steps (version > 1481)	18
Setting the ini file in the options	19
Organizing Servers in Groups	19
Adding Synthetic Servers in Groups	19
Adding a Favorite Directory to the Favorite Files	19
Connecting to ASE using SSL	20

Performance Considerations	20
Inspecting a Stored Procedure.....	20
Trees Pane	20
Filtering	21
Objects Pane	22
Queries Pane	23
Variables Pane.....	24
Defects Pane	25
Debugging a Stored Procedure	25
Steps to debug a stored procedure	25
Editing SQL Files	27
Opening SQL Files	27
Editing SQL Files - Basic Actions	27
Read-Only Analyzed Files	27
Version Control.....	28
Editing Source Files	28
Pragmas	28
Acknowledge Defect Pragma	28
Dependency Pragma	29
Result Set Title Pragma	29
Cypher Property Pragma	29
Cypher Relationship Pragma	30
Code Templates	30
Running SQL Files	30
Running SQL Files	30
Run Options	31
Selecting Rows in the Result Window	31
Monitoring.....	31
Capturing TDS	32
Generating Insert/Update/Delete	33
Dependencies And Schema.....	34

How To Exploit Dependencies And Schema Files In Sqlbrowser	35
Dependency Pragma	35
Batch Analysis	35
How to set up Batch Analysis as a Windows Scheduled Task	36
"Applications" And Coloring	39
Best Practices	40
Store SQL Source files in Source Code Control	40
Schedule Batch Analysis with results on a shared drive	41
Log File	41
Troubleshooting	42
Reporting A Problem	43
Using SQLBrowser connecting to Production Sybase Servers	44
Known Bugs And Limitations	44
Contact	44
Regular Expressions Help	45
List of Detected SQL Defects	45

Installation

An SQLBrowser setup consists of :

1. An **SQLBrowser License Server** running on a Windows Server Machine
2. Several **SQLBrowser Clients** installed (unzipped only) on Windows Machines

Prerequisites

System Requirements

- The same requirement apply to the SQLBrowser Client as well as to the SQLBrowser License server
- SQLBrowser runs with Java 8 64-bit, so is subject to Java 8 own limitation listed here as of Jul 2017 <http://www.oracle.com/technetwork/java/javase/certconfig-2095354.html>

Windows Client

Platform	CPU Architecture	Version	Introduced In	Notes
Windows 10	x86 (32-bit)		1.8.0_51	
Windows 10	x64 (64-bit)		1.8.0_51	
Windows 8.x	x86 (32-bit)		1.8.0	Modern UI (i.e. Metro Mode) is not supported
Windows 8.x	x64 (64-bit)		1.8.0	Modern UI (i.e. Metro Mode) is not supported
Windows 7	x86 (32-bit)	SP1	1.8.0	
Windows 7	x64 (64-bit)	SP1	1.8.0	
Windows Vista	x86 (32-bit)	SP2	1.8.0	
Windows Vista	x64 (64-bit)	SP2	1.8.0	

Windows Server

Platform	CPU Architecture	Version	Introduced In	Notes
Windows Server 2016	x64 (64-bit)		1.8.0_111	
Windows Server 2012 R2	x64 (64-bit)		1.8.0	
Windows Server 2012	x64 (64-bit)		1.8.0	
Windows Server 2008 R2	x64 (64-bit)	SP1	1.8.0	

- Please Note that although installing on a Windows 32bits may work, it is not supported.
- As SQLBrowser is a Java program, it could be run under Linux but it is not supported.

IMPORTANT:

*** Step 1 below is for SQLBrowser version prior to 1407.

*** From version 1407, a Windows JRE 64bits is included in the bin directory and there is no need to care about java

*** If you are using a version >= 1407, skip java installation steps

1. Both **SQLBrowser License Server** and **SQLBrowser Clients** requires **Java 8** ([Java SE Home Page](#)). Please note that version **1407 and over** come bundled with a Windows JRE 64 bits to facilitate installation.
2. **SQLBrowser License Server** requires **admin rights** as it is installed as a **Window Service**
3. **SQLBrowser Client** installation does not require Admin Rights
4. **SQLBrowser Client** connect to **SAP ASE version 12.5 to 16**

IMPORTANT POINTS:

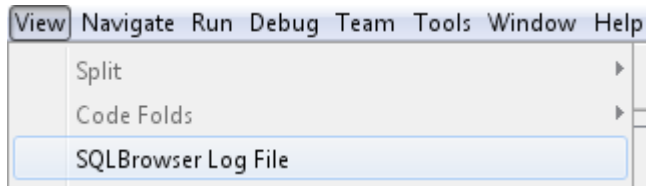
1. Please make sure each step succeeds before continuing to the next step

2. Should any step fail, send an email to support@sqlbrowser.com and always attach the 2 following files:

1. **File # 1:** The SQLBrowser client log file:

SQLBrowserlog.txt

You can view this file by using:



2. **File #2:** The SQLBrowser server log file:

SQLBLicenseServerLog0.txt

This file is in the bin directory of the server

3. It is practically impossible to resolve issues without these trace files. These files are technical logs of System Properties and SQLBrowser log traces and do not contain confidential information.

Attention: Installing a New Version Of The Service

This explains how to manage a service installation using SQLBrowser version, let's call it **V2**, when there is already an existing service running SQLBrowser, let's call it version **V1**. Normally, if the service name has not changed, the **service_install.cmd** will take care of removing the old service, however, to be cleaner, you can always remove the service running in V1 by using **service_stop.cmd** followed by **service_delete.cmd**.

Don't forget to copy the 2 arguments files from the old installation directory in case you need to conserve the old license files.

NB: some SQLBrowser Client users may get a "**Server Unreachable**" error during the operation. This will fix itself either if they restart their client or if they wait for the next polling loop which is scheduled generally every 5 minutes.

Step 1 - Sqlbrowser License Server Installation

1. Identify a **Windows Server** which is highly available to every user of SQLBrowser:

1. Make sure that this is a **production-grade** server
2. Make sure that no firewall or other network access restriction prevents access from the SQLBrowser user community
1. This host will run the **floating license server**.

This is a server process which distributes access tokens to SQLBrowser users.

- A. This host needs to be reachable from **all SQLBrowser user machines**
- B. This host will be **continuously running the license server** during the license period agreement

2. **Download the latest available SQLBrowser zip file**

3. **Unzip** the application on this windows server box in a directory called **SQLBrowser** for instance

Step 2 - Send Server Information To Establish The License Key – To do just Once

Note that this step is just required once. It outputs information which allows SQLBrowser to create the proper license key. You don't have to do this when you upgrade your server version

1. Open a **Command Prompt** in the **bin** directory of the **SQLBrowser** directory

2. Launch the command file: **start_server_manual.cmd**.

It will output lines which look like this:

```

main(1)      ===== This Host Info =====
main(1)      MAC-Address   : 42-61-73-D8-7A-29
main(1)      IP            : 192.171.21.19
main(1)      FREE PORT     : 16421
main(1)      =====

```

3. Copy and Send the 5 lines above by email to support@SQLBrowser.com

This information will be used to create the license key bound to this server

4. SQLBrowser support will send you 2 files which you will have to place in the **bin** directory

5. Important:

1. Make sure that the server that you define as the license server is the one you will use in PRODUCTION
2. SQLBrowser will not send a floating license for a different host during the contractual period
3. If you need to tryout the license server on a DEV box, you will be sent a temporary license key
4. In the exceptional case where the host running the license server is decommissioned, a new license key will be established as long as the IP Address is unchanged

Step 3 - Copy The Received License Key Files In The Server Bin Directory

1. Make sure you have **Windows Administrator Rights** on the Windows Server machine

2. Go in the directory where SQLBrowser is installed (from **STEP 1**)

1. Go in the **bin** directory

2. Copy the 2 files you received into this directory:




















A. "**arguments_license.xml**" this file contains the license key

NB: the key contained in this file must also be known by the client either by using this argument file or by typing the key directly in the Help / Register menu

B. "**distributed_arguments.xml**"

this is optional and contain preferences which will be sent to the client when the client connects and which override user preferences

3. This directory should now look like this:

	arguments.xml	1 Ko
	arguments_license.xml	1 Ko
	distributed_arguments.xml	1 Ko
	sb32.ico	4 Ko
	service_delete.cmd	1 Ko
	service_install.cmd	6 Ko
	service_start.cmd	1 Ko
	service_stop.cmd	1 Ko
	SQLBLicenseServerLog0.txt	1 Ko
	sqlbrowser	3 Ko
	sqlbrowser.cmd	1 Ko
	sqlbrowser.exe	377 Ko
	sqlbrowser_shared.cmd	8 Ko
	sqlbrowser_version.cmd	1 Ko
	sqlbrowser64.exe	1'361 Ko
	SQLBrowserLicenseServer32.exe	79 Ko
	SQLBrowserLicenseServer64.exe	102 Ko
	SQLBrowserLicenseServerw.exe	102 Ko
	start_server_manual.cmd	1 Ko

4. Open a **Command Prompt** in the bin directory

IMPORTANT:

*** Step 1 below is for SQLBrowser version prior to 1407.

*** From version 1407, a Windows JRE 64bits is included in the bin directory and there is no need to care about java

*** If you are installing a version >= 1407, skip the java steps

1. Verify that the java command has the minimum required version, by using **java -version**, e.g.:

```
C:\Temp\new\sqlbrowser_Dev_1327\bin>java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
```

In this case, java version is **1.8** which is OK as the version required by version 1327 is **1.7**

2. Launch **start_server_manual.cmd**, which should give an output like this:

```
C:\Temp\new\sqlbrowser_Dev_1327\bin>java -DargLicenseFileName=arguments_license.xml -Ddistributed
Log File is ./SQLBLicenseServerLog/g.txt
avr. 27 18:45:58.524 [01.188] main(1) ===== This Host Info =====
avr. 27 18:45:58.526 [00.003] main(1) MAC-Address : 
avr. 27 18:45:58.527 [00.001] main(1) IP : 192.168.1.8
avr. 27 18:45:58.528 [00.001] main(1) Free Port : 16421
avr. 27 18:45:58.529 [00.001] main(1) =====
avr. 27 18:45:58.530 [00.001] main(1) Distributed Arguments file passed is distributed_arguments.x
avr. 27 18:45:59.137 [00.607] main(1) LICENSE DATA: EXP: 20151222 MAX: 3 HOST
avr. 27 18:46:00.210 [01.073] SERVER(12) License Server started on port 16421
```

The important line is the last one: **License Server started on port 16421**

This line proves that the license server is running and accepting requests. It is useless to continue if this step is not successful.

3. Terminate the manual launch of the server by hitting Ctrl-C to abort the batch
4. Check the log file by using **type SQLBLicenseServerLog0.txt**. it should show the same as the console output above

5. Create the Windows Service:

1. Make sure you **Run As Administrator**. As this modifies Services, this is a pre-requisit.
2. Launch **service_install.cmd** and verify the data displayed before confirming:

```
C:\Temp\new\sqlbrowser_Dev_1327\bin>service_install.cmd

=====
This will install the SQLBrowser License as a WIndows Service
Are the variables below correct?

Java 32/64 Version:
Running Java 64 Bits

SQLBrowser bin directoy <this directory>:
SQLBROWSER_BIN=C:\Temp\new\sqlbrowser_Dev_1327\bin\

Full path of Java DLL jvm.dll:
PATH_JVM_DLL="C:\Program Files\Java\jdk1.7.0_45\jre\bin\server\jvm.dll"

Classpath:
CLASSPATH=C:\Temp\new\sqlbrowser_Dev_1327\bin\..\sqlbrowser\modules\com-sqlbrowser-sqlbm.jar

Hit Enter to continue
Hit Ctrl-C to break and modify this script
=====
```

3. If the data displayed is incorrect, please modify this script directly following the instructions found inside the script and retry until you the output is satisfying

4. Once it is OK, the screen should display several screens like this:

(This script attempts to stop and delete the existing service prior to creating a new one and starting it)


```

Press ENTER to STOP the service if already running...
Appuyez sur une touche pour continuer...
[SC] ControlService échec(s) 1062 :

Le service n'a pas été démarré.

Press ENTER to DELETE the service if service already exist...
Appuyez sur une touche pour continuer...
[SC] DeleteService réussite(s)

Press ENTER to CREATE the service...
Appuyez sur une touche pour continuer...

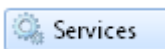
C:\Temp\new\sqlbrowser_Dev_1327\bin>SQLBrowserLicenseServer64 //IS//SQLBrowserLic
emp\new\sqlbrowser_Dev_1327\bin\ --LogPrefix=apacheDaemonLog --StdOutput=apache
enseServerMain --StartMethod=startService --StartParams=start --Classpath=C:\T
censeLogDir=. ++JvmOptions=-DlicenseDebugTrace=true --StopMode=jvm --StopPath=

Press ENTER to START the service...
Appuyez sur une touche pour continuer...

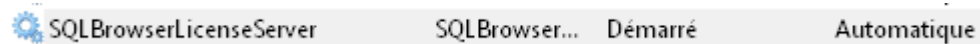
SERVICE_NAME: SQLBrowserLicenseServer
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 2  START_PENDING
                           (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0  (0x0)
        SERVICE_EXIT_CODE   : 0  (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x7d0
        PID                 : 4552
        FLAGS                 :
C:\Temp\new\sqlbrowser_Dev_1327\bin>

```

5. Launch the Services Windows Panel (Run Menu, Services)



Verify that the Service SQLBrowserLicenseServer is **Started** and in **Automatic** startup type



This will ensure that the service is automatically restarted when the box or VM is rebooted.

6. You may use the following scripts:

- A. **service_stop.cmd**
- B. **service_start.cmd**
- C. **service_delete.cmd**

to respectively stop, start or delete the service.

You don't have to use them in normal operations

7. When the service is running, verify the last line in the file **SQLBLicenseServerLog0.txt**:

<Date> SERVER(11) License Server started on port 16421

FYI: Linux Command Line Example

```

java -DargLicenseFileName=arguments_license.xml -DdistributedArgFileName=distributed_arguments.xml -
DlicenseLogDir=/apps/ log -DlicenseDebugTrace=false -cp ../sqlbrowser/modules/com-sqlbrowser-
sqlbm.jar com.sqlbrowser.ls.LicenseServerMain

```

Step 4 - Verify Sqlbrowser Client In The Server Bin Directory

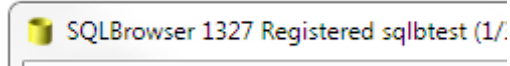
1. Launch:

```
sqlbrowser.exe --jdkhome "%cd%/jre" -J-Duser.name=sqlbtest -J-DargumentsFileName="bin\arguments.xml, bin\arguments_license.xml"
```

The first parameter sets the user to a dummy user called sqlbtest, with no preferences pre-stored anywhere

The second parameter gives the 2 files present in the directory as argument file names. One of them contains the license key

2. This should open an SQLBrowser client and the title bar should say:



3. Exit from the client

4. verify the SQLBLicenseServerLog0.txt server log file

type SQLBLicenseServerLog0.txt

```
apr. 27 20:00:00.695 [01.073] SERVER<10> License Server started on port 16421
apr. 27 20:00:27.796 [27.101] LS<14> sqlbtest 192.168.1.8 1327 320s: CONSUMES a license.
apr. 27 20:00:30.734 [02.938] LS<15> sqlbtest 192.168.1.8 1327 320s: RELEASED a license.
```

5. This has proven that the client has talked locally to the server and that they both understood the license key

Step 5 – Verify License On SQLBrowser Client

Please refer to SQLBrowser Client Installation (Windows ONLY) below

Renewal of the license key

A License Key file is produced for a **given version(s) of SQLBrowser** due to license key format change.

For instance:

1. Versions **595** and **844** requires '**type 1**' license keys
2. Version **1100** onwards requires '**type 2**' license keys
3. Version **1333** onwards requires '**type 3**' license keys

When receiving a License Key file, there can be 2 cases:

1. The key is compatible with your installed SQLBrowser version
2. The key requires an upgraded SQLBrowser version

Case 1 : license key is compatible with your installed sqlbrowser version

1. Copy the 2 license files received as per **STEP 3** above
 2. Launch the Windows **Service Manager**
 3. Stop the **SQLBrowserLicenseServer** service
 4. Start the **SQLBrowserLicenseServer** service
 5. Distribute the received key **arguments_license.xml** to all clients
- NB: there are 2 ways to enter the license key to the SQLBrowser client: (method 1 is preferred)
1. Pass it as an **argument file** (method used above)
 2. Typing it directly in **Help / Register**

Case 2 : license key requires an upgraded sqlbrowser version

1. If the key sent requires the **latest SQLBrowser Version**, apply the instructions above from **STEP 1 to 5, skipping STEP 2**

IMPORTANT: The new license server is able to serve older clients.

For instance, if you receive a License Key for Version 1333, you should update the server and have it running with the license. It will be able to serve non 1333 clients. This gives you time to update clients to the new SQLBrowser version and key

Reminder

You are bound in the contract to use the latest SQLBrowser version available.

This ensure a better experience for users as well as easier maintenance. Thanks!

SQLBrowser Client Installation (Windows ONLY)

Important:

- Please go to [Quick Steps \(version > 1481\)](#) and check if this works for you.
The steps below were necessary for older versions of the product

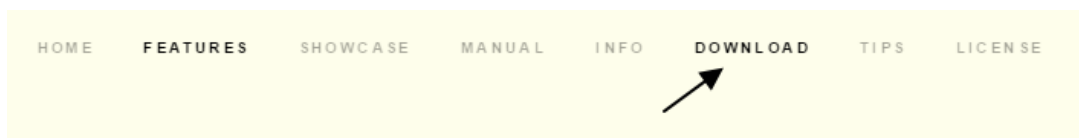
SQLBrowser does not currently have an installer. The distribution is a simple zip file that you have to unzip anywhere you like. If you install SQLBrowser for multiple users, please read **Shared Drive Installation Of Sqlbrowser Client** below.

The installation is done in 3 steps:

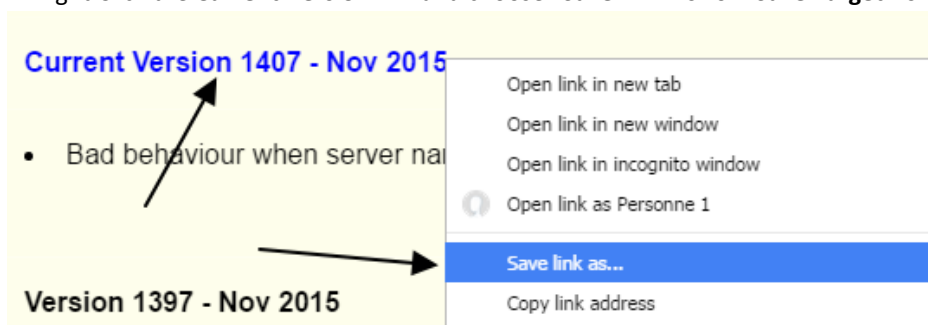
1. **Download And Unzip the latest available SQLBrowser zip file from the SQLBrowser website**
2. **[OPTIONAL STEP] Install JAVA**
3. **[OPTIONAL STEP] Change Arguments**

1. Download And Unzip the latest available SQLBrowser zip file from the SQLBrowser website

1. You **do not need Admin Rights** to do this
2. Open a web browser and go to: www.sqlbrowser.com
3. Click the **download** menu



4. Right Click the **Current Version** link and choose "Save Link As" or "Save Target As" depending on your browser



5. Save the zip file anywhere on your disk, e.g. on **C:\temp**
 6. **Unzip the zip file** in a directory where you want the installation to reside, e.g. **C:\SQLBrowser**
 7. This will create a **version-specific subdirectory**, e.g. '**C:\SQLBrowser\sqlbrowser_2015_05_1333**'
This is normal. Keep the version directory. This will allow to switch to a newer version while keeping the ability to roll back to the previous version.
 8. Go in the **bin** directory e.g. '**C:\SQLBrowser\sqlbrowser_2015_05_1333\bin**'
 9. Copy the **arguments_license.xml** file which you received from SQLBrowser in this directory (you can also register by hand but using an arguments file is the preferred method)
- NB: there are 2 ways to enter the license key to the SQLBrowser client:

- A. **Pass it as an argument file** (method used above)
- B. Type the key directly in the **Help / Register** menu

2. [OPTIONAL STEP] Install JAVA

If the OS on which you run is incompatible with the bundled **Windows JRE 64 bits** provided (for example, running on a 32 bits OS), you will need to **install Java manually** and modify the **sqlbrowser.cmd** script located in the bin directory. The 2 steps below show how to this.

Most of the time, you will want to use the bundled Windows JRE 64 bits bundled with SQLBrowser and you should skip the 2 steps below.

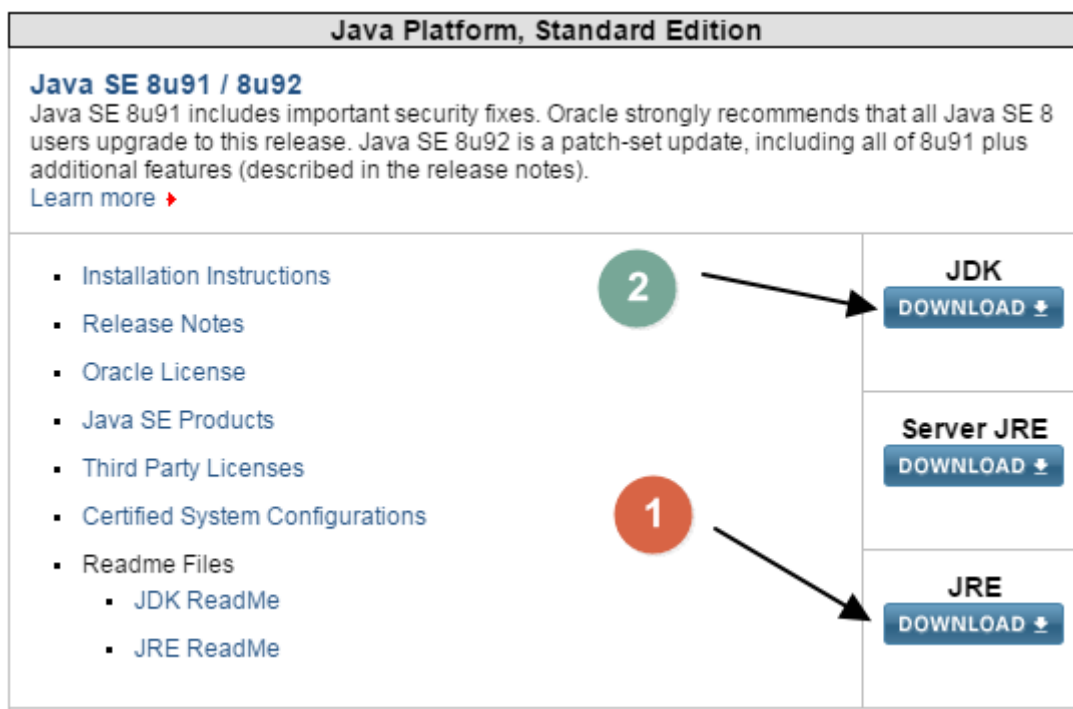
1. Java installation

A. We recommend to install the latest Java available. If you want a specific version, make sure that it is compatible with sqlbrowser requirement which is version **1.7** for SQLBrowser version **1407**

B. Open a web browser and go to:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

C. This should show a page like this:



D. SQLBrowser only requires a **JRE** (Runtime Environment) to run: click **JRE Download (1)**

However, you can install the **JDK** (Development Kit) if you prefer: click **JDK Download (2)**

E. The following screen looks like this



F. Accept the License agreement (1)

G. Download the installer (2) **Example** given here: **Windows 32 bits (x86) Offline Installer**

H. **Follow the Standard Installation Instructions**

2. Modify the "sqlbrowser.cmd" to accommodate the manually installed Java

A. Go in the **bin** directory e.g. 'C:\SQLBrowser\sqlbrowser_2015_05_1333\bin'

B. Edit the file **sqlbrowser.cmd**, e.g. with Notepad

C. Go to the last line which launches sqlbrowser. It should look like:

```
start sqlbrowser.exe --jdkhome "%~dp0jre" -J-Xmx900m -J-DargumentsFileName="bin\arguments.xml, bin\arguments_license.xml" --userdir "C:\temp\SQLBrowserUserDir1407"
```

D. Change the **-jdkhome** argument, in this case "%~dp0jre", and replace it with the **JRE location which you just installed**. This is generally located in **C:\Program Files\Java** for 64 bits versions, or in **C:\Program Files (x86)\Java** for 32 bits version. For example, if you have installed a 32 bits jre 1.8, it will be in **"C:\Program Files (x86)\Java\jre1.8.0_66"**. As a result the launching line should for example look like:

```
start sqlbrowser.exe --jdkhome "C:\Program Files (x86)\Java\jre1.8.0_66" -J-Xmx900m -J-DargumentsFileName="bin\arguments.xml, bin\arguments_license.xml" --userdir "C:\temp\SQLBrowserUserDir1407"
```

3. [OPTIONAL STEP] Change Arguments

1. Edit the **sqlbrowser.cmd** file and change the parameters of the **sqlbrowser.exe** command to suit your needs:

Argument	Meaning	Examples
start sqlbrowser.exe		start sqlbrowser.exe
IMPORTANT: *** The jdkhome argument below is for SQLBrowser version prior to 1407. *** From version 1407, a Windows JRE 64bits is included in the bin directory and there is no need to care about java *** If you are installing a version >= 1407, you should use the bundled jdk (Windows Only) *** The cmd launcher in the bin directory shows how the jdkhome is set to use the bundled JRE		

--jdkhome <JDK_HOME>	The root directory of JRE	C:\Program Files\Java\jdk1.7.0_45
-J-Xmx<Memory>	The max memory allocated	-J-Xmx800m (for 800 Mb) -J-Xmx5G (for 5 Gb, requires 64bit Java)
--userdir <UserDirectory>	Directory where SQLBrowser saves user preferences, windows positions etc...	--userdir "C:\temp\SQLBrowserUserDir"
-J-DargumentsFileName=<arg file name, arg file name...>	Absolute file name containing preferences. You can pass several comma separated files	-J-DargumentsFileName="c:\SQLB\arguments.xml" -J-DargumentsFileName="C: /SQLB/arg_license.xml, C:/ SQLB/arg.xml"

3. **IMPORTANT:** If the user had a previous version of sqlbrowser installed, it is preferable that the **<userdir>** of the 2 versions are different. (see [SQLBrowser Directories](#) to understand what is and where is the user directory)

4. For the first start, please read the [Getting Started](#) section.

Shared Drive Installation Of Sqlbrowser Client

It is advised to install SQLBrowser in a directory located on a shared network drive.

Doing so provides the following benefits:

- All users use the same version of SQLBrowser
- Version upgrades are done in a single place
- User can share globally defined preferences
- User can share the registration key
- Users can share the same launcher (Java Memory Size, User Directory location etc.)

To perform a network installation:

1. Choose a folder on a shared drive which SQLBrowser users can access
2. Unzip the distribution in this folder
3. Create a cmd file **sqlbrowser_shared.cmd** (see example in the bin directory)
4. You may have to modify it to accommodate your needs.
5. To share preferences, provide an argument file on the command line containing these preferences.

License Key Input in SQLBrowser Client

1. The license key to use in SQLBrowser Client is the same as the one used by the License Server.
2. It has been sent in a file named **arguments_license.xml**
3. There are two ways to input this key in the client:

1. Paste this key into the **Help / Register** Menu Entry
2. Use an **arguments.xml** containing the snippet:

```
<option name="/com/sqlbrowser/sqlbm/licenseData" value="XXX"/>
```

for more info on arguments file see [Arguments xml File](#)

Command Line Example

You can find a command line example file, **sqlbrowser.cmd** in the bin directory. e.g.: (version >= 1407, bundled JRE)

```
start sqlbrowser.exe --jdkhome "%~dp0jre" -J-Xmx900m -J-DargumentsFileName="bin\arguments.xml, bin\arguments_license.xml" --userdir "C:\temp\SQLBrowserUserDir1407"
```

Sybase Ini File

The Sybase Ini file is mandatory in SQLBrowser.

1. The Sybase ini file is part of your **Sybase Open Client** software.
2. It contains the definitions (host & port) of the Sybase servers.
3. SQLBrowser users may also have installed **Sybase Open Client**. In this case the ini file resides in the distribution directory.
4. Otherwise get a copy of an ini file.
5. Set up the path to the ini File in **SQLBrowser Menu: Tools / Options / Database**.

Arguments Xml File

The arguments.xml file is the way to load SQLBrowser preferences from a file. It is typically used for the registration key, but not only. You can store any setting that you set interactively in the Menu **Tools / Options**. This file become more interesting if it is shared across users: it allows to **setup group policies**. To share this file you can use the shared drive approach described in this manual, or you can share it through any packaging approach that your company uses. You can think about the way by the sql.ini file is distributed across client workstations to find a similar way to share an arguments file. It is not uncommon to dedicate one argument file (say arguments_license.xml to only hold the license information) - you can pass multiple argument file names. You can check the sqlbrowserlog.txt to verify that the argument file passed was indeed parsed. In case it is not, try passing an absolute file name.

1 . Argument File Name Format

The argument file can be specified either as a regular file name or as URL. Examples:

Format	Example
Regular File	C:/temp/arguments.xml
URI for local file	file:///C:/temp/arguments.xml
URI for remote file	file://remotehost/remotedir/arguments.xml
URI for file on web server	http://remotehost/remotedir/arguments.xml

2 . File Format

```
<arguments>
  <option name="myoption" value="myvalue"/> [...]
  <batch_analysis server="myserver" databases="db1 db2" crdate="31/01/1996" />
</arguments>
```

3 . Generating an arguments.xml file from the current preferences

Go to Menu **Tools / Options / SQLBrowser / Export / Export as Arguments.xml**.

This will generate an arguments.xml file which you can modify to your needs.

4 . Option names are:

- a netbeans specific option, e.g.
`<option name="/org/netbeans/core/proxyType" value="2" />`
 - an sqlbrowser option, i.e. any simple name found in the file produced by the menu, e.g.
`<option name="/com/sqlbrowser/sqlbm/registeredUser" value="Ngx"/>`
 - a JConnect Property option, e.g.
`<option name="/com/sqlbrowser/sqlbm/jcon_PACKETSIZE" value="1024"/>`
- prefix the option name by the string "jcon_", the option name is the string described in the JConnect Manual, please note that

not all jConnect options can be overridden.

The batch_analysis option

This is a very important option which starts a non-interactive SQLBrowser.

See [Batch Analysis](#) for more information

5. Example:

```
<arguments>
  <option name="/org/netbeans/core/proxyType" value="1"/>
  <option name="/com/sqlbrowser/sqlbm/jcon_PACKETSIZE" value="1024"/>
  <option name="/com/sqlbrowser/sqlbm/prefs_ba_doGenerateScript" value="true"/>
</arguments>
```

6. Multiple Argument Files:

You can pass several arguments file names separated by commas. This can be handy to separate license from other settings for instance. Example:

```
-J-DargumentsFileName="C:/SQLBrowser/arguments_license.xml,
C:/SQLBrowser/arguments.xml"
```


The Four SQLBrowser Directories to know

Installation Directory

This is where you unzip SQLBrowser. It may be in program files e.g.:

C:\sqlbrowser\sqlbrowser_2015_05_1333

Or it can be a directory on a shared drive e.g.:

S:\sqlbrowser\sqlbrowser_2015_05_1333

Important: This directory must be new for every new installation of SQLBrowser. i.e. do not unzip SQLBrowser over an old distribution

How to find the Installation Directory:

Click 'Help / System Info' and lookup the **System Properties List** for **netbeans.dirs**.

User directory

This is where sqlbrowser saves the user preferences. This directory is by default located on:

C:\Documents and Settings\<user>\Application Data\.sqlbrowser\dev (XP)

or

C:\Users\<user>\AppData\Roaming\.sqlbrowser\dev (Win 7)

This directory is Hidden by default on XP. This directory contains your last windows locations, your templates, your server names etc... We strongly advice to re-point this directory to something like **c:/temp/sqlbrowserUserDir**

You do this by using the **--userdir** command line option

NB: You may have to delete this directory in case of trouble restarting SQLBrowser after an update.

This directory contains many configuration files, but the main SQLBrowser files that you want to save are under:

config\Preferences\com\sqlbrowser

How to find the User Directory:

Click 'Help / System Info' and lookup the **System Properties List** for **netbeans.user**.

Output Directory

This directory is specified in the **Tools / Options** menu and is used to produce:

- the SQLBrowser log file named **sqlbrowserlog.txt** (to send in case of issues) and
- the batch analysis HTML report

A typical location for the output directory is:

c:/temp/sqlbrowserOutputDir

How to find the Output Directory:

Click '**Tools / Options - SQLBrowser / General**' and lookup **Output Directory**

Input Directory or URL

This is the directory or URL from which the **dependencies.txt** and **schema.xml** batch analysis output files are read when a server is opened. **This directory is set per Server**. See [Batch Analysis](#) for more information

How to find the Input Directory:

Click '**Tools / Options - SQLBrowser / Servers**'

Getting Started

Quick Steps (version > 1481)

The installation is made simpler as the license file is no longer required on the client side.

Customization is done through the 3 environments variables below:

- **SQLB_licenseServerIpAndPort** is read by SQLBrowser executable and points to the license server so that the client fetches its license remotely.
- **SQLB_commandLine** is expanded in the **sqlbrowser.cmd** launcher file and allows the user to customize **memory** and **user directory** or any other command line arguments needed.
- **SQLB_argumentsFileName** allows the user to pass an argument file which is read before the one distributed by the license server.

Step	How	Syntax	Default Value	Example
License Server	Create the Env. Variable SQLB_licenseServerIpAndPort	<host>:<port>	none	169.28.71.43:16421
License Server Distributed Arguments File	Make sure that the License Server Distributed Argument File contains the license data .	Arguments	none	<option name="/com/sqlbrowser/sqlbm/licenseData" value="..." />
User Dir	Add it to the Env. Variable SQLB_commandLine	--userdir <location>	%APPDATA%/.sqlbrowser/dev	--userdir C:/temp/SqlbUserDir
Memory	Add it to the Env. Variable SQLB_commandLine	-J-Xmx<value>	-J-Xmx900m	-J-Xmx5G
Arguments	Create the Env. Variable SQLB_argumentsFileName	Arguments	none	http://host/arguments.xml
Output Dir	Through Option Menu	<directory>	%TMP%	C:/temp/sqlOutputDir
Ini File	Through Option Menu	<filename>	none	C:/temp/sql.ini

For example, the three variables could look like:

Environment Variables

User variables for Nicolas

Variable	Value
SQLB_argumentsFileName	C:\Work\SQLBrowser\Dev\sqlbm\src2\argumentsMini.xml
SQLB_commandLine	-J-Xmx4G --userdir C:/temp/SQLBrowserUserDirMini
SQLB_licenseServerIpAndPort	192.168.1.9:16421

Setting the ini file in the options

Set the ini file path in the Options menu ("File/ Options")

Go to Preferences / Database and set the Sybase Ini File to the full path name of the ini file, e.g. *C:\sybase1502\ini\sql.ini*

Organizing Servers in Groups

To move a server to a Server Group:

1. Right click in the server pane, and click **"Add Server Group"**
2. Give it a name (Highlight and hit **F2**)
3. Select one or many servers from the **"Ini File"** group
4. Click **"Cut"**
5. Click on the newly created server group
6. Click **"Paste"**

Note that you can still easily access servers, even if you don't use groups:

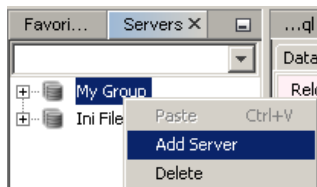
1. servers are naturally sorted by usage.
2. Rarely accessed servers can be accessed through the server drop down.

[ScreenShots](#)

Adding Synthetic Servers in Groups

You can create add a "synthetic" server in a Group, for example if it is not present in the ini file:

1. Select the Server Group in which you want to create a Server (You cannot create a synthetic server in the "Ini File" group. If you don't have a group created, create a group first as explained above.
2. Right Click **"Add Server"**



3. Fill in the name, host and port and click **OK**

This server will behave just like any other server present in the ini file

Adding a Favorite Directory to the Favorite Files

Right click in the favorites pane, and click "Add To Favorites"

Choose a directory with your SQL Files

[ScreenShots](#)

Connecting to ASE using SSL

1. Works from version **2019_01_1518**
2. Make sure that the server you want to connect to has SSL enabled (i.e. *sp_configure "enable ssl", 1...*)
3. Get the Server certificate in PEM format, say **cert.pem**. Go to SAP documentation to get more information
4. Insert this certificate inside the Java embedded **cacerts** file through a command like:
keytool -import -trustcacerts -alias a1 -file cert.pem -keystore c:/SQLBrowser/bin/jre/lib/security/cacerts
5. NB: An alternative is to use a specific cacert file by setting a specific environment variable, e.g.:
SQLB_commandLine=...-J-Djavax.net.ssl.trustStore=C:/xxx/cacerts
6. On the **Connection Params** Pop-Up Menu, / Connection Properties, Set the flag **ENABLE_SSL** to **true**

Performance Considerations

Performance Hints

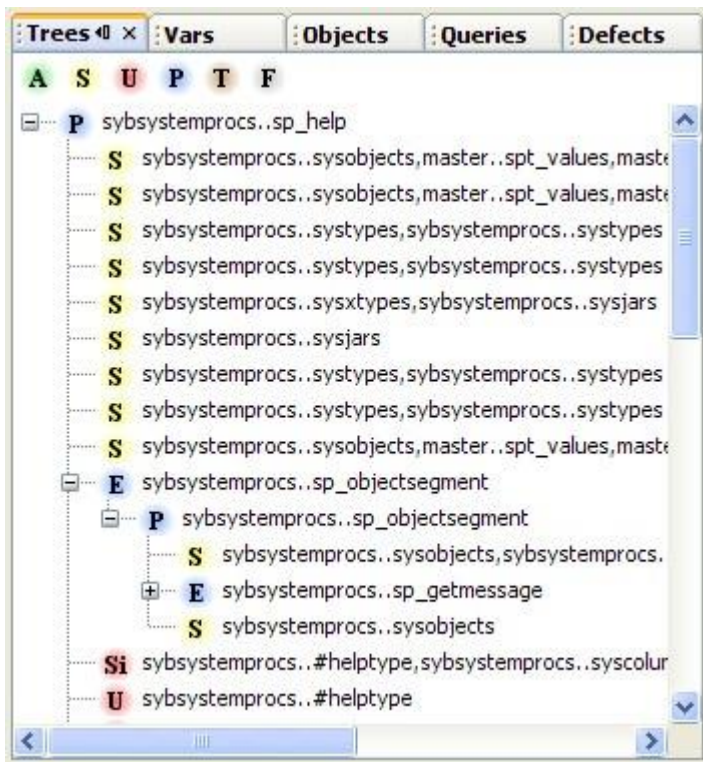
1. **Java**: Now bundled since version 1407.
2. **Memory**: Always pass enough memory on the launch script. SQLBrowser is greedy. Especially for Batch Analysis. Install a 64 bits Java to do large batch analysis.
3. **Local Drive**: Class loading from a network drive is slow. Always install sqlbrowser on a local drive. If you install on a shared drive, use the **sqlbrowser_shared.cmd** provided in the bin directory which copies the distribution to the local drive. See the [Installation](#) section for more information.
4. **Remove 'Auto Analyze Source Files'** option in Options / General if you find that it eats too much resources
5. **Anti-virus**: Batch Analysis produces a lot of files. Exclude SQLBrowser output directory from the antivirus write scans if applicable.
6. **Update Stats on System Tables**: SQLBrowser reads a lot from system tables. On ASE 15, it is sometimes necessary to run manual 'update statistics' on system tables (!) to speed up sqlbrowser. This was not necessary in V12.5.
7. **Subversion**: When linked with SVN, SQLBrowser can be slow to start. This is because of the number of SVN stats it launches (when either opened files are under SVN or favorites point to SVN directories)
8. **Monitoring Throttling**: To avoid sqlbrowser monitor taking too much resource on the server you can act on 3 parameters: (Options/Monitor)
 1. **Set the 'sleep time' to more than 1 second**
 2. **Set the 'stop after N iterations'**, to avoid it running unattended
 3. **Untick the 'Show Locks'** in the Option/Monitor Panel. If many row-level locks are used, this can help

Inspecting a Stored Procedure

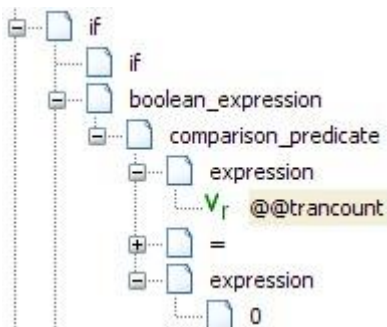
If you just need to inspect a stored procedure without willing to modify it, use the 'Analyze' feature Watch this [Animation](#)

Trees Pane

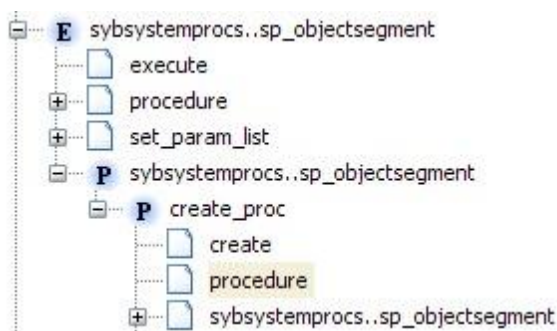
Analysing a stored procedure produces a parse tree, which shows 'interesting parts' of the stored procedure call tree. The left pane shows some selection (filters) of nodes from the full syntax tree. This allows you to quickly grasp what a stored procedure does. for example, this a typical syntax tree:



It shows the main statements contained in a procedure tree. The full syntax tree is the basis for all filtered trees and is a tree-style representation of the entire program, where nodes are syntactical elements of the Transact-SQL language. for instance an "if statement" like if @@ will be represented by the following tree:



The principle of SQLBrowser is to graft each stored proc body under each sub-proc execution node. For instance, since the stored proc sp_help calls the stored proc sp_objectsegment, the syntax tree will show the body of sp_objectsegment grafted under the execution node.



Doing so allows to see the proc effects in depth. Note that to avoid huge trees, only the first instance of a proc is grafted, i.e. you won't see a proc grafted twice.

Filtering

A S U P T F

Clicking on an icon filters the tree on nodes of interest:

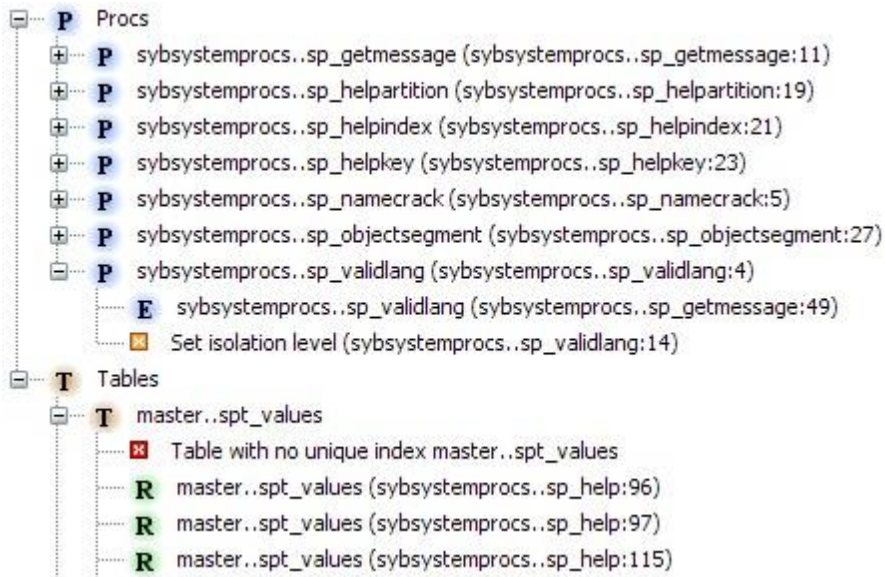
- **S** shows all selects which are result sets, i.e. things that go back to the client. Selects that are assignment to local variables are not shown. Using this filter, you can quickly see what result sets go back to the client.
- **U** shows all updates in the broadest sense of the word, i.e. all update, delete, select into or insert statements. This allows you to quickly see what side effects this proc causes. (A side effect in software jargon is a modification to the environment)
- **P** shows all sub-procs calls. this tree shows you all the sub-procs used by this proc
- **T** shows all transactions statements, i.e. begin tran, commit tran, save tran and rollback tran found in this proc tree. This allows you to see the transactional behaviour of a proc.
- **A** this filter shows all of the above. It gives a broad vision of what this proc does. It is the filter shown by default.
- **F** this filter shows the full syntax tree. It is of no great interest but shows the totality of the syntax.

Icon Meanings

- ◆ **S** select returning rows
- ◆ **Sa** select assignment into variables
- ◆ **Sq** select from subquery
- ◆ **So** other selects update insert delete
- ◆ **Si** select into
- ◆ **Qs** query specification
- ◆ **Qe** query expression
- ◆ **Qt** query term
- ◆ **R** table read
- ◆ **W** table write execute
- ◆ **P** stored procedure definition
- ◆ **Bt** begin tran
- ◆ **Ct** commit tran
- ◆ **Rt** rollback tran
- ◆ **St** save tran
- ◆ **Vd** variable declare
- ◆ **Vi** variable input param
- ◆ **Vo** variable output param
- ◆ **Vr** variable read
- ◆ **Vw** variable write

Objects Pane

This tab shows all objects (i.e. procs and tables) that are used throughout the proc execution. For instance, given a table, you can find immediately all accesses to this table. Tables indices are also shown in this tree.



Queries Pane

This tab takes all SQL queries (insert/select/update/delete) and presents them in a readable way. This is particularly convenient when you deal with very complex queries, where the "where clause" has been amended many times and is not easy to read. Here, the presentation of the "where clause" is ordered:

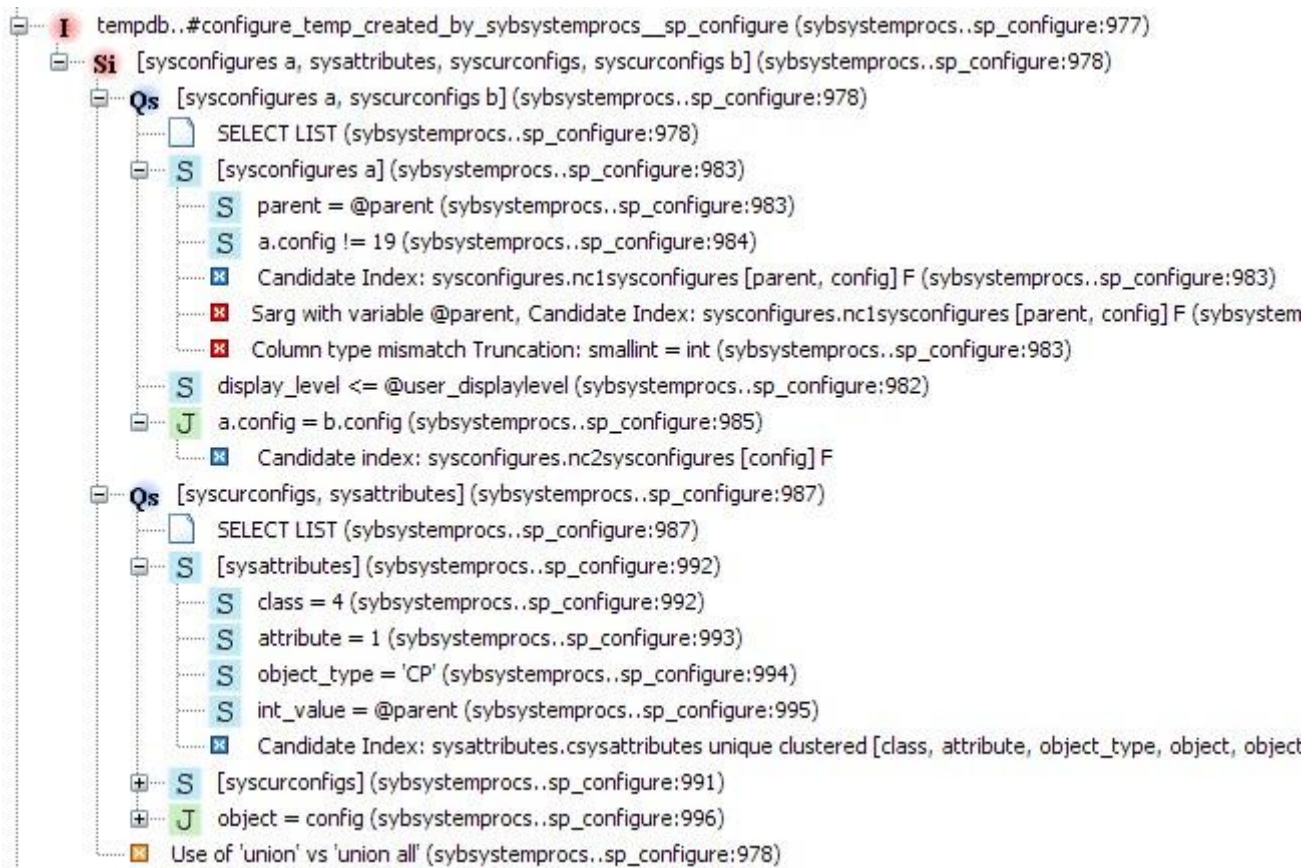
1. SARGS are shown first
2. SARGS are sorted by table (all SARGS on one table are shown close to each other)
3. JOINS are shown second
4. JOINS are sorted by table (all JOINS on two tables are shown close to each other)

for instance, the query:

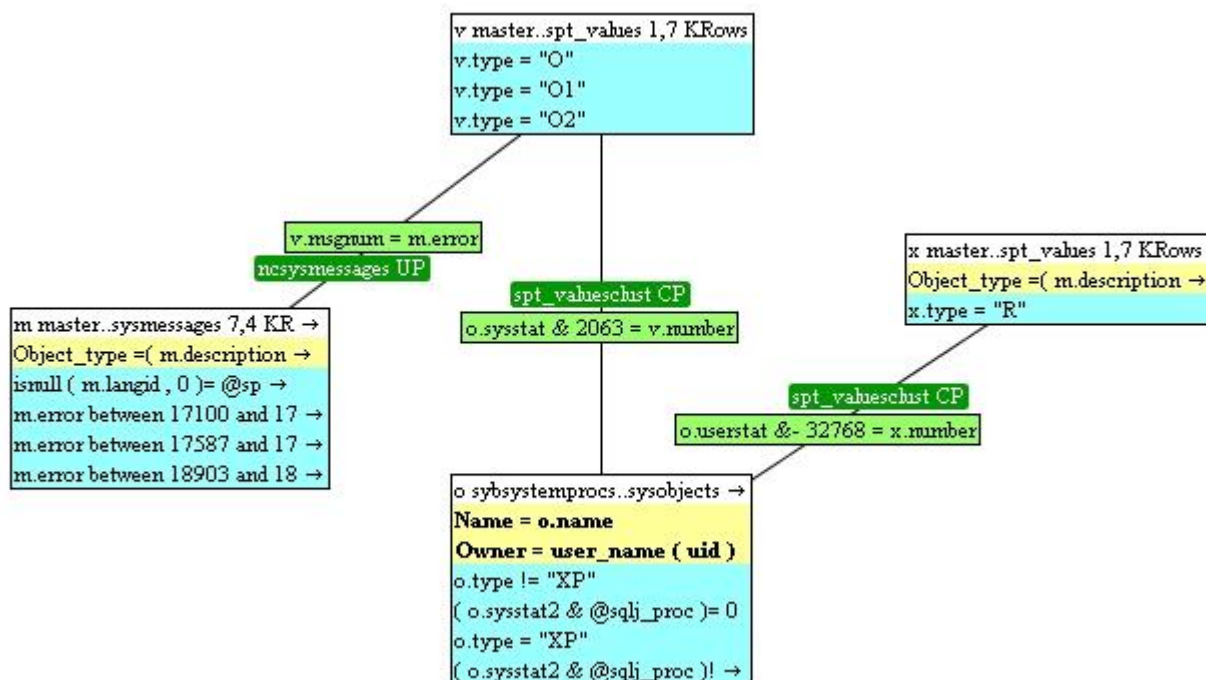
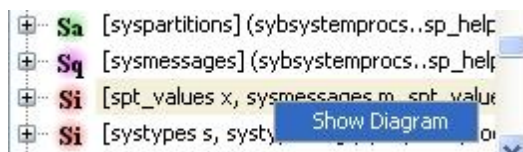
```

insert into #configure_temp
select a.config
from master.dbo.sysconfigures a ,
     master.dbo.syscurconfigs b
where
    display_level <= @user_displaylevel
    and parent = @parent
    and a.config != 19
    and a.config = b.config
union
select config
from master.dbo.syscurconfigs ,
     master.dbo.sysattributes
where
    display_level <= @user_displaylevel
    and class = 4
    and attribute = 1
    and object_type = 'CP'
    and int_value = @parent
    and object = config
    and config != 19
  
```

is broken down like this:



You can Right-Click on a query to display a query's graph:



Variables Pane

This tab shows all events that happens to the stored procedure variables. This tab is convenient when you want to track a parameter down to where it is used in deep sub procs. When variables are passed to a subproc, even if their name changes, the event tree of the variable in the subproc is grafted to the event tree for this variable:



Variable Events

- **V_d Variable declared:** the variable is part of a **declare** statement
- **V_r Variable read:** the variable is being read
- **V_w Variable written:** the variable is being written to
- **V_i Variable passed as input:** the variable is being passed (input only) to a subproc
- **V_{io} Variable passed as input/output:** the variable is being passed as an "output parameter" to a subproc

Defects Pane

This tab lists all potential defects found in this call tree.

[List of detected defects](#)

Note that this list is growing as more features are added to SQLBrowser.

Debugging a Stored Procedure

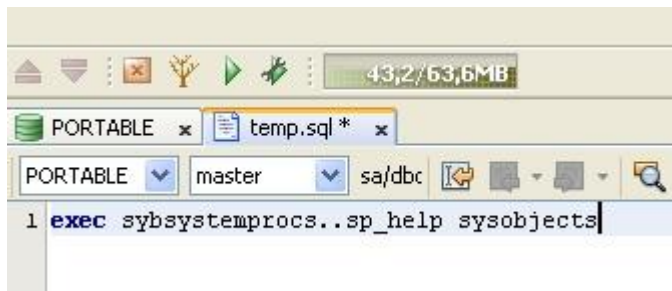
SQLBrowser implements a simple step-by-step debugger by merely sending executables statements one by one to the server. The state of local variables is kept, which allows for substitution when sending statements. There are two main usages of the debugger:

- to understand at what point a stored procedure does not behave the way expected
- to tune a stored procedure

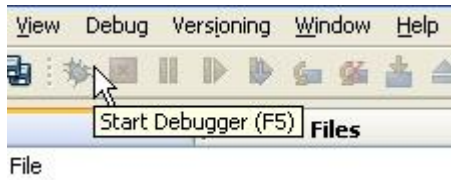
In both cases, local variables, cursors and temp tables are present in the context, which greatly helps debugging compared to doing it by hand.

Steps to debug a stored procedure

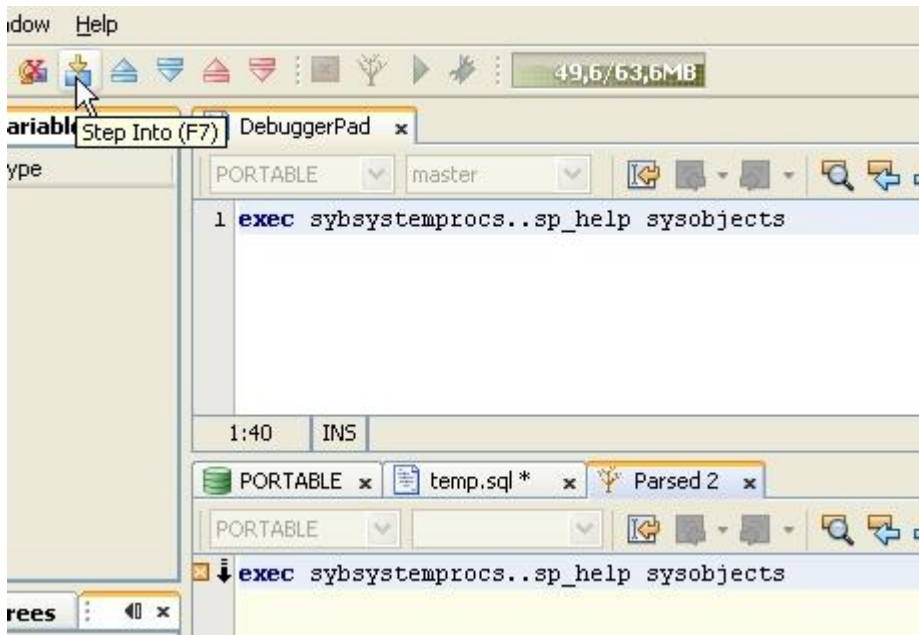
1. Open the server on which you want to debug a stored procedure
2. Open a New File and type the invocation of the stored procedure



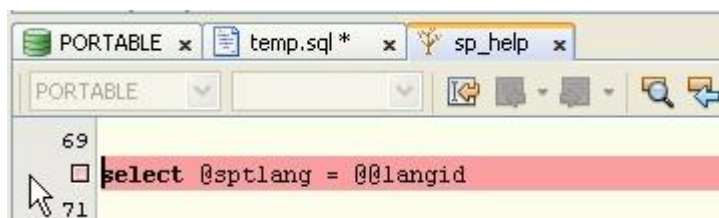
3. Click on Start Debugger



4. Click on Step Into to enter the debugged stored procedure body



5. Step into / over the executable statements
6. At any time, you can replay any statement
7. You can toggle break points by clicking in the line number gutter



8. Once you have reached the end of the debugged stored procedure, or if you press Stop before the end, you are presented a summary of all statements IOs. This helps spotting statements with long IOs

File Edit View Debug Versioning Window Help			
46,3/75,3MB			
Servers		Favorite Files	
Reads	Writes	Statement	Proc
1 493	0		
528	0	execute dbo.sp_helppartition @objname	sp_help
370	0	execute dbo.sp_helpindex @objname	sp_help
261	0	exec sp_autoformat @fulltabname = #helptype , @selectlis	sp_help
101	0	select column_name = isnull (c.name , 'NULL'), col_order =	sp_help
94	0	exec sp_autoformat @fulltabname = #sphelp5rs	sp_help
84	0	select name = o.name , owner = user_name (uid), object_	sp_help
12	0	execute dbo.sp_helpkey @objname	sp_help
8	0	update #helptype set length = length / @@unicharsize wh	sp_help
3	0	select @sysstat = sysstat , @sysstat2 = sysstat2 from sys	sp_help
3	0	exists (select 1 from syscolumns where id = object_id (@o	sp_help
3	0	select @sysstat2 =(sysstat2 & 57344) from sysobjects wh	sp_help
3	0	exec sp_getmessage 17578 , @msg out	sp_help

- To completely stop the debugger, press Stop again



Editing SQL Files

Opening SQL Files

There are 4 ways to open an SQL file for editing (in the worse-to-better order):

1. Use the **File / Open** Menu (slow, discouraged)
2. Find the file in the Operating System Explorer and **Drag** it to the editor (cumbersome)
3. Use the **Favorites** tab on the left to add Folders and Files (Right Click: Add to Favorites). If favorite tab is absent, make it appear in the Windows menu
4. Use the **Find Object** tab (**Ctrl-Shift-O**) and tick the '**Source**' Option as the default action for Enter. (You will need to setup the Sources Options in the preferences). **This is the most efficient way to open a source file**

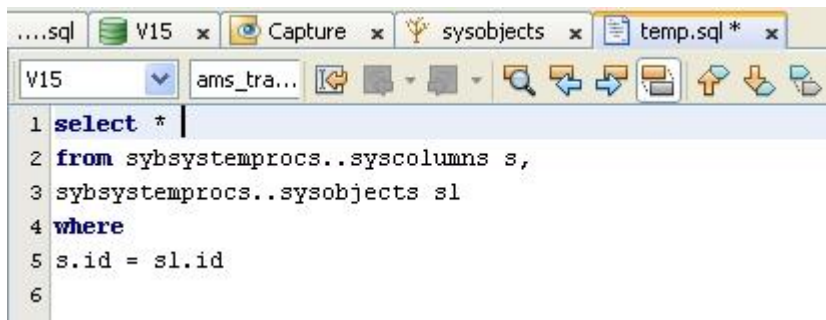
Editing SQL Files - Basic Actions

1. Use **Alt-Ctrl-Shift-F** to reformat the code
2. Use **Tab** or **Shift-Tab** on marked text to indent/unindent the marked block
3. Use **Ctrl-Slash** to comment/uncomment a block
4. Use **Ctrl-Click** on objects to jump to their definition
5. Use **Ctrl-Space** or **Ctrl-Tab** for auto-completion. For auto completion of joins, please refer to [Dependencies And Schema](#)
6. Use **Ctrl-K** to auto-complete any word that appers in the editor (convenient for temp tables names or columns for instance)
7. Use **Templates** to speed up code writing. For example, type `crp <Tab>` will expand a stored proc creation template. You can create your own templates in File / Option / Editor / Code Templates (Language:SQL)

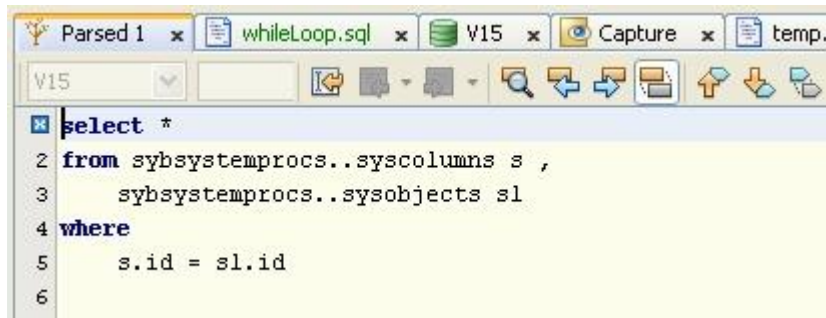
Read-Only Analyzed Files

SQLBrowser displays SQL code in two kind of files:

- **Editable Files:** this is what you obtain by opening an SQL file for editing.



- **Parsed Files:** This is the result of an *Analysis*. These files are read-only and presented in a yellow background window. These 'parsed' objects have associated analysis in the form of Trees on the left hand side. This is mostly historical and is rarely used as source files are automatically parsed.



It is important to understand the distinction between these two kind of files

Version Control

It is strongly advised that SQL sources are placed under Version Control.

To see how this works, hit F1 (Netbeans General Help) and consult IDE Basics / Version Control and File History.

Editing Source Files

SQLBrowser recommends to work on source files and not on extracted objects. To work on real SQL Source files, you need to:

1. Have your source files located under a well defined root on disk
2. It is advised to have the files organized in directories matching the object type and also the database. For example, C:\SQL\procs\database1\proc1.sql
3. Create the object-to-source file association in the **Tools / Options / SQLBrowser / Sources** panel
 1. Set the **Root Dirs** field to a comma separated list of root directories where the source files are. It is advised to only have one root directory.
 2. Create **Rules** to enable SQLBrowser to find the location on disk of a given object. See the explanations at the bottom of the panel. Rules are matched top to bottom (you can move rules up and down)

Pragmas

Pragmas are special tokens in comments which allows to convey certain informations to SQLBrowser

Acknowledge Defect Pragma

Abbreviation: pra<TAB>

Sometimes, SQLBrowser reports a defect that you know about and that you cannot/do not want to address. A typical example is a deliberate cartesian product. You can 'acknowledge' such defects. To do this, insert a pragma in the form of a comment just before the statement which generates the defect.

The pragma has the following syntax:

```
-- pragma acknowledge next defect <DEFECTCODE> comment_explaining_why_we_acknowledge
```

for example:

```
-- pragma acknowledge next defect QAFM key of currency is wrong but we know that
select @id_ccy = id_currency
from narval..currency where
cd_iso_ccy = @cd_iso_ccy
```

1. The pragma comment must appear before the token which provoked the defect
2. It can be just before the executable statement which provokes the defect
3. The pragma is considered valid only if it has a comment at the end explaining the reason of the acknowledge
4. Acknowledged defects appear differently (with a green dot) in both the editor and in the batch analysis
5. You can un-tick 'Detect acknowledged defects' in the Option / Defects pane to stop detection of such defects
6. The list of defects and their code is [here](#)

Dependency Pragma

Abbreviation: dep<TAB>

When using dynamic exec or dynamic SQL, it can be useful to declare the implicit run-time dependency(ies) induced by the dynamic call. To do this use the following pragma immediately before the dynamic exec or dynamic sql

```
-- pragma dependency (exec|select|insert|update|delete) <fully_qualified_name>
```

For example:

```
-- pragma dependency select tiny..AView2
exec ("select * from tiny..AView2")
```

Result Set Title Pragma

Abbreviation: tit<TAB>

Giving a title to result set makes it easy to spot results in a many result-set output. It gives also title to Excel sheets in Excel Export

```
-- pragma title <Result Set Title>
```

For example:

```
-- pragma title Sales for India
select * from sales_india
```

Cypher Property Pragma

Abbreviation: cyp1<TAB>

To be placed before a select expression in result set

```
-- pragma cypher (CREATE|MERGE|MATCH) property n NODE_LABEL propid [as lineLabel]
[unique]
```

Cypher Relationship Pragma

Abbreviation: cyp2<TAB>

To be placed before a select query

```
-- pragma cypher (CREATE|MERGE|MATCH) relationship n1 REL_TYPE n2
```

Code Templates

You can reduce the number of keystrokes when you are typing code by typing abbreviations to generate code from templates. The abbreviation is expanded into the template after you press the Tab key. SQLBrowser comes with a set of code templates. You can also create your own code templates. To create your own code template, go to File / Options / Editor / Code Template and select the SQL Language.

Do one of the following:

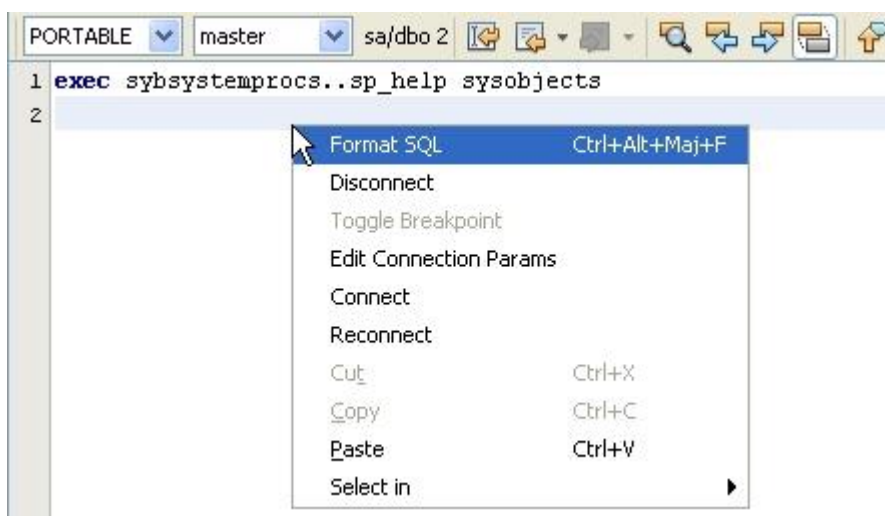
- Select an abbreviation from the list and type in the Abbreviation or Expanded Text field to modify the abbreviation.
- Select an abbreviation from the list and click Remove to remove the abbreviation.
- Click New and type the abbreviation and description in the respective fields.

You can use the Expand Template On drop-down list to change the key or key combination that is used to expand the abbreviations. By default, abbreviations are expanded when you press Tab. The Expand Template On drop-down list also contains the options of Space, Shift-Space, and Enter.

Running SQL Files

Running SQL Files

1. You can either open an existing SQL file or create a new file to type SQL statements. [Example](#)
2. Each window has its own connection, which is opened on first SQL run
3. Right Click in the window opens a menu which allows to change the connection state:



4. You can separate batches to be sent by using the classical *go* separator on its own line
5. You send SQL code by pressing the "Run SQL" button or by hitting F10

6. You can send only a portion of SQL by selecting a block of text first

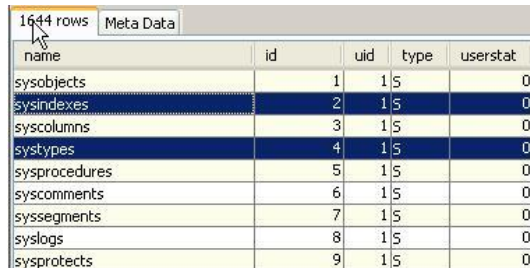
Run Options

1. Run Options allow to choose Sybase Options in order to view IO or Query Plan information
2. MonProcessActivity displays the Sybase MonProcessActivity table at each run. It requires:
3. `exec sp_configure "enable monitoring", 1`
4. `exec sp_configure "wait event timing", 1`
5. `grant role mon_role to my_user`
6. `set role mon_role on`

Selecting Rows in the Result Window

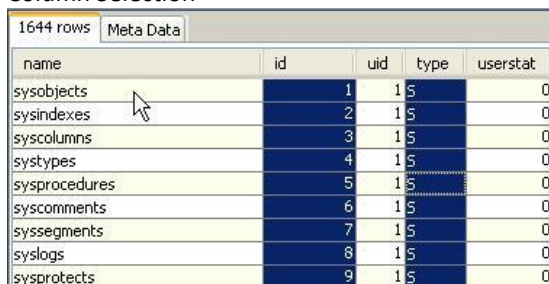
There are 3 selection 'modes':

- Row Selection



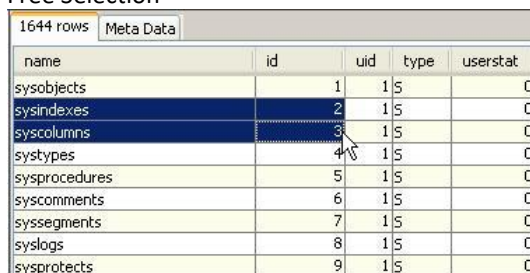
1644 rows	Meta Data				
name	id	uid	type	userstat	
sysobjects	1	1 S		0	
sysindexes	2	1 S		0	
syscolumns	3	1 S		0	
systypes	4	1 S		0	
sysprocedures	5	1 S		0	
syscomments	6	1 S		0	
syssegments	7	1 S		0	
syslogs	8	1 S		0	
sysprotects	9	1 S		0	

- Column Selection



1644 rows	Meta Data				
name	id	uid	type	userstat	
sysobjects	1	1 S		0	
sysindexes	2	1 S		0	
syscolumns	3	1 S		0	
systypes	4	1 S		0	
sysprocedures	5	1 S		0	
syscomments	6	1 S		0	
syssegments	7	1 S		0	
syslogs	8	1 S		0	
sysprotects	9	1 S		0	

- Free Selection



1644 rows	Meta Data				
name	id	uid	type	userstat	
sysobjects	1	1 S		0	
sysindexes	2	1 S		0	
syscolumns	3	1 S		0	
systypes	4	1 S		0	
sysprocedures	5	1 S		0	
syscomments	6	1 S		0	
syssegments	7	1 S		0	
syslogs	8	1 S		0	
sysprotects	9	1 S		0	

Switching between these modes is through:

- **Simple Click** - selects either the row, column or cell depending on the mode (Ctrl/Shift to add to selection)
- **Double Click** - switch to 'row selection' mode and selects the row
- **Triple Click** - switch to 'column selection' mode and selects the column
- **Click and Drag** - switch to 'free selection' mode and selects the block

Monitoring

Right click on a server node and choose 'Monitor' will launch the monitoring of this server. The Monitor shows a 'sp_who' like panel refreshed every second or so. It allows to spot long running processes

Spid	Login	Client Name	Client App	Prog Name	IOs	CPU	Mem	Database	Stored Procedure	Command
23	sa			isql	0	240	14	download		INSERT
24	sa			isql	0	211	14	download		INSERT
20	sa	null	SQLBrowser	SQLBrowser	0	32	4	propagation		AWAITING COMMAND
6					5 091	0	0	master		HK WASH
5					2 483	0	0	master		CHECKPOINT SLEEP
2					0	0	0	master		DEADLOCK TUNE
3					0	0	0	master		SHUTDOWN HANDLE
4				<astc>	0	0	0	master		ASTC HANDLER
7					0	0	0	master		HK GC
8					0	0	0	master		HK CHORES

spid	dbname	objectname	locktype	page
23	download	sysobjects	Shared intent lock	0
24	download	sysobjects	Shared intent lock	0
23	tempdb	#t00000230009969304	Exclusive table lock	0
24	tempdb	#t00000240009971062	Exclusive table lock	0
23	tempdb	syspartitions	Shared intent lock	0
24	tempdb	syspartitions	Shared intent lock	0
23	tempdb	syspartitions	Shared row lock	83
24	tempdb	syspartitions	Shared row lock	83

- **Display**
 - Active lines are always on top and painted in light red, unless you sort differently
 - Blocked lines are showed in strong red
- **Toolbar**
 - Sleep time is configurable
 - "Resume/Pause" suspends the monitoring
 - "Reset Sorting" resets the sorting to the natural sort, i.e. most active line first. This is when you have otherwise sorted by clicking on a column
 - "System Spids" tick box allows to show/hide system spids
 - "Trace" outputs a trace in the output window. **Warning, this consumes a lot of memory!**
 - "Back and Forth Arrows" navigate through history
 - "History" is the history size. **Warning, changing this to a large number consumes a lot of memory!**
 - "Clear" clears history
- **Contextual Menu**
 - "Mark or Unmark" highlights a spid or group of spids to follow them visually
 - "Show-Plan" launches a sp_showplan (or substitution proc) on the curent spid, provided you have the proper authorization
 - "Kill" kills the current spid, provided you have the proper authorization

Capturing TDS

Right click on a server and choose "Set as Capture Server" allows to consider the SQLBrowser process as a pseudo SQL Server which will carry forward each request to the server set up as the "Capture Server". To use the capture feature you need to follow these steps:

- On the client that you wish to capture, create a ini entry or an ODBC entry pointing to the host where sqlbrowser runs, default port is 8723 (see preferences). e.g.:

```
[SQLBROWSER]
```

```
master=NLWNSCK, PORTABLE, 8723
```

```
query=NLWNSCK, PORTABLE, 8723
```

- Launch SQLBrowser and define the capture server through the contextual menu of the server nodes
- Connect to this server. (don't worry about the user name, the gateway will use the original login name used in the client, however, a connection to the server is needed)
- Launch the client by connecting to the SQLBrowser datasource
- The "Capture Window" should show. If this window is closed, you can open it in the Window menu
- The capture traces are saved in temporary files in the output directory. Clicking on the file name opens the file in the Notepad



Unsetting the capture server

- Actually, the only way to un-assign the capture server from a server is to assign the capture to another server, (that you don't use that much for example)
- However, note that there is no inconvenience for a server to be set as the capture server other than the 'C' icon showing...
- This is why there is no 'unset' capability
- Now if you really want no capture server at all, shutdown SQLBrowser, edit
<userdir>/config/Preferences/com/sqlbrowser/sqlbm.properties
e.g. "C:\Temp\SQLBrowserUserDir\config\Preferences\com\sqlbrowser\sqlbm.properties" and remove the line starting with CaptureServer=XXX

Generating Insert/Update/Delete

It is possible to generate a file containing either insert, update or delete statements for each record of a table. You can narrow down the set of records for which to generate the statements by specifying a restrictive where clause. For example, given the table:

```
create table t (
  c1 int primary key,
  c2 int
)
```

with content:

```
c1 c2
1 11
2 12
3 13
```

If we specify `c1>1` in the where clause field, it will only generate updates for rows 2 & 3:

```
/*
Generated with SQLBrowser on 20 08 2009
Where Clause: where c1>1
*/
update propagation..t
set c2 = 12
where c1 = 2

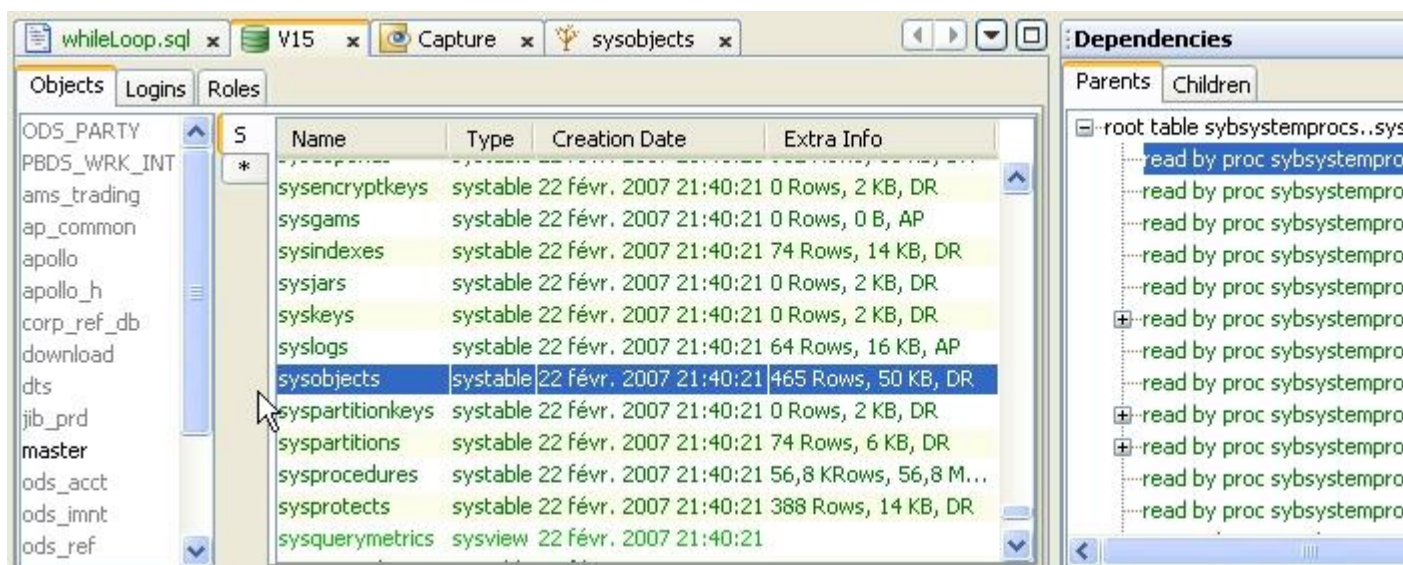
update propagation..t
set c2 = 13
where c1 = 3
```

The update and delete need to know the PK of the table in order to generate the where clause

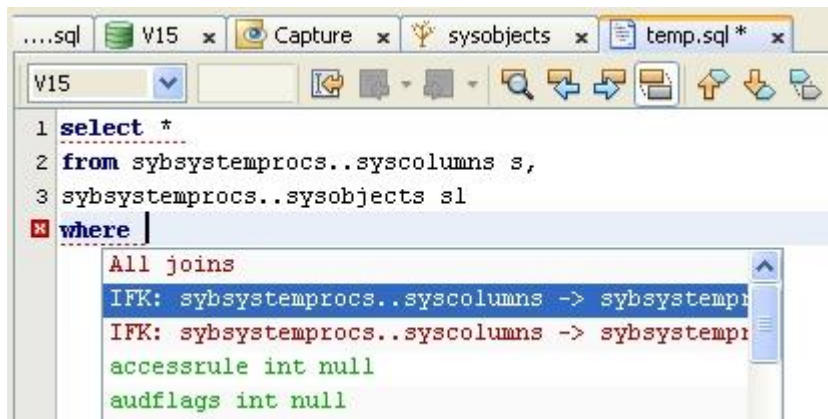
Dependencies And Schema

The Batch Analysis produces both a "Dependency File" which contains inter-objects cross reference information and a "Schema File" which contains foreign keys information. You can use these files in your own programs if you wish, the structure is quite simple. You can also exploits these files interactively directly in SQLBrowser:

- In the "Dependencies" Window: If dependencies are available, clicking on an object will highlight the callers and dependent objects. For convenience, you may as well double click on a dependent node to get it analyzed:



- In the editor: If the schema is available, you can use the completion:



How To Exploit Dependencies And Schema Files In Sqlbrower

- Understand how batch analysis works. Please refer to [Batch Analysis](#). Setup the output directory as you wish.
- Make sure that centralized and regular batch analysis are performed. We advise that batch analysis are organized as Windows tasks and performed on a contingency or development server.
- SQLBrower needs to know the output directory of the batch analysis to pick up the dependency and schema files. You need to specify this location in the General Preferences, in the "Input Directory or URL" preferences for each server (servers tab). There are two ways of specifying this location:
 - By a file name, if the output directory is accessible from the machine where SQLBrower is running. e.g.: D:/temp/SQLBrowerOutputDir
 - By a URL, if the output directory is accessible through a web server running on a different machine e.g.: http://remotehost/remotedir
- SQLBrower will attempt to open the files named *sqlbhtml/<ServerName>/dependencies.txt* or *sqlbhtml/<ServerName>/schema.xml*, rooted from the input directory or URL
- If successful, this will be the dependency file for the server opened
- If not, SQLBrower will attempt to open *dependencies.txt* or *schema.xml* rooted from the "Default Input Directory or URL" location. This is useful to put a dependency or a schema file common to different servers who may differ in content but whose programs are the same.

Dependency Pragma

When using dynamic exec or dynamic SQL, it can be useful to declare the implicit run-time dependency(ies) induced by the dynamic call. To do this use the following pragma immediately before the dynamic exec or dynamic sql

```
-- pragma dependency (exec|select|insert|update|delete) <fully_qualified_name>
```

example:

```
-- pragma dependency select tiny..AView2
```

```
exec ("select * from tiny..AView2")
```

Batch Analysis

Batch Analysis performs an analysis of one or several databases, and produces an HTML report which documents the entire model. To launch the batch analysis, select some databases in the object chooser (Use the usual Ctrl or Shift keys to select several databases), and right click "Batch Analyze". The HTML report produced is located in the output directory (set in the options) under the "sqlbhtml/<SERVERNAME>" directory.

An example of HTML output can be found [here](#)

The interest of the batch analysis is to run periodically as a scheduled task on a server. Doing so enables the developer community to point to the documentation page at any moment.

How to set up Batch Analysis as a Windows Scheduled Task

- create an argument file containing the <batch_analysis> tag, for example:

```
<arguments>
```

```
<batch_analysis server="NGX3" databases="db1 db2" />
```

```
</arguments>
```

- **<batch_analysis> tag attributes:**
 - **server** is the name of the server you need to perform the analysis for. The credentials used to connect will be the one used during the last interactive session on this server
 - **databases** is the list of white-space separated databases you want to scan
 - **anti_databases** is the list of white-space separated databases you don't want to scan (negative list approach) (use either **databases** or **anti_databases**)
 - **crdate** is an optional object creation date. Only objects created after this date will be parsed. This is to allow shorter batch analysis.

- **Other attributes:**

SQLBrowser takes all its options from either the user preferences saved in the 'userdir', or they can be overridden by values in the arguments.xml.

Go to Menu **Tools / Options / SQLBrowser / Export / Export as Arguments.xml**.

This will generate an arguments.xml file which you can trim down and modify to your needs.

For instance you can extract the **ba_XXX** options in order to retain the batch analysis options you want.

You also need to capture the **successful connection** for the desired server with the desired login.

You then end up with an argument file which looks like this:

```
<arguments>
```

```
<batch_analysis server="HPH8NGX_16SP02" databases="master" />
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_dependencyColoring" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_doDrawings" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_doGenerateScript" value="false"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_doResultSet" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_doWriteAnalysis" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_simulateApplications" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/prefs_ba_writeAnalysisShowDetailedCWandDP" value="true"/>
```

```
<option name="/com/sqlbrowser/sqlbm/last_server_opened" value="HPH8NGX_16SP02"/>
```

```
<option  
name="/com/sqlbrowser/sqlbm/servers/HPH8NGX_16SP02/successful_connections/00000"  
value="sa|true|83F5946D16C3D986DFF47D7BC1A9DAAF|true|"/>
```

```
</arguments>
```

- **Credentials:**
SQLBrowser needs the credentials to be passed encrypted. Currently, the only way to get the value of an encrypted

password is to login interactively with a login **which has the same password as the one of the login used for the batch analysis**. For instance, if you know that the password for the batch analysis login will be "123456", in order to get an encrypted password, you can do the following (provided you can create logins):

On any server, create a dummy login e.g.: **exec sp_addlogin 'fake', '123456'**.

Then login from SQLBrowser **with this login**. Then generate the arguments.xml through the Menu **Tools / Options / SQLBrowser / Export / Export as Arguments.xml**.

In this file, lookup for the entry matching the last successful param for this login **fake**. It must look like this:

```
<option name="/com/sqlbrowser/sqlbm/servers/A_157/successful_connections/00000"
value="fake/true/30632147070A8DD22A6E86B8F5C6D228|true|CHARSET=&lt;NULL&gt;[...more options...]" />
```

You can now take the encrypted part (in yellow) and place it in your arguments.xml

- create a command file to launch SQLBrowser
 1. With enough memory
 2. With optional -nosplash argument if you want to avoid the splash screen

```
C:\temp\sqlbrowser\bin\sqlbrowser_w.exe -nosplash -J-Xms64m -J-Xmx3g -J-
DargumentsFileName="C:/temp/sqlbrowser/arguments.xml"
```

- Exit code of the exe
 1. 0: no error
 2. 1: error
- open the Scheduled Task from the Configuration Panel and create a Windows Task using the command file
- schedule this task every day or every week, preferably when the database is idle.
The analysis typically takes about 5 minutes for 10,000 objects
- SQLBrowser launched in batch mode will still use all the options and preferences which are defined for the user which is used to run the task unless they are overridden by the arguments file.
- SQLBrowser will run the batch analysis and then exit

Memory Considerations

The Batch Analysis requires a fair amount of memory to run correctly. The memory parameters are set by the -Xms64m -Xmx400m command line parameters. The maximum size depends on the number of objects scanned.

If batch analysis takes too long to complete because of Memory Issues:

1. In the preferences, disable 'Table Drawings', 'Write Analysis' and 'Result Sets' in the preferences. This will consume less memory
2. Verify that the -J-Xmx parameter is high enough given the number of objects scanned. For a 32Bits Java version, maximum memory is around 1G
3. Use a 64 bits Java VM and pass a few GB as in -J-Xmx5G

"Applications" And Coloring

This is a way to illustrate either data propagation (for write analysis) or dependent objects through colored tags.

DEFINING APPLICATIONS

To give "Application" attributes to tables, tables columns or stored procs params, you need to create a couple of special tables with the following structure, in any database you like. For example:

```
use anydb
go
drop table sqlbrowser_application_object
go
drop table sqlbrowser_application -- in this order!
go
create table sqlbrowser_application (
```

```

    application varchar(255) not null,
    color varchar(255) not null,
    primary key ( application )
)
go
create table sqlbrowser_application_object (
    object varchar(255) not null,
    application varchar(255) not null references sqlbrowser_application( application ),
    primary key ( object )
)
go

```

The first table describes your 'applications' and attach a Color to them in Hex RGB format.

For examples of color codes, see [this link](#).

The second table represents the association between objects (tables or procs) and applications. You fill this table with fully qualified object names and with an application.

You will find these attributes appearing in the "Column Writes" page and in "schema.xml".

example:

```

use mydb
go
insert sqlbrowser_application values ( 'Application_1', '0xFF4f4f' )
insert sqlbrowser_application values ( 'Application_3', '0xFFFF952' )
...
go

```

or:

```

use sybsystemprocs
go
insert mydb.sqlbrowser_application_object
    select lower(db_name())+'..' +lower(name), 'Application_1'
    from sysobjects where type in ('P','TR') and name like '%ua_%'
go
...

```

You can also choose to get a random color per database through the Assign Random Color to each Database switch

EFFECTS IN WRITE ANALYSIS:

The Column Write analysis shows each write to a permanent table. When "Write Analysis" is turned on, the color of top level stored procs parameters or source columns will propagate into the columns of permanent tables recursively.

EFFECTS IN DEPENDENCY PROPAGATION:

When you click "Propagate Coloring Through Dependencies", each proc will get painted with the color of any object it touches

Best Practices

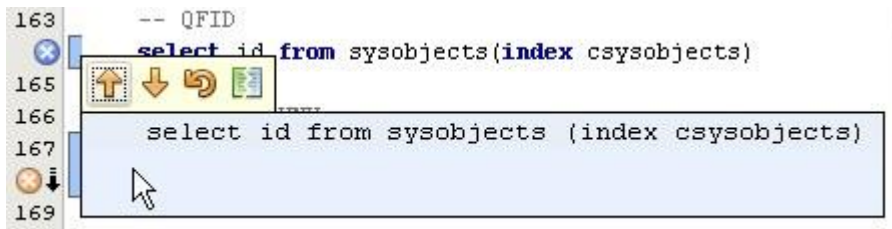
Store SQL Source files in Source Code Control

Many customers use Subversion to store SQL source files, therefore we recommend this choice naturally. SVN is now very well integrated in SQLBrowser and the 'source' option allows to work directly on the sources from the object chooser. This has the following productivity advantages:

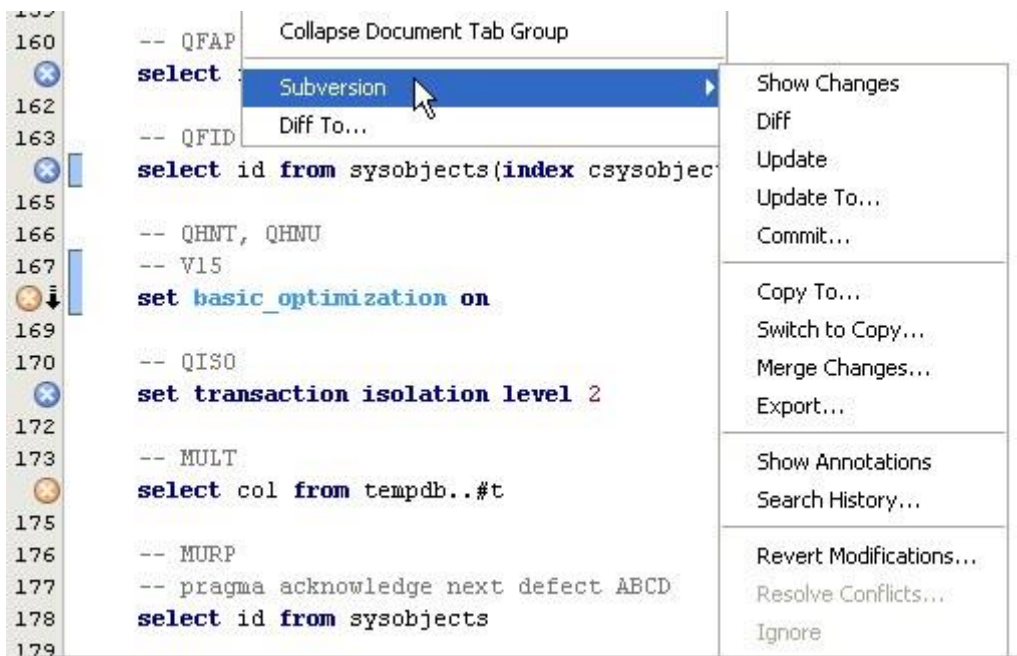
1. versioned directories are immediately recognized in the favorites



2. local changes are viewable from the left colored bars



3. check in, checkout, merge, diffs, annotations and search history are all available



Schedule Batch Analysis with results on a shared drive

Use a Windows Task or a crontab to schedule daily or weekly batch analysis. This will have the following advantages:

1. Users will bookmark these analysis pages and get the reflex of getting to them naturally
2. Dependencies and Schemas (Foreign Keys) will be available from every interactive SQLBrowser (Server Input Dir option)

Please follow the instructions [here](#) to set this up

Log File

The log file is located in the output directory.

It contains traces which help diagnosing issues with SQLBrowser.

It is accessible through the 'View / SQLBrowser Log File' Menu.

Please always attach this log to your enquiries.

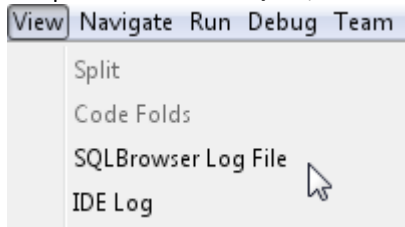
Please make sure you understand what are the main directories used by SQLBrowser first by looking at [SQLBrowser Directories](#).

1. If you need to report a bug or an issue:

1. Please always attach the **SQLBrowserLog.txt** file (see point 2)
2. If it relates to license, please also attach the **SQLBLicenseServerLog0.txt** located in the bin directory of the license server
3. Send the mail to support@sqlbrowser.com

2. If you have issues, the first thing to look at is the content of the SQLBrowser log file:

1. This file is named **SQLBrowserLog.txt**
2. Open it via the **View / SQLBrowser Log File** Menu



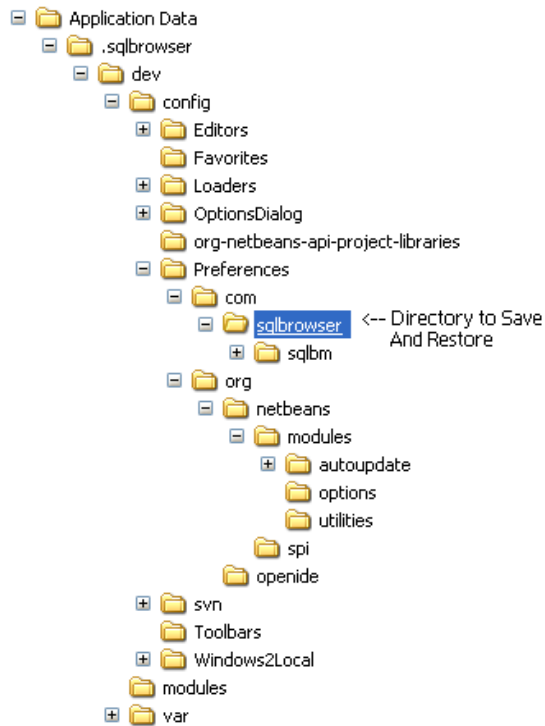
3. If SQLBrowser hangs at startup or refuses to start or has certain Bug Notifications in the lower right of the screen:

1. Shutdown SQLBrowser
 - A. If needed, kill the process with the task manager
 - B. Check that the file named **lock** in the **SQLBrowserUserDir** is absent: **Remove it if it is present.**
If it can't be deleted, it means that a process is still holding a lock to it, thus that SQLBrowser is still running.
Kill this process before removing the file
2. Remove entirely the **var** directory in the **SQLBrowserUserDir**
3. Restart SQLBrowser

4. After an upgrade, it may happen that some persisted classes are no longer compatible with the new version.
You may find that the memory grows and you may find Exceptions in the IDE LOG (Accessible from View Menu)

In such case please do the following (Under Windows):

- A. Kill the process with the task manager if needed
 - B. Check that the file named **lock** in the **SQLBrowserUserDir** is absent. **Remove it if it is present.**
Make sure that the Windows Explorer flag which allows to see Hidden files is turned on (File or Folder Options Menu)
2. Locate the **"user directory"** by looking it up as described in [Directories](#)
 3. Copy this directory to a safe place or rename it e.g. rename it to ".sqlbrowserOLD"
 4. Delete the original directory after you have copied it to a safe place or renamed
 5. Launch SQLBrowser (SQLBrowser should start clean)
 6. Shutdown SQLBrowser
 7. Check that SQLBrowser has recreated the "user directory" content with default values
 8. Restore the interesting preferences directory from the saved directory:
 - A. copy the entire sub directory: **"config\Preferences\com\sqlbrowser"** (see image below) into the new "user directory" at the same relative location
 9. Launch SQLBrowser (SQLBrowser should now restore your options)



5. if windows don't show up properly, it may be due to the persistence mechanism trying to restore incompatible versions of the stored windows configuration.

To solve the issue, do either of the following in this order:

Click on the '**Window/Reset Windows**' menu item

1. Quit SQLBrowser
2. go to the 'user directory' (see [SQLBrowser Directories](#)), and delete the **config/Windows2Local** directory
3. If the windows which does not show is the **Options Window** for instance, go to the 'user directory' and delete the **config\Preferences\org\netbeans\modules\options** directory.

The options panel coordinates are stored in this file rather than in the Windows2Local directory mentioned in the point before

6. If the program is slow and saying '**Checking for External Changes**'

1. This means that either favorites directories or open files are being seek for external change in order to reflect the changes
2. Make sure you open files from a local drive and not from a remote directory
3. Make sure your favorites are also from the local drive
4. Check whether your version control (e.g. SVN) is fast enough
5. If this is still not working , go in Preferences / Misc and untick Enable auto-scanning of sources

7. In general, removing the **<userdir>/var** directory (when sqlbrowser is not running) is never really harmful and can help clearing certain situations

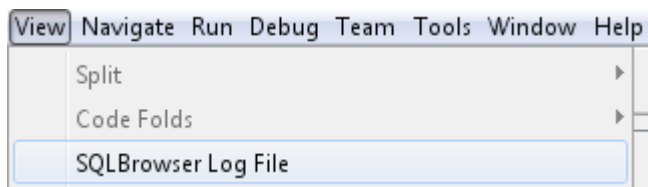
Reporting A Problem

If your problem persists, send an email to support@sqlbrowser.com and always attach the following files:

File # 1: The SQLBrowser client log file:

SQLBrowserlog.txt

You can view this file by using:



File #2: If the issue lies with the License Server, also send the SQLBrowser server log file:

SQLBLicenseServerLog0.txt

This file is in the bin directory of the License server

It is practically impossible to resolve issues without these trace files. These files are technical logs of System Properties and SQLBrowser log traces and do not contain confidential information.

Using SQLBrowser connecting to Production Sybase Servers

It is not recommended to use SQLBrowser on Production Servers because many Analysis Tasks and Development tasks can be performed as well on Dev servers.

However, if one wants to use SQLBrowser against production servers, you can use server colors to paint windows background in light RED for instance, so that it becomes explicit that one is connected to a Production Server.

If your policy requires that such tools are forbidden in production, create a login trigger which rejects connection which are named **SQLBrowser**.

Known Bugs And Limitations

- SQLBrowser works on Windows. SQLBrowser could work on Unix, Linux or Mac but is not tested on these platforms
- SQLBrowser works on Sybase ASE version 12.5 or higher
- SQLBrowser may not parse certain exotic statements. As a consequence, do not use the DDL extracts for re-creation of database objects as it may miss some attributes
- Because procs and other objects are managed by their object ID, When a proc is resubmitted, certain operations such as Extracting or Debugging require a reload of the main connection (left hand side connection panel)
- Debugging a proc which declares a cursor named c and which calls a proc which declares the same cursor name may fail
- Debugging is done through sending executables as individual batches with all variables expanded with their literal values.

This has certain limitations , for instance the following construct will not work:

```
declare main_cursor cursor for select * from #t1
declare sub_cursor cursor for select * from #t2 where c2 = @c1 -- ! deferred eval!
open main_cursor
fetch main_cursor into @c1 -- ! @c1 is only known here!
open sub_cursor
```

- The Metal look-and-feel does not behave properly in certain places (e.g. Server Dropdown actions): it is recommended to stay with the Windows L&F
- When a table has a column whose name is a reserved word, then debugging a proc containing a select * from this table is impossible as the Sybase automatic expansion of '*' will have not quoted the reserved word. E.g.

```
create proc p as
create table #t (a int, [new] int)
insert #t values (1,2)
select * from #t -- will expand to "select #t.a, #t.new from #t"
```

Contact

Please contact us only through emails :

- For support please send a mail to support@sqlbrowser.com
- For quotes please send a mail to sales@sqlbrowser.com

Regular Expressions Help

[Regular Expressions Help](#)

List of Detected SQL Defects

[List of Detected SQL Defects](#)