

Displaying of database files for IBM i

User Guide

Contents

Introduction	3
Using the application	4
<i>Objects used in the application</i>	<i>4</i>
<i>Directories</i>	<i>4</i>
<i>Location of the application.....</i>	<i>4</i>
<i>Start of the application.....</i>	<i>5</i>
Parameters	6
<i>Application language</i>	<i>6</i>
<i>Server address</i>	<i>6</i>
<i>User name.....</i>	<i>6</i>
<i>Size of the window displaying the database file</i>	<i>6</i>
<i>Mark for null field values.....</i>	<i>6</i>
<i>Size of the font to display data</i>	<i>6</i>
<i>Maximum number of records to display</i>	<i>6</i>
<i>Limit for length of displayed data fields</i>	<i>7</i>
<i>Library with database files.....</i>	<i>7</i>
<i>Select database file</i>	<i>7</i>
<i>File member</i>	<i>7</i>
<i>Connect / check.....</i>	<i>7</i>
Run	8
Selection and ordering of records	9
Seeing details of a record	11
Selection of columns.....	12
<i>Example</i>	<i>13</i>
Data members of physical and logical files	15
<i>Alias objects</i>	<i>15</i>
Data types CLOB and BLOB	16
<i>Displaying a CLOB column</i>	<i>17</i>
<i>Finding text by a pattern.....</i>	<i>18</i>
<i>Saving column value to a file.....</i>	<i>18</i>
<i>Displaying a BLOB column.....</i>	<i>19</i>

Introduction

Motivation for this application – *IBMiSqlDisplay* – was the fact that the popular utility DFU (Data File Utility) is unable to display or print all Unicode characters, especially UTF-8, UTF-16 or UCS-2.

The application, in contrast to the application *IBMiSqlUpdate*, enables to only display data of database files (tables). User interface was taken from the application *IBMiSqlUpdate* and was adequately limited. It also works with physical files (tables) and logical files (views). Only one file can be processed at a time.

Physical file should contain at least one member. Reference file without a data member displays only its header with field (column) names.

Logical file that refers to a single physical file can also be used to enter and update data.

A file (physical or logical) with multiple data members can be processed using ALIAS type objects corresponding to the members.

Traditional IBM i terms are mainly used in the application. Correspondence of IBM i and SQL terms are listed in the following table.

<i>Traditional names</i>	<i>SQL names</i>
library	schema
physical file	table
field	column
record	row
unique key	primary key unique key
key	index
logical file with selection	view
DDM file for a member	ALIAS object

Programs are written in Java language and require version *Java SE 8* or higher. They cooperate with programs from package *IBM i Toolbox for Java* (or *JTOpen*). Programs were created in the operating system OS X 10.9 (Mavericks). They were subsequently tested in operating systems OS X up to *macOS 26* (Tahoe) and *Windows 7* up to 11 with remote internet connection to the IBM i, versions 7.1 up to 7.5.

Using the application

The application is not installed, it is delivered as a *directory* that can be placed at a suitable location in the computer (maybe in a flash disk).

The same copy of the application works the same in systems macOS and Windows.

Objects used in the application

Objects that are contained in the application are placed in the current directory. This is ensured by delivering all the application as a directory (folder). When the application is launched the directory becomes current.

Directories

- *columnfiles* - contains text files with lists of fields (columns) for the SELECT statement,
- *helpfiles* - contains this document in Czech and English languages,
- *logfiles* - contains text files *err.txt* and *out.txt*, to which redirected output of the files System.err and System.out (i.e. from the console),
- *paramfiles* - contains the file *U_Parameters.txt* with application parameters,
- *selectfiles* - contains text files with data for selection and ordering of records in database files.
- File *jt400-21.0.4.jar* - contains classes from the package IBM i Toolbox for Java.
- File ***IBMiSqlDisplay.jar*** - contains Java classes of the application and *launches the application*.

Files *err.txt* and *out.txt* in the directory *logfiles* serve to find the cause of an error in program.

The directories *selectfiles* and *columnfiles* can be deleted without damage, they are created if needed. Contents of these files should not be changed manually.

Location of the application

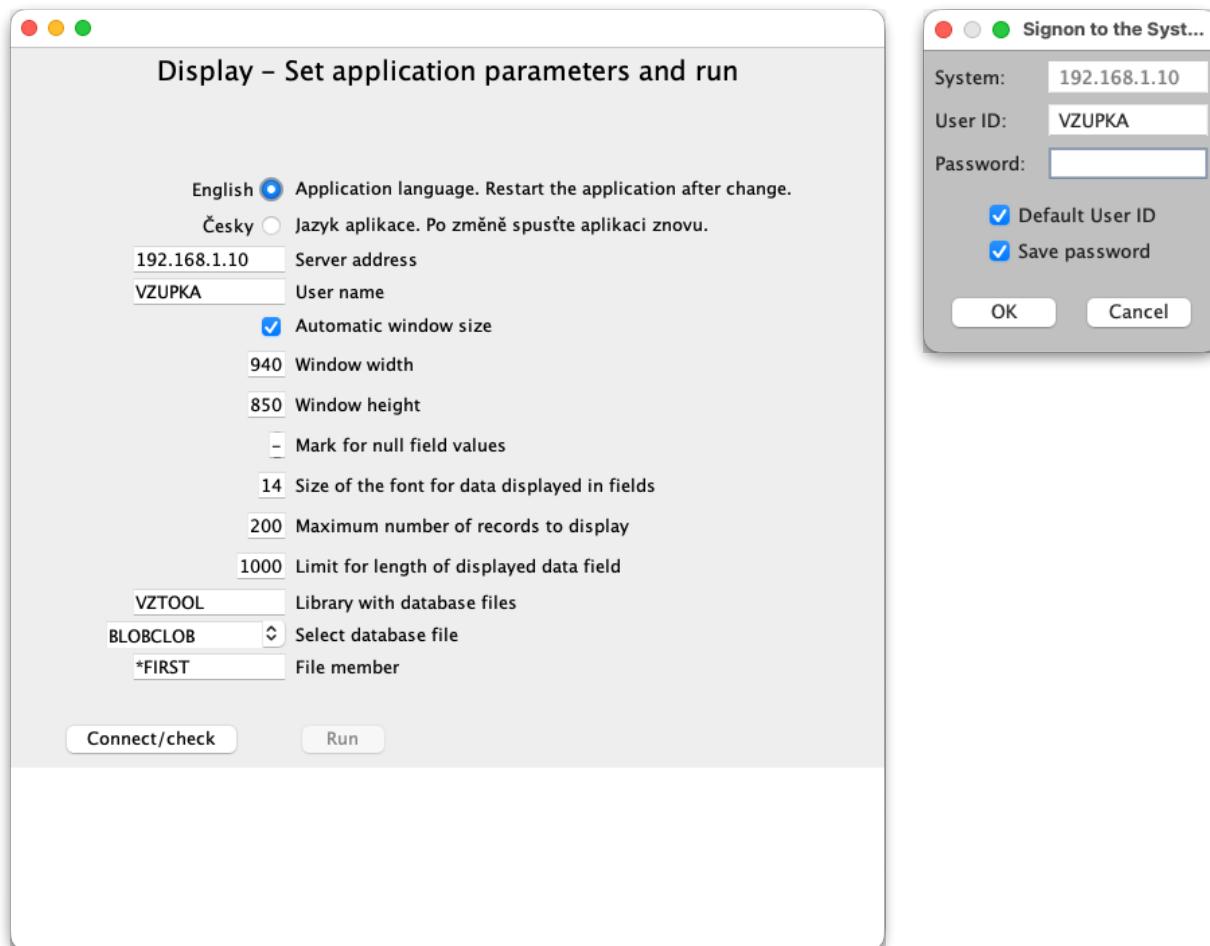
The application directory can be placed anywhere and possibly renamed. There we find the file ***IBMiSqlDisplay.jar*** of which shortcuts (aliases) can be created and placed on desktop or somewhere else.

Start of the application

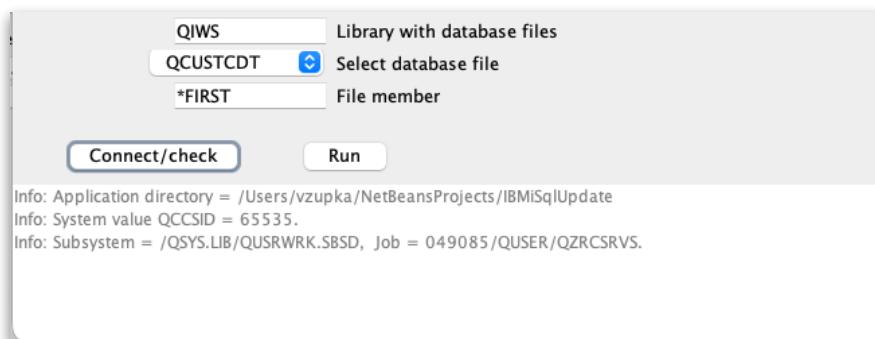
The application starts by *double click* on a shortcut (or the original) file *IBMiSqlUpdate.jar*.

The window *Set application parameters and run* is displayed. Default parameters are contained in the input fields. The user can change them. Only *Select database file* cannot be used until the application is connected to the server.

After *Server address* and *User name* is set, the user presses the button *Connect/check* and *signon dialog* for entering password to access to the IBM i system appears.

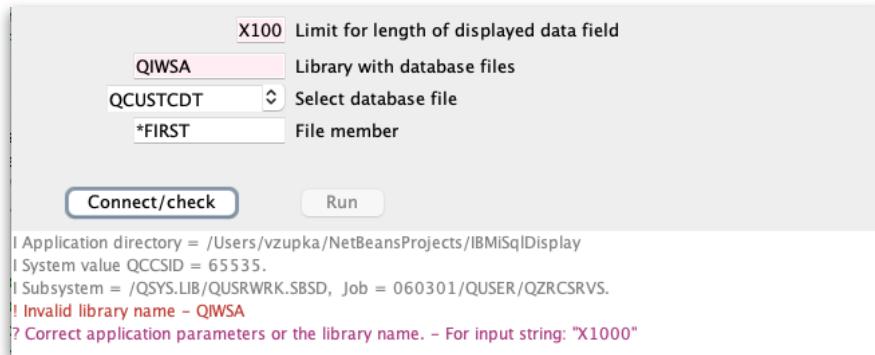


After successful signon, the user can still adjust parameter values, including *Library with database files* and *Select database file*, save them using the *Connect/check*.



Parameters

Note: If the user enters some important parameters incorrectly, then after attempt of connecting to the server or starting the application, messages are reported in the lower part of the window.



Application language

The application can be processed in English (en_US) or Czech (cs_CZ) localization. The localization concerns titles, messages, button labels. The user can choose the localization by clicking the button. The selected option is applied only after ending the application and launching it again.

Server address

An IP address in dot form or domain form. This address is copied into the dialog window *Signon to the System*.

User name

Name of the profile that has the authority to write and change data in database files. This name is copied into the dialog window *Signon to the System*.

Size of the window displaying the database file

If the box *Automatic window size* is checked, the window is accommodated to the size of the displayed results. Otherwise the window will have dimensions specified in input fields *Window width* and *Window height* with possible sliders.

Mark for null field values

The user enters a text (of arbitrary length) that will be shown everywhere the field value is NULL. It is *not used* for large objects CLOB and BLOB.

Size of the font to display data

This entry represents the number of points on the display. It determines size of letters in titles and cells in table cells.

Maximum number of records to display

This entry represents maximum number of records that are selected from the database file and will be shown in the window table.

Limit for length of displayed data fields

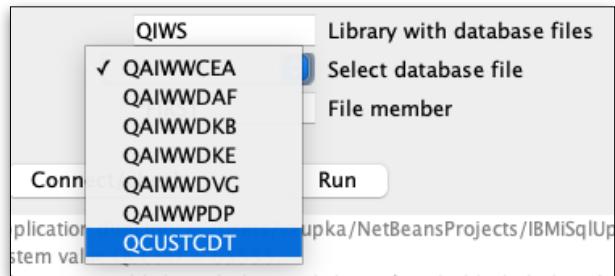
This entry represents maximum length of the display field containing the data field (column) value.

Library with database files

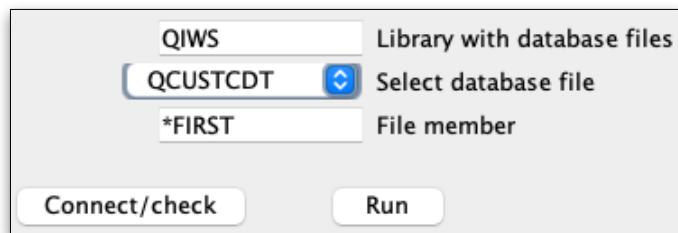
The name convention “system” is used to process SQL statements. Enter the name of the library containing database files to work with.

Select database file

The user presses the button and selects a name from the name list, say QCUSTCDT.



The selected name is then shown as the name of the button.



File member

The input field *File member* is constantly set to the value *FIRST. The application works with files of **PF-DATA** type, that is without source files (PF-SRD type). The user may overwrite the value *FIRST by name of a data member of the file if it exists.

See the chapter [Data members of physical and logical files](#) below.

Connect / check

Button Tlačítko *Connect/check* connects the application to the server, checks parameters and reports eventual error messages.

Run

If there are no errors in parameters, the button *Run* invokes a window with a table containing maximum number of the first records selected from the file (see *Parameters* above). File QCUSTCDT in library QIWS was chosen in this example because it is contained in every IBM i server.

The screenshot shows a window titled "Table QIWS/QCUSTCDT". The table has 12 rows and 11 columns. The columns are labeled: RRN, CUSNUM, LSTNAM, INIT, STREET, CITY, STATE, ZIPCOD, CDTLMT, CHGCOD, BALDUE, and CDTDUE. The data in the table is as follows:

RRN	CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
1	938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37.00	0.00
2	839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100.00	0.00
3	392859	Vine	S S	P O Box 79	Broton	VT	5046	700	1	439.00	0.00
4	938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.50	0.50
5	397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	0.00	0.00
6	389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	0.50
7	846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10.00	0.00
8	475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250.00	1.00
9	693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	0.00	0.00
10	593029	Williams	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25.00	0.00
11	192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.50	0.50
12	583991	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500.00	0.00

Below the table, there are two text frames:

- "Enter condition WHERE for row selection and press Refresh."
- "Enter condition ORDER BY for row ordering and press Refresh."

At the bottom of the window are four buttons: "Exit", "Show selected", "Refresh", and "Select columns".

Complete SELECT statement is shown under two text frames (above the button row). The list of columns and maximum number of records is apparent from the statement. The expression *RRN(QCUSTCDT) as RRN* represents relative number of the record in the file. It is displayed for information in the first column of the table in blue color. Column names are colored red, what means that data can't be changed.

Selection and ordering of records

The first display of the file is a result of the SELECT statement without using clauses WHERE an ORDER BY, so the first unordered records are selected in the maximum number specified in Parameters. More accurate, the records are ordered by the relative record number named as RRN. This order corresponds to the arrival sequence in which the records were written to the file.

To select and order records input frames under the displayed table can be used.

- To select records, clause **WHERE** of the SELECT statement is entered in the first frame.
- To order records, clause **ORDER BY** of the SELECT statement is entered in the second frame.

Refresh button performs new selection and/or ordering of records.

For example, if the entry in the first field is **BALDUE >= 30 and CHGCOD < 3** and the entry in the second field is **BALDUE DESC**, the following window is displayed with selected and ordered records and the corresponding SELECT statement.

Table QIWS/QCUSTCDT

Change a cell value: Double click, rewrite the cell value and press ENTER (or click TAB or another cell).

RRN	CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
4	938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.50	0.50
11	192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.50	0.50
3	392859	Vine	S S	P O Box 79	Broton	VT	5046	700	1	439.00	0.00
8	475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250.00	1.00
2	839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100.00	0.00
6	389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	0.50

Enter condition WHERE for row selection and press Refresh.
BALDUE >= 30 and CHGCOD < 3

Enter condition ORDER BY for row ordering and press Refresh.
BALDUE DESC

```
SELECT RRN(QCUSTCDT) as RRN, CUSNUM, LSTNAM, INIT, STREET, CITY, STATE, ZIPCOD, CDTLMT, CHGCOD, BALDUE, CDTDUE
FROM QIWS/QCUSTCDT
WHERE BALDUE >= 30 and CHGCOD < 3
ORDER BY BALDUE DESC
FETCH FIRST 200 ROWS ONLY FOR READ ONLY
```

Exit Show selected Refresh Select columns

The expressions in frames are stored in a text file with ending **.sel** in the directory **selectfiles**.

Contents of the text file has the following form.

```
selection;  
ordering
```

where *selection* is either empty text or the expression for WHERE clause and *ordering* is either empty text or the expression for ORDER BY clause. The two parts are separated by semicolon.

The file from the example is named **QIWS-QCUSTCDT.sel** and contains

```
BALDUE >= 30 and CHGCOD < 3;  
BALDUE desc
```

Note 1: As an entry in both fields, the expression **RRN(QCUSTCDT)** denoting the relative record number can be used.

Note 2: To select a binary field (type BINARY, VARBINARY), function **HEX()** must be used because the value must be written as couples of hexadecimal characters. For example, if the field BIN01 of the BINARY type the expression for WHERE is entered as follows.

```
hex(BIN01) like '%cd%'
```

or

```
hex(BIN01) >= 1A
```

and the like.

Seeing details of a record

To see details of a record select its row and press button *Show selected*. Another window is displayed with a list of field names and values from the row (record).

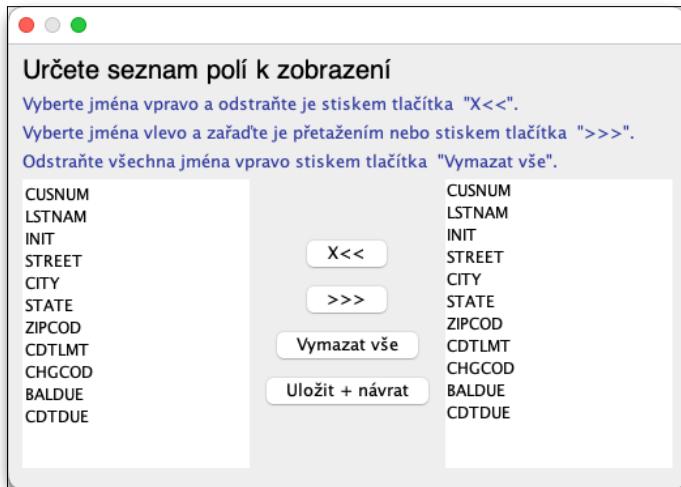
Data of the table QIWS/QCUSTCDT		
<input type="button" value="Return"/>		
CUSNUM	NUMERIC (6, 0)	938485
LSTNAM	CHAR (8)	Johnson
INIT	CHAR (3)	J A
STREET	CHAR (13)	3 Alpine Way
CITY	CHAR (6)	Helen
STATE	CHAR (2)	GA
ZIPCOD	NUMERIC (5, 0)	30545
CDTLMT	NUMERIC (4, 0)	9999
CHGCOD	NUMERIC (1, 0)	2
BALDUE	NUMERIC (6, 2)	3987.50
CDTDUE	NUMERIC (6, 2)	0.50

Button *Return* just returns the preceding window.

Selection of columns

This function enables displaying a reduced set and different ordering of columns.

Button *Select columns* shows a window with two frames and buttons between.



The left frame contains always the complete list of fields (columns) of the database file. The right frame contains the same list at first. The user can write another list in the right frame, possibly shortend, or reordered.

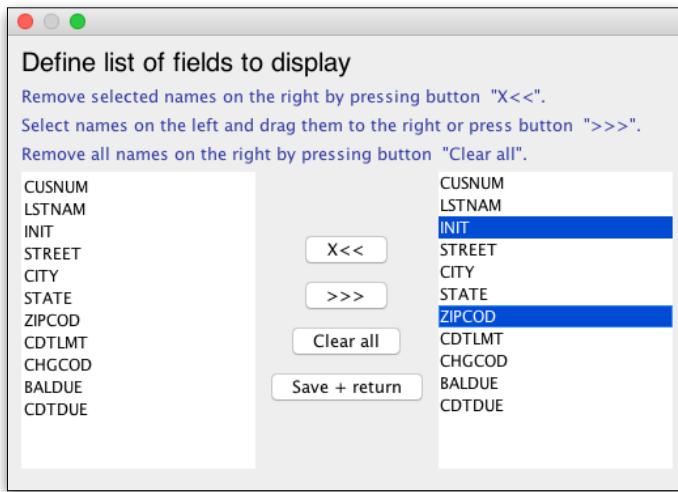
- Button **X<<** removes selected names from the right frame.
- Drag selected names from the left frame into a suitable position in the right frame or copy them to the end of the right frame by pressing button **>>>**.
- Button **Clear all** clears the whole contents of the right frame.
- Button **Save + return** saves the list of field names from the right frame to a text file in the directory *columnfiles* and shows the table with the columns selected and ordered this way.

A single name from the list is selected by left mouse click. A contiguous group of names is selected by holding *Shift* key and left mouse clicking. Non-contiguous names are selected by holding *Ctrl* key (*Cmd* key in macOS) and left mouse clicking.

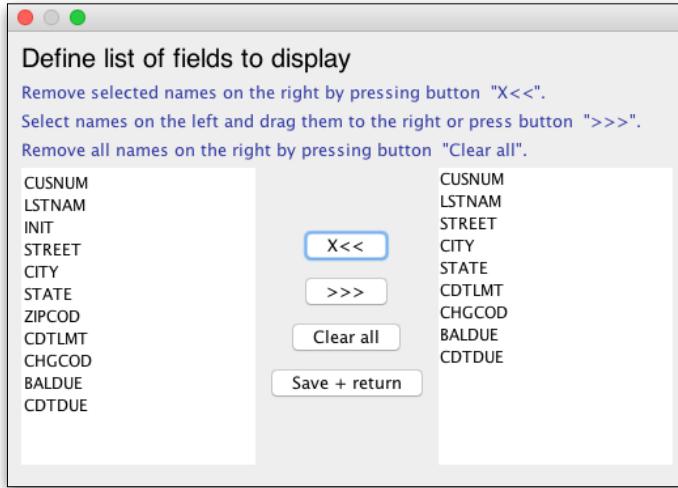
The list of selected fields amended by separating commas is stored in a work text file suffixed with **.col**. The file is placed in the directory *columnfiles*. This list and selection lasts until another selection is made on the file.

Example

After pressing the button *Select columns* the following window is displayed. Select fields ZIPCOD and INT in the right frame.



Press button X<<, the right field list is shortened.



Press button *Save + return* the resulting list is stored in the text file and the list of records is displayed with a new field configuration.

The fields list is stored to the work file **QIWS-QCUSTCDT.col** of the following contents.

, CUSNUM, LSTNAM, INIT, CDTLMT, CHGCOD, BALDUE, CDTDUE

Note: The first comma separates the list from the column RRN, which is always displayed.

Next picture shows the result.

Table QIWS/QCUSTCDT

Change a cell value: Double click, rewrite the cell value and press ENTER (or click TAB or another cell).

RRN	CUSNUM	LSTNAM	STREET	CITY	STATE	CDTLMT	CHGCOD	BALDUE	CDTDUE
4	938485	Johnson	3 Alpine Way	Helen	GA	9999	2	3987.50	0.50
11	192837	Lee	5963 Oak St	Hector	NY	700	2	489.50	0.50
3	392859	Vine	P0 Box 79	Broton	VT	700	1	439.00	0.00
8	475938	Doe	59 Archer Rd	Sutter	CA	700	2	250.00	1.00
2	839283	Jones	21B NW 135 St	Clay	NY	400	1	100.00	0.00
6	389572	Stevens	208 Snow Pass	Denver	CO	400	1	58.75	0.50

Enter condition WHERE for row selection and press Refresh.

BALDUE >= 30 and CHGCOD < 3

Enter condition ORDER BY for row ordering and press Refresh.

BALDUE DESC

```
SELECT RRN(QCUSTCDT) as RRN, CUSNUM, LSTNAM, STREET, CITY, STATE, CDTLMT, CHGCOD, BALDUE, CDTDUE
FROM QIWS/QCUSTCDT
WHERE BALDUE >= 30 and CHGCOD < 3
ORDER BY BALDUE DESC
FETCH FIRST 200 ROWS ONLY FOR READ ONLY
```

Exit

Show selected

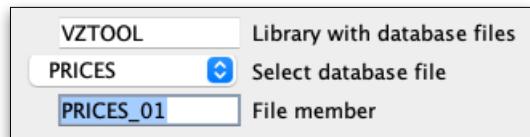
Refresh

Select columns

Data members of physical and logical files

The application can process data members of physical and logical files as if there were files. An alias object must be created in the library for the member whose name differs from the default of the file (*FIRST or file name). The *application* creates the alias object *itself* if it does not exist.

For example, the user selects the file VZTOOL/PRICES in application parameters and changes the input field *File member* to PRICES_01.



If the member exists, the application automatically creates a new alias object named identically PRICES_01 and displays the list of records contained in the member (or empty member). As soon as the application leaves the window with the displayed data, it deletes the corresponding alias object.

Alias objects

The alias objects are physical files (object type *FILE) with attribute DDMF. DDMF (Distributed Data Management File) is an object serving in access to a Remote Location from a Local Location. It is an object from SNA, APPC architecture. Here, both locations are the same and the alias object serves as a medium of access to the data file.

The user can also create alias objects using SQL statements using SQL statement CREATE ALIAS. A different means than this application must be used, e.g. STRSQL CL command. For example, if the physical file PRICES has members PRICES_01, PRICES_02, the user creates alias objects using the following SQL statements.

```
DROP ALIAS PRICES_01  
CREATE ALIAS PRICES_01 FOR VZTOOL.PRICES(PRICES_01)  
DROP ALIAS PRICES_02  
CREATE ALIAS PRICES_02 FOR VZTOOL.PRICES(PRICES_02)
```

The alias names need not be the same as the member names but it is practical.

The objects in the library look like this:

PRICES	*FILE	PF-DTA
PRICES_01	*FILE	DDMF
PRICES_02	*FILE	DDMF

The same method can be used for logical file members. If the logical file is named PRICESL and has members PRICESL_01, PRICESL_02, the alias objects look like this:

PRICESL	*FILE	PF-DTA
PRICESL_01	*FILE	DDMF
PRICESL_02	*FILE	DDMF

Data types CLOB and BLOB

These data types large objects - LOB containing data of so called advanced types CLOB (Character Large Object) and BLOB (Binary Large Object).

Columns of these types are not displayed in the list of rows in the table because the amount of data is too large for a cell in the table row. Data is displayed in a separate window by pressing the button *Show selected* if the column already contains a value that can be displayed. This means a text in case of CLOB or a bit map of certain type (e.g. a photo) in case of BLOB.

Columns can obtain a value using the application *IBMiSqlUpdate*.

In the following explanation we work with the table VZTOOL/BLOBCLOB created by statement

```
CREATE TABLE VZTOOL.BLOBCLOB (
    COL0 DECIMAL (7, 0),
    CLOB1 CLOB (1000),
    COL2 CHAR (5),
    CLOB2 CLOB (500),
    BLOB1 BLOB (10000000),
    BLOB2 BLOB
)
```

Note the sizes in parentheses at the columns of CLOB and BLOB types.

Columns CLOB1 and CLOB2 are relatively short with capacities 1000 and 500 characters respectively.

Columns BLOB1 and BLOB2 are long with capacities 10 million bytes and no explicit size (1 MB is the default value) respectively.

Displaying a CLOB column

Only two simple columns, COL0, COL2, are shown in the table row.

Table VZTOOL/BLOBCLOB

Change a cell value: Double click, rewrite the cell value and press ENTER (or click TAB or another cell).

RRN	COL0	COL2
1	1	A
2	2	B

```
SELECT RRN(BLOBCLOB) as RRN, COL0, COL2
FROM VZTOOL/BLOBCLOB
FETCH FIRST 200 ROWS ONLY FOR READ ONLY
```

Exit Show selected Refresh Select columns

Button *Show selected* invokes the next window with a list of the record fields.

Data of the table VZTOOL/BLOBCLOB

Return

COL0 DECIMAL (7, 0)	1
COL2 CHAR (5)	A
CLOB1 CLOB (1000)	CLOB1
CLOB2 CLOB (500)	CLOB2
BLOB1 BLOB (10000000)	BLOB1
BLOB2 BLOB (1048576)	BLOB2

The simple columns are placed at the beginning even if not defined in this order. CLOB and BLOB columns follow.

Pressing button CLOB2 invokes a window with the contents of the column.

Column CLOB2

Start of text: 1 Length of text: 113 Find text:

```
Manifest-Version: 1.0
Class-Path: ./jt400-21.0.4.jar ./jt400-21.0.4-javadoc.jar
Main-Class: update/U_MainWindow
```

Return Refresh Put File Page Setting Print

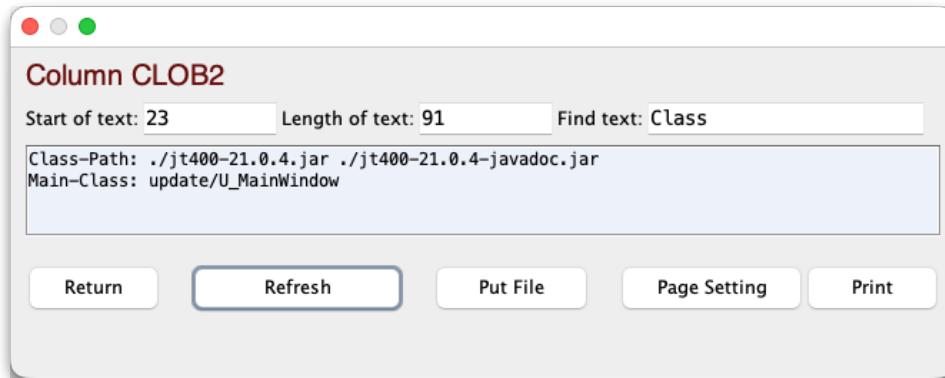
Column length is 113.

This example shows a text file that was shorter (113) than the capacity of the column (500). Data in input fields are set to initial values:

- *Start of text* is 1.
- *Length of text* is 113, i.e. length of all text contained in the column CLOB2.
- *Find text* is empty.

Finding text by a pattern

Enter text **Class** (pattern to be found) to *Find text*. and press the button *Refresh*.

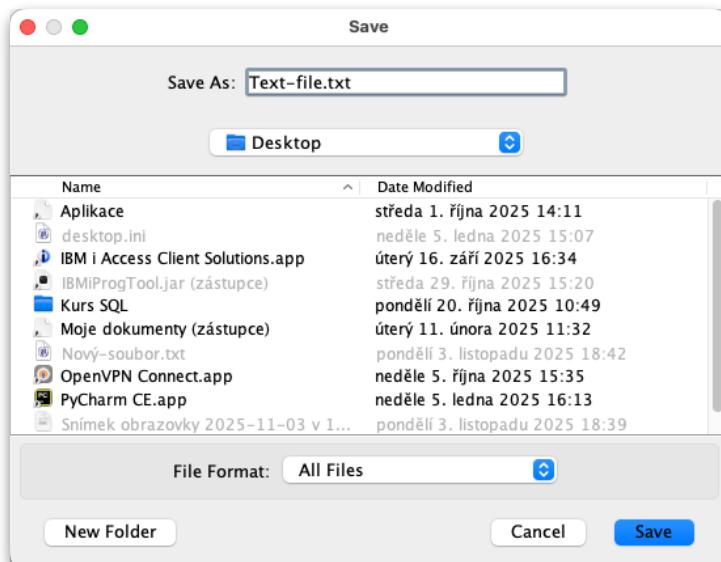


The pattern is searched from the position 1 in *Start of text* and found in position 23. Length of remaining text is 91 characters. It is the length of *displayed* text (not the whole text).

Note: Positions are counted from 1.

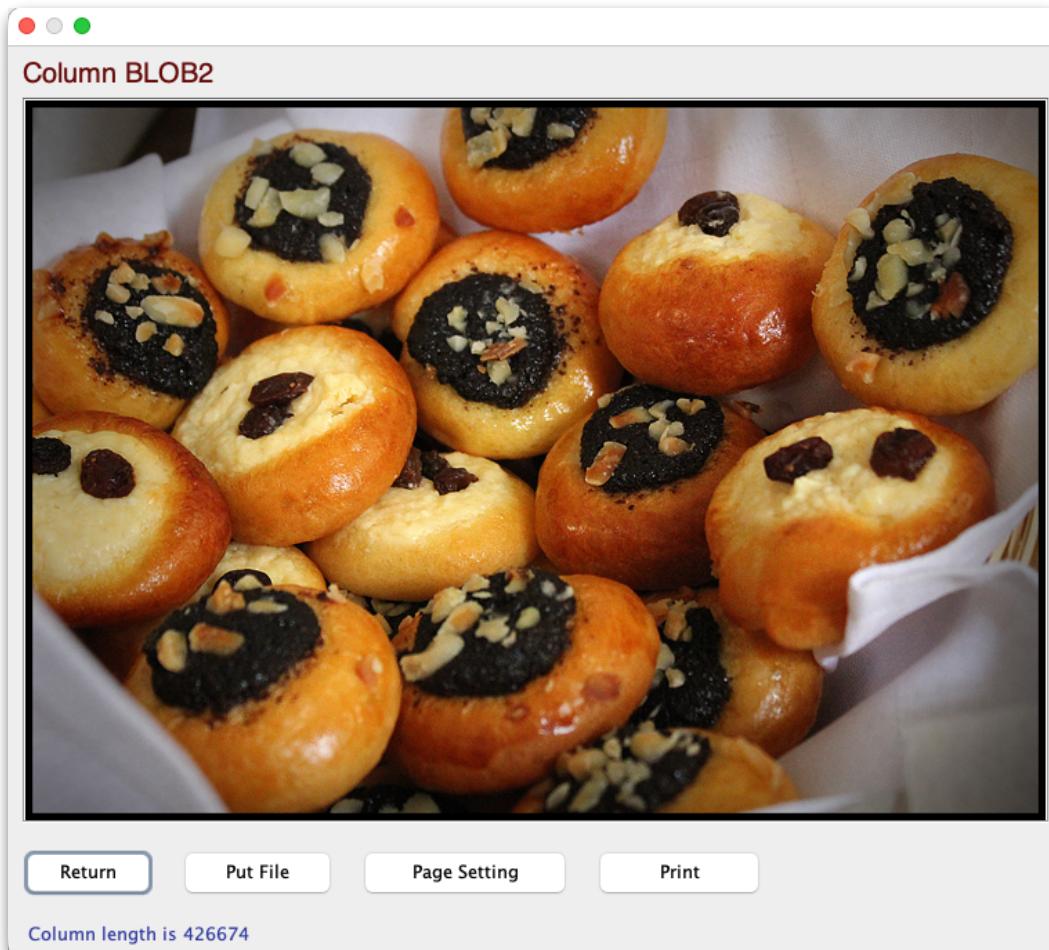
Saving column value to a file

The button *Put file* invokes a window where we enter the name of a file to the input field *Save As:* and select a directory. Then after pressing *Save* button the file is created in the directory (Desktop).



Displaying a BLOB column

Press button BLOB2 and contents of the column is displayed.



Note: Bitmap files of types JPEG, JPG, GIF, PNG, BMP, WBMP can be displayed as a picture, other types cannot.