CHT ListBoxBrowseExtender New, 2017 Features

An article about ListBoxBrowseExtender from which this one carries on, was originally written early in 2016. With this writing, we expand and update that document to reflect new template features added since the original article.

While introducing SQL-VIEWs via the HNDSCHOOL.APP and SQLite feature study, we began once again - *since we've spoken on this before* - to notice how much of making ABC browse stuff work in new and more efficient ways -- *especially using DATABASE VIEWS directly* -- is an exercise in ABC Browse workarounds.

The Client Server paradigm as it exists already in the SQL world, is the simplest of all models for displaying and browsing data, yet with the ABC Browse Template we tend to make it into something way more complex than it actually is.

In short, the client server data paradigm is as follows: "Send a query to the data base and get back matching data". No query, no data. Simple.

There are a few corollaries to this, and all of them revolve around having the data base do as much of the work *(eg: filtering, field formatting, field qualifying, and so on)* as possible.

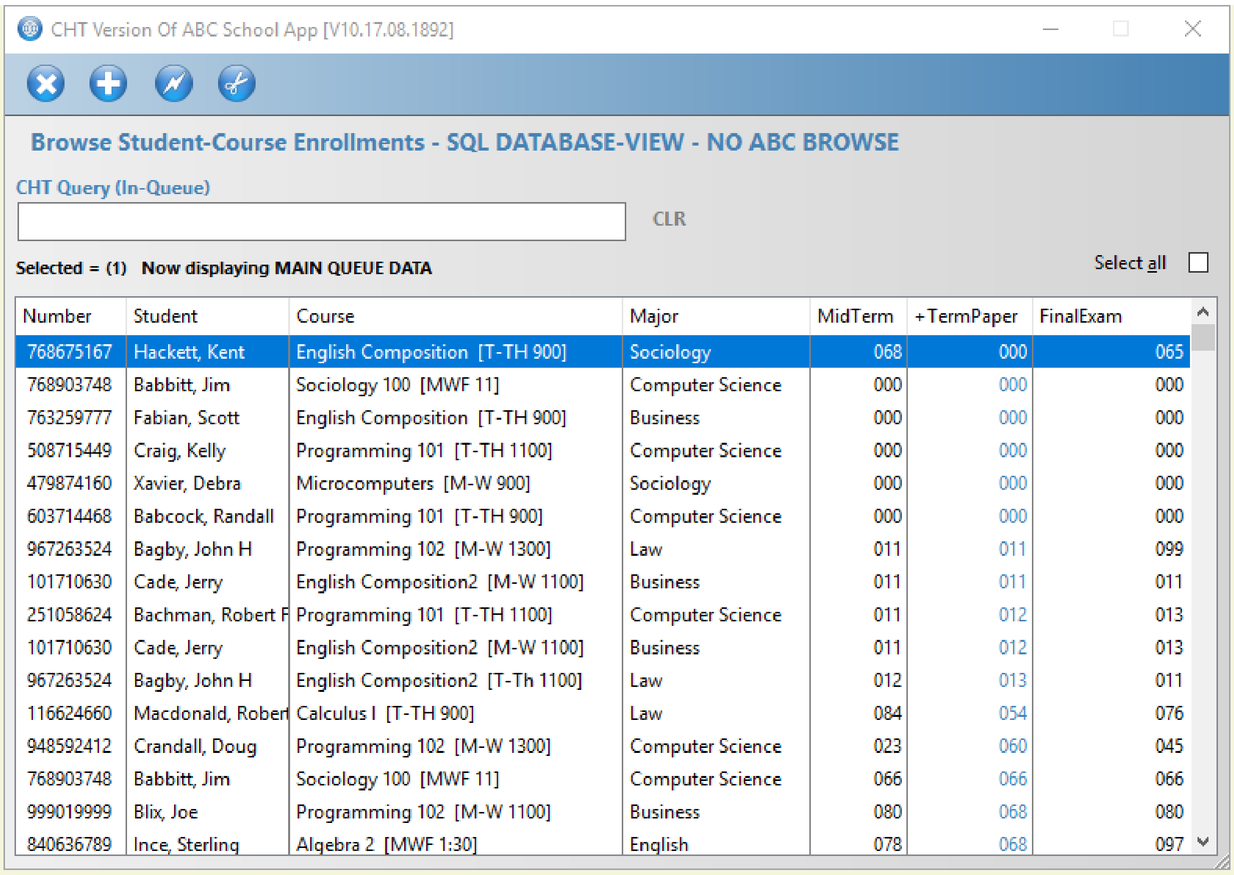
This SQL client server model sets you up to insert the internet, WAN or network between your app and your data tables and not see a lot of degredation in performance.

That bit of background re-introduces for 2017, our discussion of CHT *ListBoxBrowseExtender*.

This template is a *HandyMarkerBrowse* look-and-act-alike, but it does *not need an ABC browse template underneath* it to fill the list box queue. It just needs a Clarion List Control and some files or data views to build the browse. Since our last version of this article, we've added a procedure to HNDSCHOOL.APP that builds a browse on an SQL view that we defy you to distinguish from any other ABC based browse like *HandyMarkerBrowse*.

The next image pictures just such a *ListBoxBrowseExtender* based browse:

SQL DATABASE VIEW - NO ABC BROWSE



Browse Student-Course Enrollments - SQL DATABASE-VIEW - NO ABC BROWSE

CHT Query (In-Queue) CLR

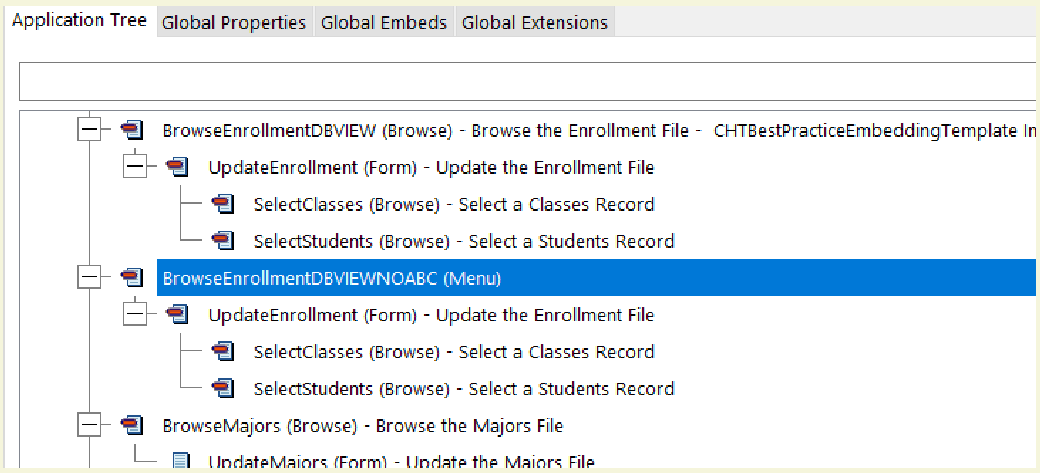
Selected = (1) Now displaying MAIN QUEUE DATA

Select all

Number	Student	Course	Major	MidTerm	+TermPaper	FinalExam
768675167	Hackett, Kent	English Composition [T-TH 900]	Sociology	068	000	065
768903748	Babbitt, Jim	Sociology 100 [MWF 11]	Computer Science	000	000	000
763259777	Fabian, Scott	English Composition [T-TH 900]	Business	000	000	000
508715449	Craig, Kelly	Programming 101 [T-TH 1100]	Computer Science	000	000	000
479874160	Xavier, Debra	Microcomputers [M-W 900]	Sociology	000	000	000
603714468	Babcock, Randall	Programming 101 [T-TH 900]	Computer Science	000	000	000
967263524	Bagby, John H	Programming 102 [M-W 1300]	Law	011	011	099
101710630	Cade, Jerry	English Composition2 [M-W 1100]	Business	011	011	011
251058624	Bachman, Robert F	Programming 101 [T-TH 1100]	Computer Science	011	012	013
101710630	Cade, Jerry	English Composition2 [M-W 1100]	Business	011	012	013
967263524	Bagby, John H	English Composition2 [T-Th 1100]	Law	012	013	011
116624660	Macdonald, Robert	Calculus I [T-TH 900]	Law	084	054	076
948592412	Crandall, Doug	Programming 102 [M-W 1300]	Computer Science	023	060	045
768903748	Babbitt, Jim	Sociology 100 [MWF 11]	Computer Science	066	066	066
999019999	Blix, Joe	Programming 102 [M-W 1100]	Business	080	068	080
840636789	Ince, Sterling	Algebra 2 [MWF 1:30]	English	078	068	097

Here is the HNDSCHOOL.APP procedure we've built to show you how it's done:

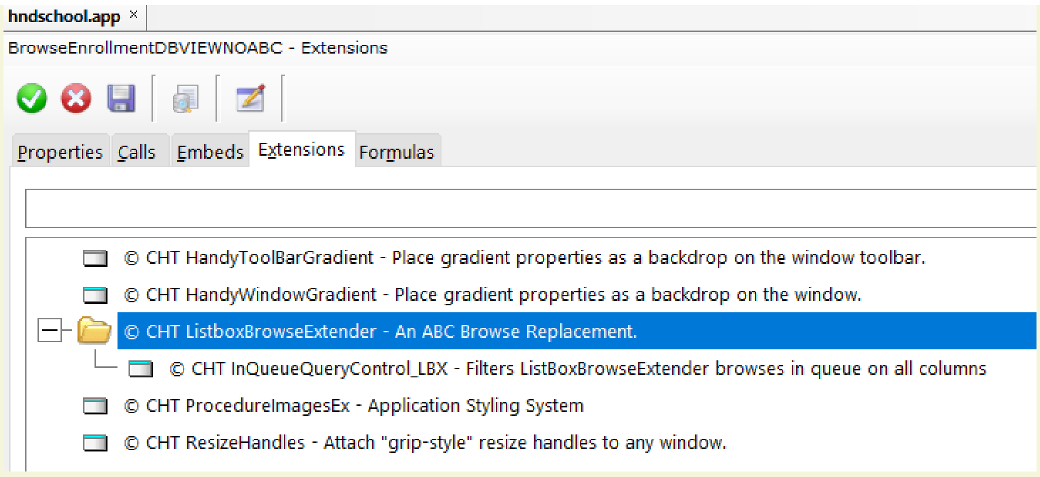
HNDSCHOOL.APP PROCEDURE TREE



```
BrowseEnrollmentDBVIEW (Browse) - Browse the Enrollment File -  CHTBestPracticeEmbeddingTemplate In
    UpdateEnrollment (Form) - Update the Enrollment File
        SelectClasses (Browse) - Select a Classes Record
        SelectStudents (Browse) - Select a Students Record
BrowseEnrollmentDBVIEWNOABC (Menu)
    UpdateEnrollment (Form) - Update the Enrollment File
        SelectClasses (Browse) - Select a Classes Record
        SelectStudents (Browse) - Select a Students Record
BrowseMajors (Browse) - Browse the Majors File
    UpdateMajors (Form) - Update the Majors File
```

And here's the extension template list for that procedure. Note, no ABC Browse!

HNDSCHOOL.APP EXTENSION TEMPLATE LIST


hndschool.app ×

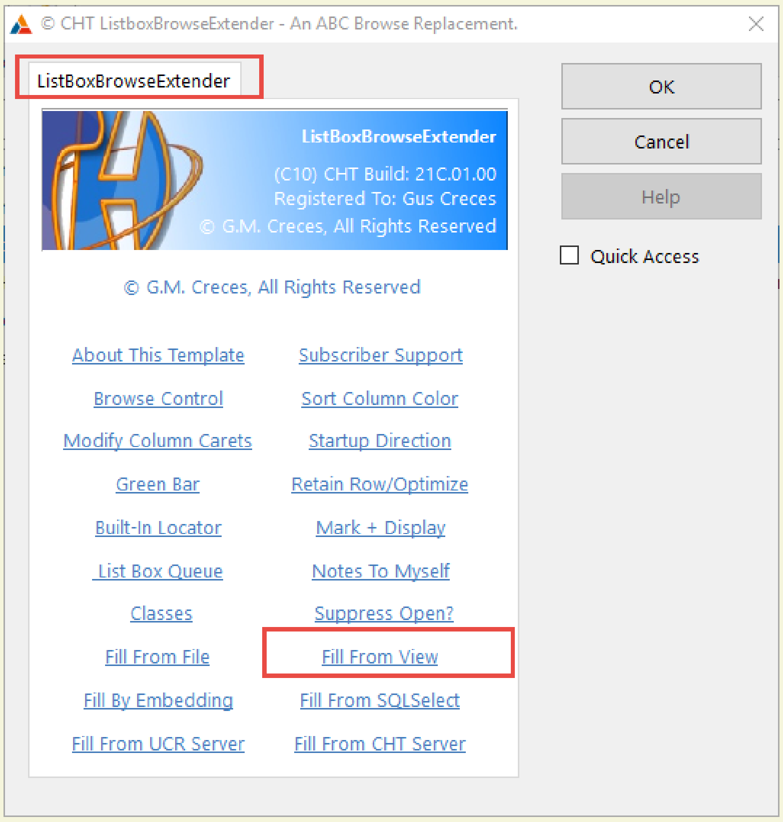
BrowseEnrollmentDBVIEWNOABC - Extensions

Properties | Calls | Embeds | Extensions | Formulas

© CHT HandyToolBarGradient - Place gradient properties as a backdrop on the window toolbar.
© CHT HandyWindowGradient - Place gradient properties as a backdrop on the window.
© CHT ListboxBrowseExtender - An ABC Browse Replacement.
© CHT InQueueQueryControl_LBX - Filters ListBoxBrowseExtender browses in queue on all columns
© CHT ProcedureImagesEx - Application Styling System
© CHT ResizeHandles - Attach "grip-style" resize handles to any window.

The next group of images are *ListBoxBrowseExtender* configuration dialogs. They assume that a vanilla Clarion List Control has been dropped on the window which has fields (columns) populated in it in the same way that you populate fields (columns) to an ABC browse template.

LISBOXBROWSEEXTENDER (LBX) MAIN DIALOG


© CHT ListboxBrowseExtender - An ABC Browse Replacement.

ListBoxBrowseExtender

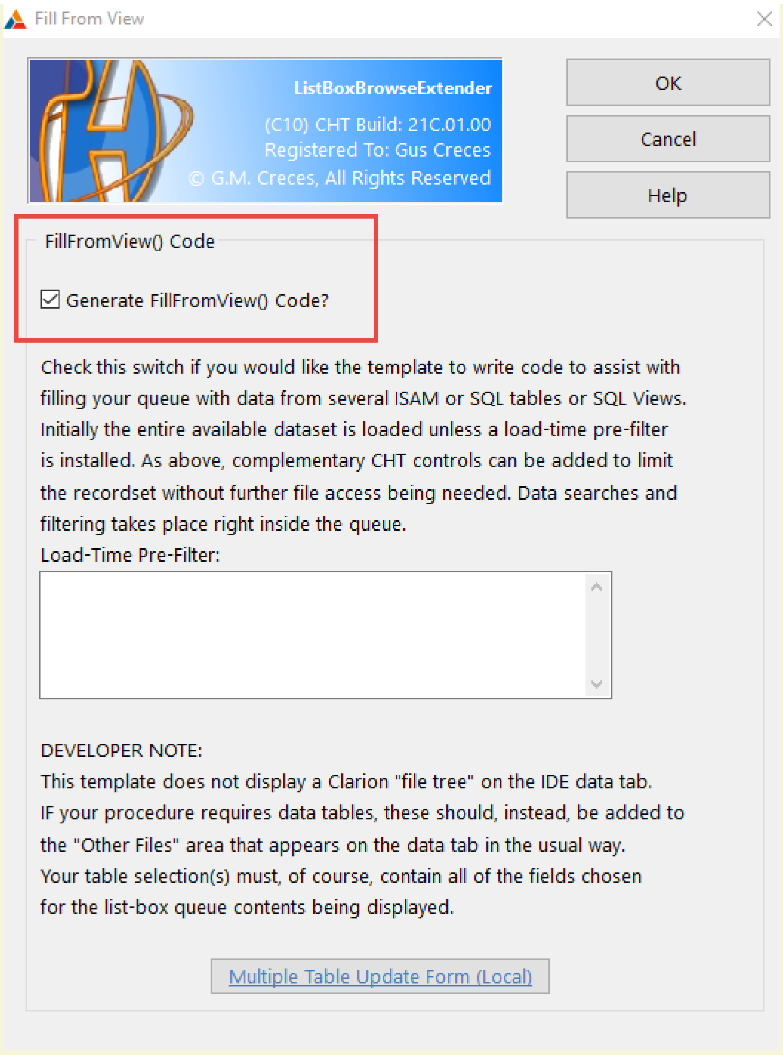
ListBoxBrowseExtender
(C10) CHT Build: 21C.01.00
Registered To: Gus Creces
© G.M. Creces, All Rights Reserved

© G.M. Creces, All Rights Reserved

About This Template | Subscriber Support
Browse Control | Sort Column Color
Modify Column Carets | Startup Direction
Green Bar | Retain Row/Optimize
Built-In Locator | Mark + Display
List Box Queue | Notes To Myself
Classes | Suppress Open?
Fill From File | Fill From View
Fill By Embedding | Fill From SQLSelect
Fill From UCR Server | Fill From CHT Server

OK
Cancel
Help
☐ Quick Access

LBX FILL FROM VIEW DIALOG


Fill From View

ListBoxBrowseExtender
(C10) CHT Build: 21C.01.00
Registered To: Gus Creces
© G.M. Creces, All Rights Reserved

FillFromView() Code

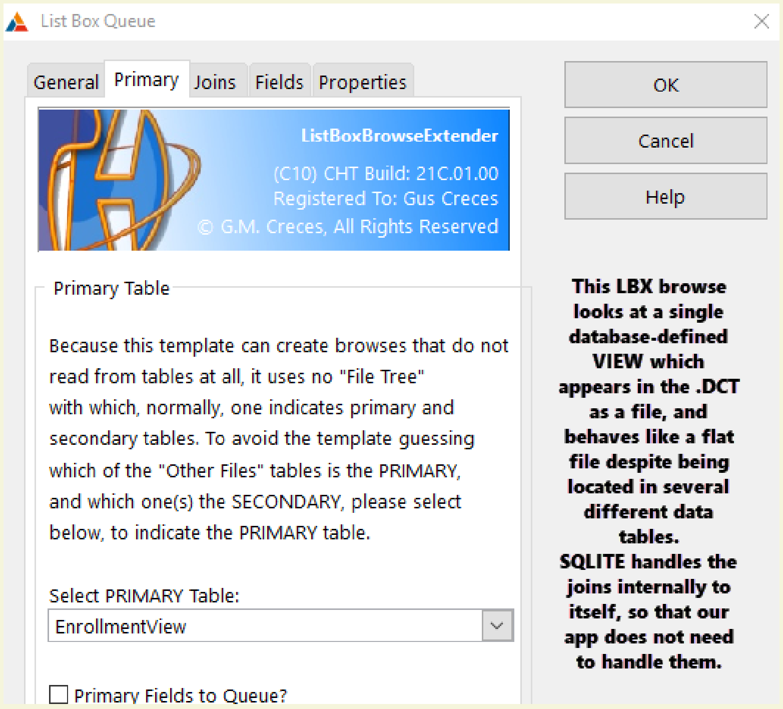
☑ Generate FillFromView() Code?

Check this switch if you would like the template to write code to assist with filling your queue with data from several ISAM or SQL tables or SQL Views. Initially the entire available dataset is loaded unless a load-time pre-filter is installed. As above, complementary CHT controls can be added to limit the recordset without further file access being needed. Data searches and filtering takes place right inside the queue.

Load-Time Pre-Filter:

DEVELOPER NOTE:
This template does not display a Clarion "file tree" on the IDE data tab.
IF your procedure requires data tables, these should, instead, be added to the "Other Files" area that appears on the data tab in the usual way.
Your table selection(s) must, of course, contain all of the fields chosen for the list-box queue contents being displayed.

LBX PRIMARY TABLE DIALOG


List Box Queue

ListBoxBrowseExtender
(C10) CHT Build: 21C.01.00
Registered To: Gus Creces
© G.M. Creces, All Rights Reserved

Primary Table

Because this template can create browses that do not read from tables at all, it uses no "File Tree" with which, normally, one indicates primary and secondary tables. To avoid the template guessing which of the "Other Files" tables is the PRIMARY, and which one(s) the SECONDARY, please select below, to indicate the PRIMARY table.

Select PRIMARY Table: EnrollmentView

☐ Primary Fields to Queue?

This LBX browse looks at a single database-defined VIEW which appears in the .DCT as a file, and behaves like a flat file despite being located in several different data tables. SQLITE handles the joins internally to itself, so that our app does not need to handle them.

LBX TABLE FIELDS DIALOG

sqlnoabcbrowse06.png

ENROLLMENTVIEW DICTIONARY DEFINITION

sqlnoabcbrowse08.png

ENROLLMENTVIEW CREATE-VIEW CODE INSIDE HNDSCHOOL.SQLITE

By examining the SQL view definition below, extracted from HNDSCHOOL.SQLITE using SQLITESTUDIO.EXE located in `\accessory\hnd\sqlite\sqlitestudio\`, we are able to show that the table we see from Clarion and define in the HNDSCHOOL.DCT as "EnrollmentView" consists, in actuality, of fields from 5 different data tables.

Without any effort on our part in the way of coding inside our HNDSCHOOL.APP we are able to treat the fields in these 5 tables, enumerated in "EnrollmentView" as if they were located in a single table. The SQLITE data base handles all of the joining work for us.

The data base definition of EnrollmentView even handles several concatenations with the "STUDENTNAME" field providing a "LastName, FirstName" concatenation and the "COURSEDESCRIPTION" field providing a "Description [ScheduleTime]" concatenation.

All without us needing to write any Clarion code to maintain the joins and to perform the concatenations.

Look at the first image provided with this article depicting a Clarion-style browse entitled "SQL DATABASE VIEW - NO ABC BROWSE".

That is a browse from HNDSCHOOL.APP built with LBX, configured with its "FillFromView()" setting, and which requires NO EXTRA EMBED CODE.

```
CREATE VIEW enrollmentview AS SELECT

                         Enrollment Table

Enrollment.ID AS ENRID,
Enrollment.StudentNumber AS STUDENTNUMBER,
Enrollment.ClaNumber AS CLANUMBER,
Enrollment.MidtermExam AS MIDTERMEXAM,
Enrollment.FinalExam AS FINALEXAM,
Enrollment.TermPaper AS TERMPAPER,

                         Classes Table

Classes.ClassNumber AS CLASSNUMBER,
Classes.CourseNumber AS COURSENUMBER,
Classes.ScheduledTime AS SCHEDULEDTIME,

                         Students Table

Students.StuNumber AS STUNUMBER,
Students.FirstName AS FIRSTNAME,
Students.LastName AS LASTNAME,
Students.Major AS MAJOR,
trim(Students.LastName) || ', ' ||
trim(Students.FirstName) AS STUDENTNAME

                         Majors Table

Majors.MajNumber AS MAJNUMBER,
Majors.MajDescription AS MAJDESCRIPTION,

                         Courses Table

Courses.CouNumber AS COUNUMBER,
Courses.Description AS DESCRIPTION,
trim(Courses.Description) || ' [' ||
trim(Classes.ScheduledTime) || ']' AS COURSEDESCRIPTION,

                    Table and Join Definition

FROM Enrollment, Classes, Students, Majors, Courses WHERE
Classes.ClassNumber = Enrollment.ClaNumber AND
Students.StuNumber = Enrollment.StudentNumber AND
Majors.MajNumber = Students.Major AND
Courses.CouNumber = Classes.CourseNumber
```

ENROLLMENTVIEW CREATE-VIEW CODE From Inside HNDSCHOOL.APP

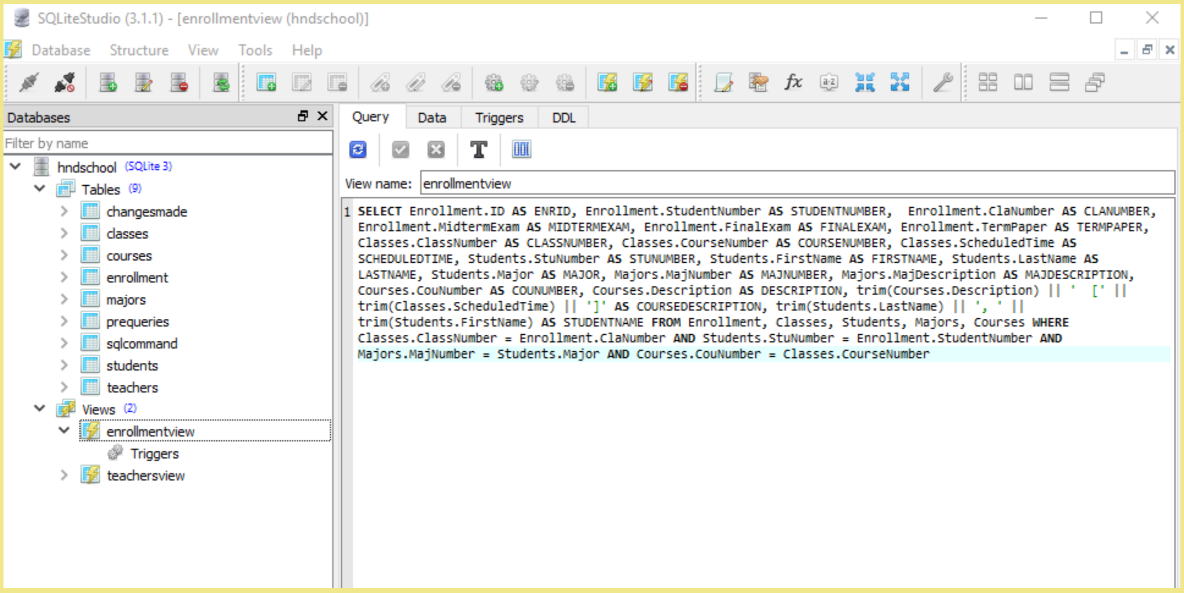
How did the definition of "EnrollmentView" get into the HNDSCHOOL.SQLITE data base, we hear you ask?

How does any data table or data view definition get into an SQL data base? You write a "CREATE" SQL statement and execute it once, in the data base. A creation string of this sort can be typed in any editor and executed in the data base using a utility like SQLLITESTUDIO.EXE.

Some very clever developers at SQLITE STUDIO provide a donation-funded database table maintenance and browse utility which you can find packaged in your CHT installation in directory `\accessory\hnd\sqlite\sqlitestudio\`*. The executable is called SQLITESTUDIO.EXE.*

We did not create this app but we use it and recommend that you do the same. Our inclusion of this utility will help to make your SQLITE experience as easy and rich as possible. Thank the developers of SQLITESTUDIO.EXE, and if possible, make your donation, for their hard work and for providing a very useful SQLITE support tool that easily rivals SV's TOPSCAN.EXE utility!

SQLITESTUDIO.EXE



View name: enrollmentview

```
SELECT Enrollment.ID AS ENRID, Enrollment.StudentNumber AS STUDENTNUMBER,  Enrollment.ClaNumber AS CLANUMBER,
Enrollment.MidtermExam AS MIDTERMEXAM, Enrollment.FinalExam AS FINALEXAM, Enrollment.TermPaper AS TERMPAPER,
Classes.ClassNumber AS CLASSNUMBER, Classes.CourseNumber AS COURSENUMBER, Classes.ScheduledTime AS
SCHEDULEDTIME, Students.StuNumber AS STUNUMBER, Students.FirstName AS FIRSTNAME, Students.LastName AS
LASTNAME, Students.Major AS MAJOR, Majors.MajNumber AS MAJNUMBER, Majors.MajDescription AS MAJDESCRIPTION,
Courses.CouNumber AS COUNUMBER, Courses.Description AS DESCRIPTION, trim(Courses.Description) || '  [' ||
trim(Classes.ScheduledTime) || ']' AS COURSEDESCRIPTION, trim(Students.LastName) || ', ' ||
trim(Students.FirstName) AS STUDENTNAME FROM Enrollment, Classes, Students, Majors, Courses WHERE
Classes.ClassNumber = Enrollment.ClaNumber AND Students.StuNumber = Enrollment.StudentNumber AND
Majors.MajNumber = Students.Major AND Courses.CouNumber = Classes.CourseNumber
```

SQL INJECTION - WITH CLARION "SEND()"

Another way is to create the string using the Clarion editor and then "*injecting*" it into the data base using the Clarion "SEND()" command as shown below.

SQL Injection with SEND() is how we did it in HNDSCHOOL.APP. And that code was executed a sum-total of *once only* in order to create the "EnrollmentView" definition as you see it here.

You can see this *injected* code in our HNDSCHOOL.APP example by opening a procedure called "*CreateDB()*".

Here is the code that creates the "EnrollmentView" via injection of our SQL CREATE VIEW code into the SQLLITE data base.

Once created, the view thus created handles all view-related back-end work for us, exactly as described in this article.

```
!CREATE ENROLLMENTVIEW STRUCTURE IN THE DATA BASE
CLEAR(SQlCommandB)
```

```
SQLCommandB = |
'CREATE VIEW enrollmentview AS SELECT ' & |
```

Enrollment Table

```
'Enrollment.ID AS ENRID, ' & |
'Enrollment.StudentNumber AS STUDENTNUMBER, ' & |
'Enrollment.ClaNumber AS CLANUMBER, ' & |
'Enrollment.MidtermExam AS MIDTERMEXAM, ' & |
'Enrollment.FinalExam AS FINALEXAM, ' & |
'Enrollment.TermPaper AS TERMPAPER, ' & |
```

Classes Table

```
'Classes.ClassNumber AS CLASSNUMBER, ' & |
'Classes.CourseNumber AS COURSENUMBER, ' & |
'Classes.ScheduledTime AS SCHEDULEDTIME, ' & |
```

Students Table

```
'Students.StuNumber AS STUNUMBER, ' & |
'Students.FirstName AS FIRSTNAME, ' & |
'Students.LastName AS LASTNAME, ' & |
'Students.Major AS MAJOR, ' & |
'trim(Students.LastName) || <39>,<32><39> || ' & |
'trim(Students.FirstName) AS STUDENTNAME ' & |
```

Majors Table

```
'Majors.MajNumber AS MAJNUMBER, ' & |
'Majors.MajDescription AS MAJDESCRIPTION, ' & |
```

Courses Table

```
'Courses.CouNumber AS COUNUMBER, ' & |
'Courses.Description AS DESCRIPTION, ' & |
'trim(Courses.Description) || <39><32><32>[<39> || ' & |
'trim(Classes.ScheduledTime) || <39>]<39> AS COURSEDESCRIPTION, ' & |
```

Table and Join Definition

```
'FROM Enrollment, Classes, Students, Majors, Courses ' & |
'WHERE Classes.ClassNumber = Enrollment.ClaNumber AND ' & |
'Students.StuNumber = Enrollment.StudentNumber AND ' & |
'Majors.MajNumber = Students.Major AND ' & |
'Courses.CouNumber = Classes.CourseNumber '
```

Inject the VIEW into SQLLITE Data Base

```
SEND(SQLCommand,SQLCommandB)
IF ERRORCODE() THEN
  MESSAGE('ERROR CREATING EnrollmentVIEW ' & ERROR(),|
  'Error...',ICON:Asterisk)
END
```

FUTURE LBX ARTICLES

In an upcoming ListBoxBrowseExtender article we'll review further the use of DATA VIEWS as they apply in CHT Client Server applications providing data to clients located across the NETWORK, the WAN, and the INTERNET.

That discussion will revolve around another LBX example application already in your tool kits, called HNDPEOPLE_LBX.APP.