

Revelation

Character to OpenInsight

9.2

Quick Start Guide



REVELATION
S O F T W A R E
A Division of Revelation Technologies, Inc.

COPYRIGHT NOTICE

© 1996-2010 Revelation Technologies, Inc. All rights reserved.

No part of this publication may be reproduced by any means, be it transmitted, transcribed, photocopied, stored in a retrieval system, or translated into any language in any form, without the written permission of Revelation Technologies, Inc.

SOFTWARE COPYRIGHT NOTICE

Your license agreement with Revelation Technologies, Inc. authorizes the conditions under which copies of the software can be made and the restrictions imposed on the computer system(s) on which they may be used. Any unauthorized duplication or use of any software product produced by Revelation Technologies, Inc., in whole or in part, in any manner, in print or an electronic storage-and-retrieval system, is strictly forbidden.

TRADEMARK NOTICE

OpenInsight is a trademark and Advanced Revelation is a registered trademark of Revelation Technologies, Inc.

Windows Explorer ® is a registered trademark of Microsoft, Inc.

Part No. 103-963

Printed in the United States of America.

TABLE OF CONTENTS

INTRODUCTION	4
STARTING OPENINSIGHT & CTO.....	5
I. STARTING OPENINSIGHT	5
II. OPENING AN EXISTING APPLICATION	6
STARTING CTO.....	6
I. STARTING CTO.....	6
II. CREATING YOUR TRANSLATE CONFIGURATION ITEM	12
III. SELECTING YOUR VIRTUAL TAPE.....	15
IV. LOGGING ON TO YOUR RESTORED APPLICATION.....	18
V. COMPILING AND DEBUGGING YOUR PROGRAMS.....	22
VI. DEFINING PRINTERS AND MANAGING THE SPOOLER	27
VII. LAUNCH YOUR APPLICATION IN CTO	32
STARTING OPENINSIGHT	35
I. STARTING OPENINSIGHT.....	35
II. OPENING AN EXISTING APPLICATION.....	36
III. CREATING A NEW APPLICATION	38
WORKING WITH LINEAR HASH DATA	40
I. LINEAR HASH DATA	40
II. CREATING A DATA TABLE	40
III. ADDING DATA COLUMNS TO A DATA TABLE.....	44
IV. ADDING CALCULATED COLUMNS TO A DATA TABLE	48
WORKING WITH FORMS	50
I. CREATING A DATA ENTRY FORM.....	50
II. ADDING DATA TO THE TABLE.....	52
III. CREATING AN MDI FRAME.....	54
IV. CREATING AN MDI CHILD	56
WORKING WITH POPUPS	61
I. CREATING A POPUP.....	61
II. ADDING POPUP FUNCTIONALITY TO A WINDOW	69
WORKING WITH REPORTS.....	74
I. REPORT BUILDER+	74
II. CREATING A REPORT	74
SETTING AN APPLICATION ENTRY POINT	81
I. LAUNCHING A CHARACTER FORM FROM A GUI MENU	84
II. ADDING AN OPTION TO DEFINE MACROS AND COLORS TO YOUR MENU	88

Introduction

Congratulations on using one of the most flexible and powerful databases in the world, OpenInsight from Revelation Software.

This Quick Start guide is designed to help you port your existing multivalued application into OpenInsight using the Character to OpenInsight (CTO) interface. With this new set of features built into OpenInsight, multivalued developers are able to take a virtual ACCOUNT-SAVE of their existing application and restore the saved account as an application within OpenInsight. All files will be created in a user-specified location, all dictionaries will be converted as best as possible and character-based programs will be able to be pre-compiled and run “as is” under a VT100 emulation.

The CTO interface supplements the GUI interface that is synonymous with OpenInsight. Both the GUI and CTO interfaces communicate with the Revelation OpenEngine (the “heart” of the database environment). Therefore both interfaces have access to all the tables and programs stored in the database, and can interoperate. Developers can have the best of both worlds (GUI and CTO), with no middleware add-on costs.

Rather than drawing Windows or Linux desktop graphical forms, the CTO handles the processing of traditional multivalued PRINT, INPUT, etc. statements (including of course support for cursor movement and screen attribute control). In addition, the CTO emulates a tool set and environment that are familiar to the traditional multivalued developer and user (including a command line prompt, support for PROCs and Paragraphs, a “master dictionary”, a spooler interface, and system tools like ED, BASIC, LIST/SORT, etc.)

By following this document, you should be able to:

1. Logon to SYSPROG via the CTO
2. Setup your TRANSLATE configuration item using the editor
3. Select your virtual account save tape
4. Perform an ACCOUNT-RESTORE
5. Log on to your restored application
6. Compile and debug your programs
7. Define printers and use the spooler interface
8. Launch your application in CTO
9. Modify your application using OpenInsight’s graphical interface

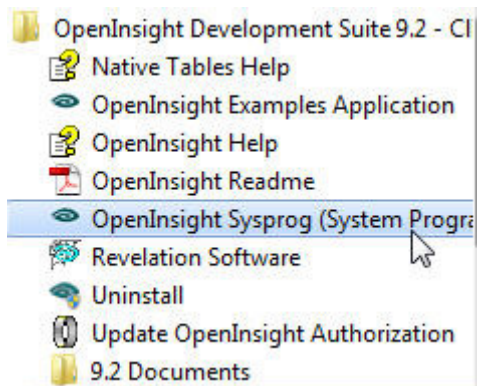
So, in less than ten steps, you will have ported an entire application.

Thanks again for trying out such a great product, and good luck!

Starting OpenInsight & CTO

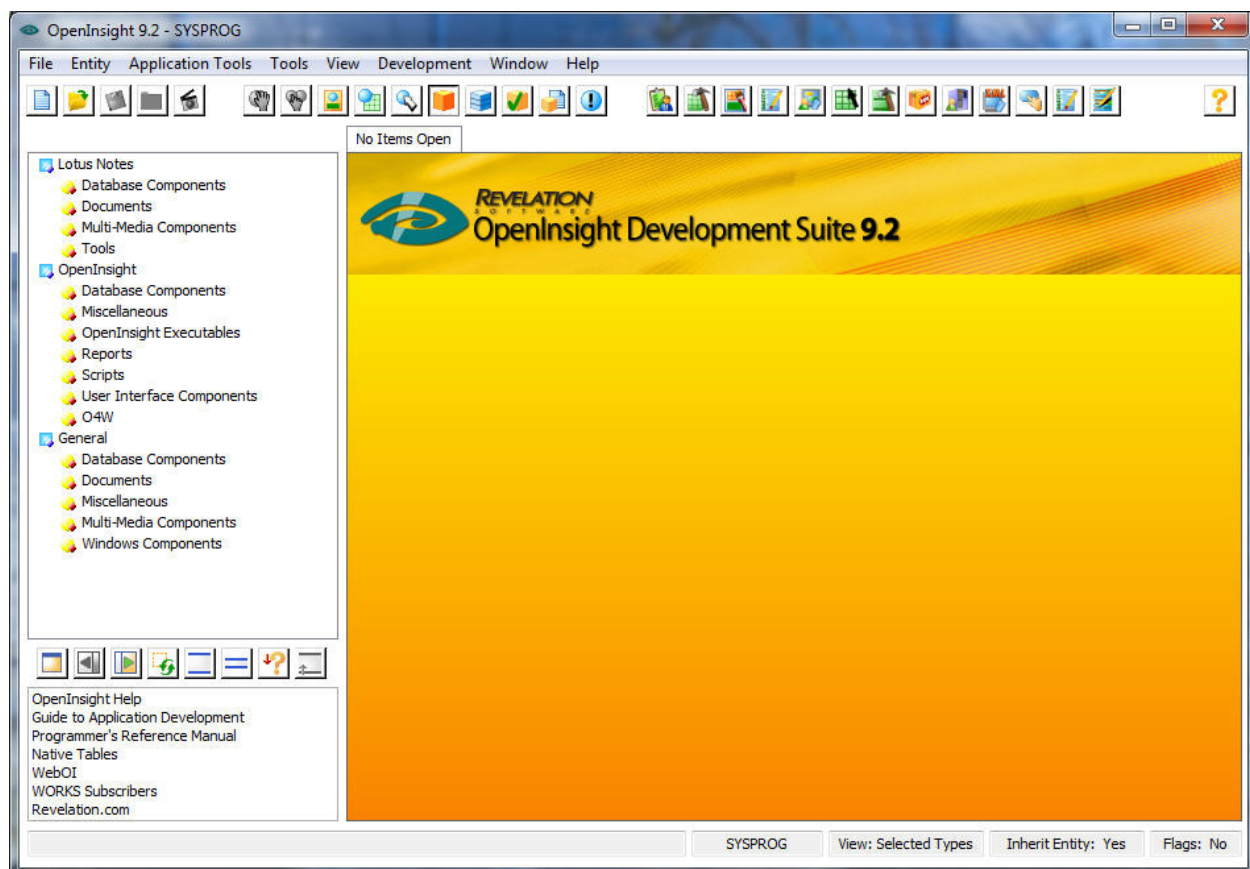
I. Starting OpenInsight

1. Launch OpenInsight.
2. Start, Programs, OpenInsight Development Suite, OpenInsight Sysprog.



II. Opening an Existing Application

3. The Application Manager window for the SYSPROG application will be displayed as shown below.

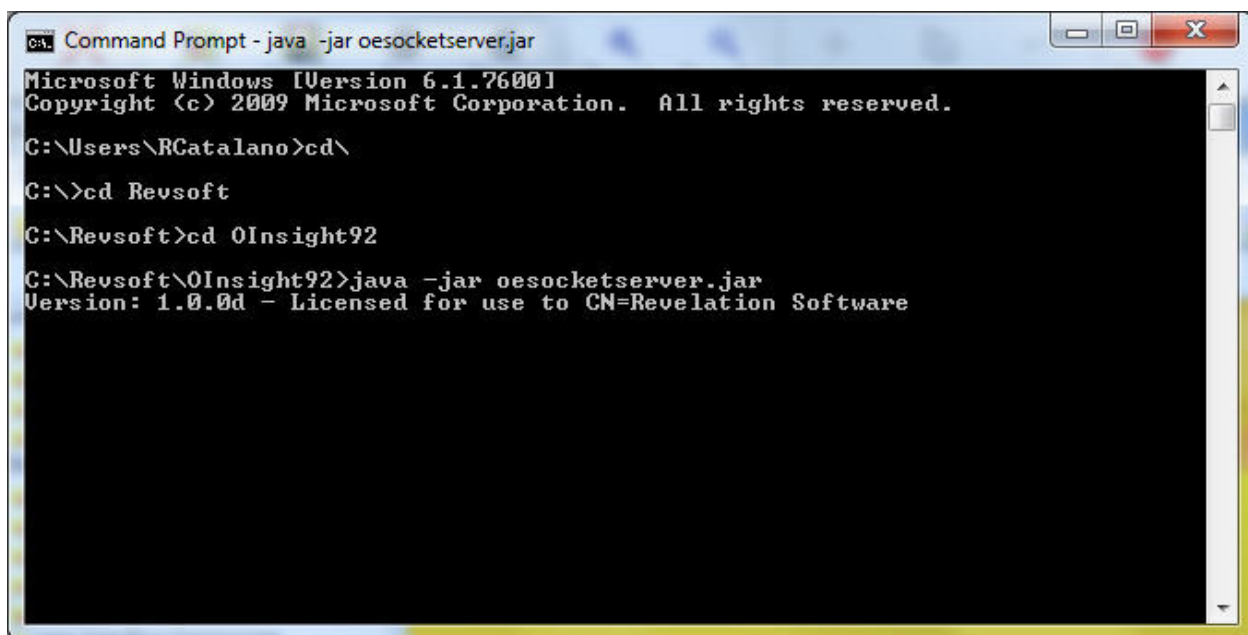


Starting CTO

I. Starting CTO

1. The OEngineServer must be running as a service or you must launch the OEngineServer application with the following command from a DOS command prompt in your OpenInsight directory:
java -jar oesocketserver.jar

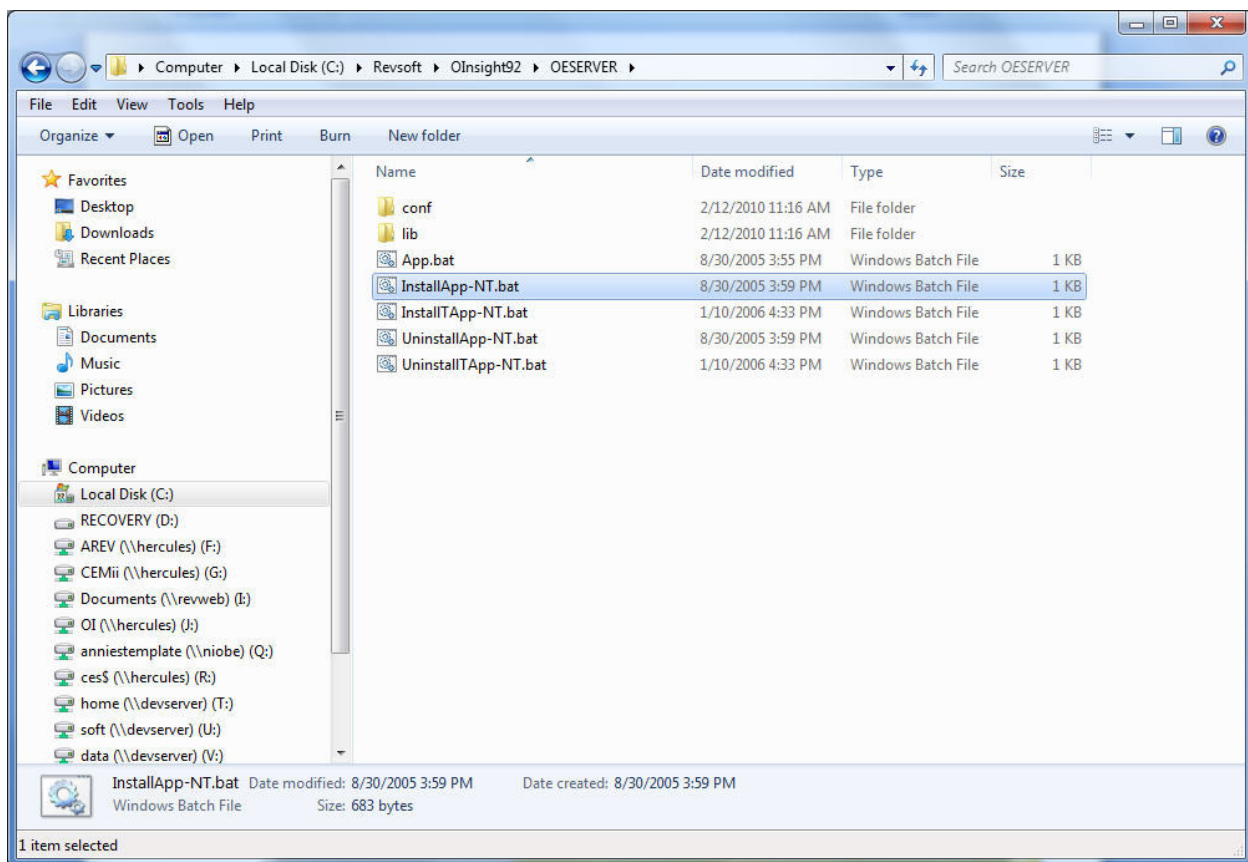
Your system will require the 32-bit Java Runtime environment which can be downloaded at:
<http://www.java.com/en/download/index.jsp>



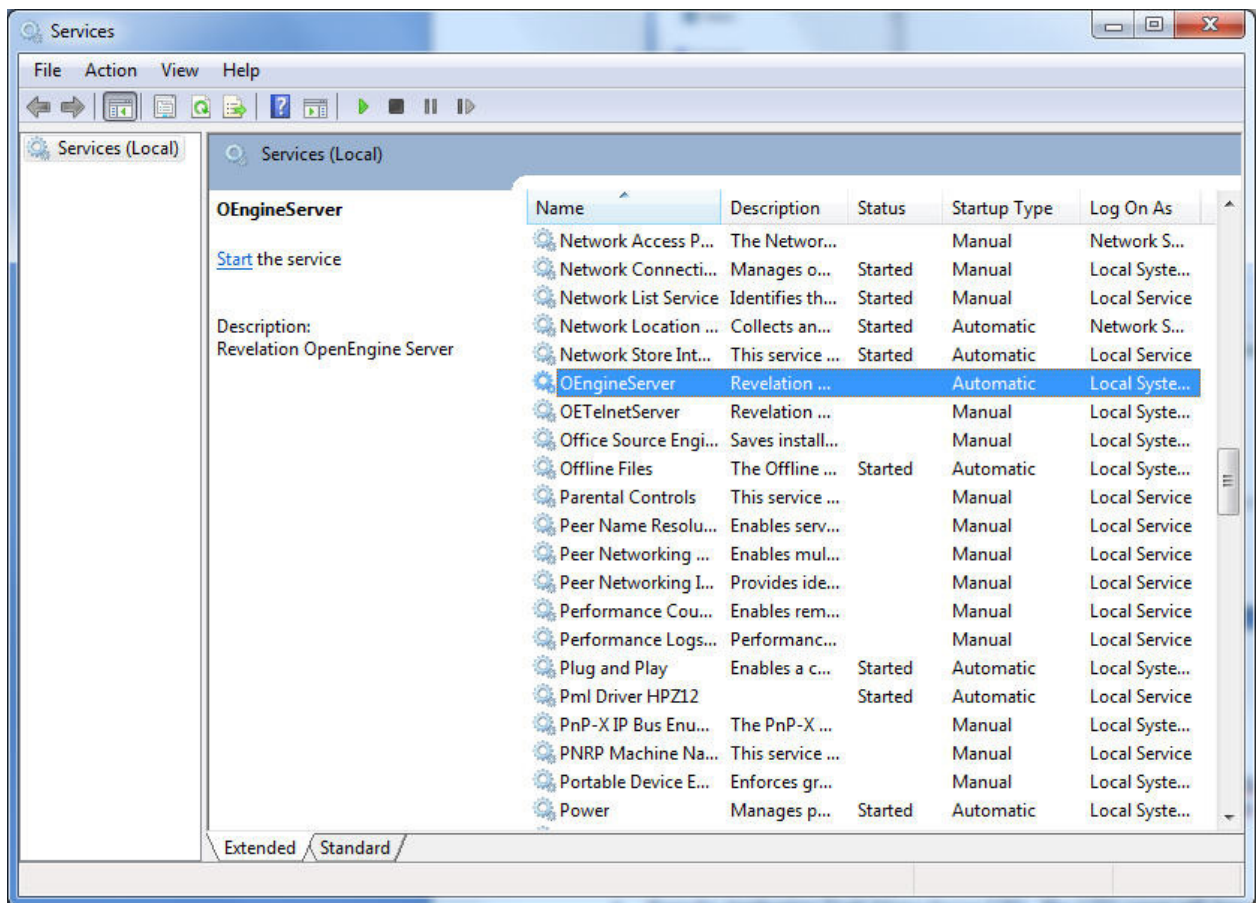
```
Command Prompt - java -jar oesocketserver.jar
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\RCatalano>cd\
C:\>cd Revsoft
C:\Revsoft>cd OInsight92
C:\Revsoft\OInsight92>java -jar oesocketserver.jar
Version: 1.0.0d - Licensed for use to CN=Revelation Software
```

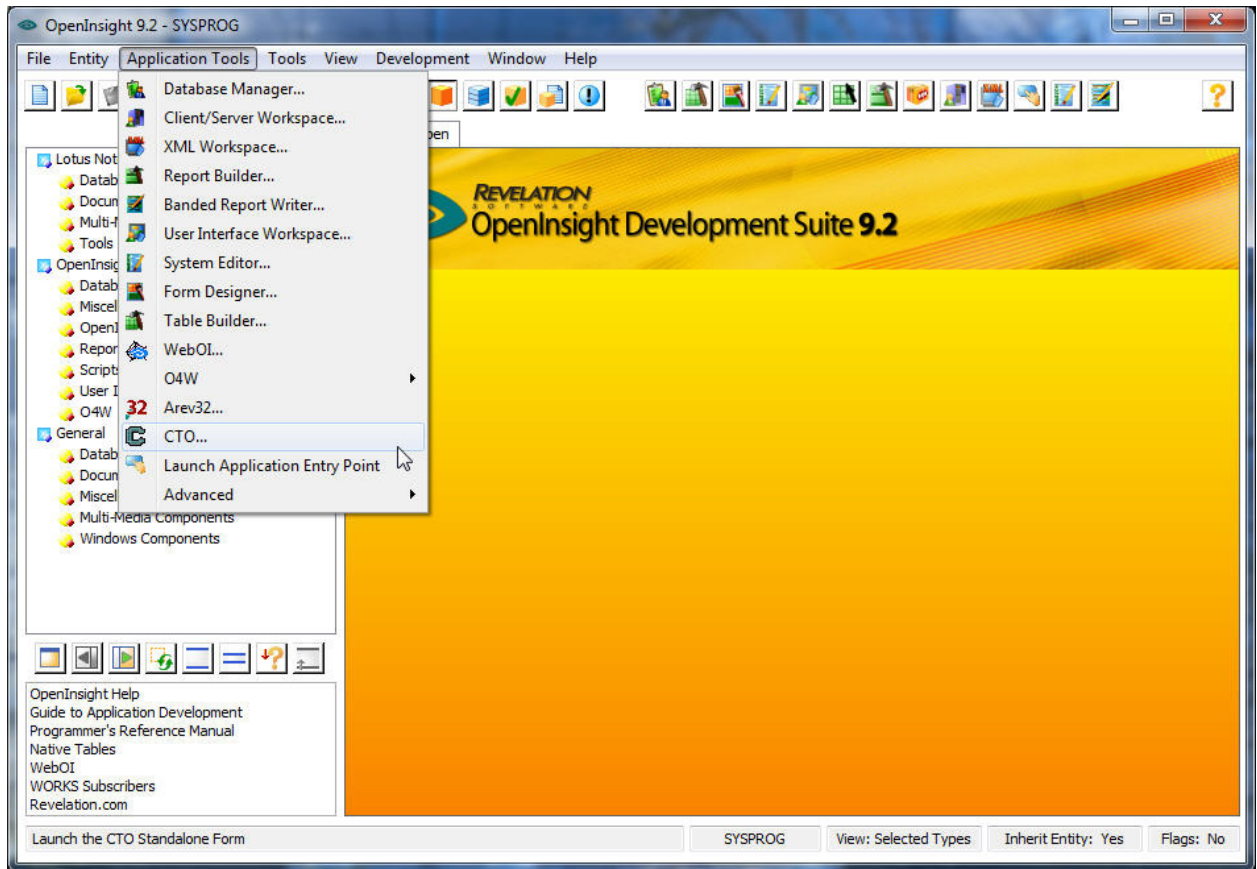
2. To install the OEngineServer as a service double-click on the InstallApp-NT.bat file found in the OESERVER folder within your OpenInsight directory.



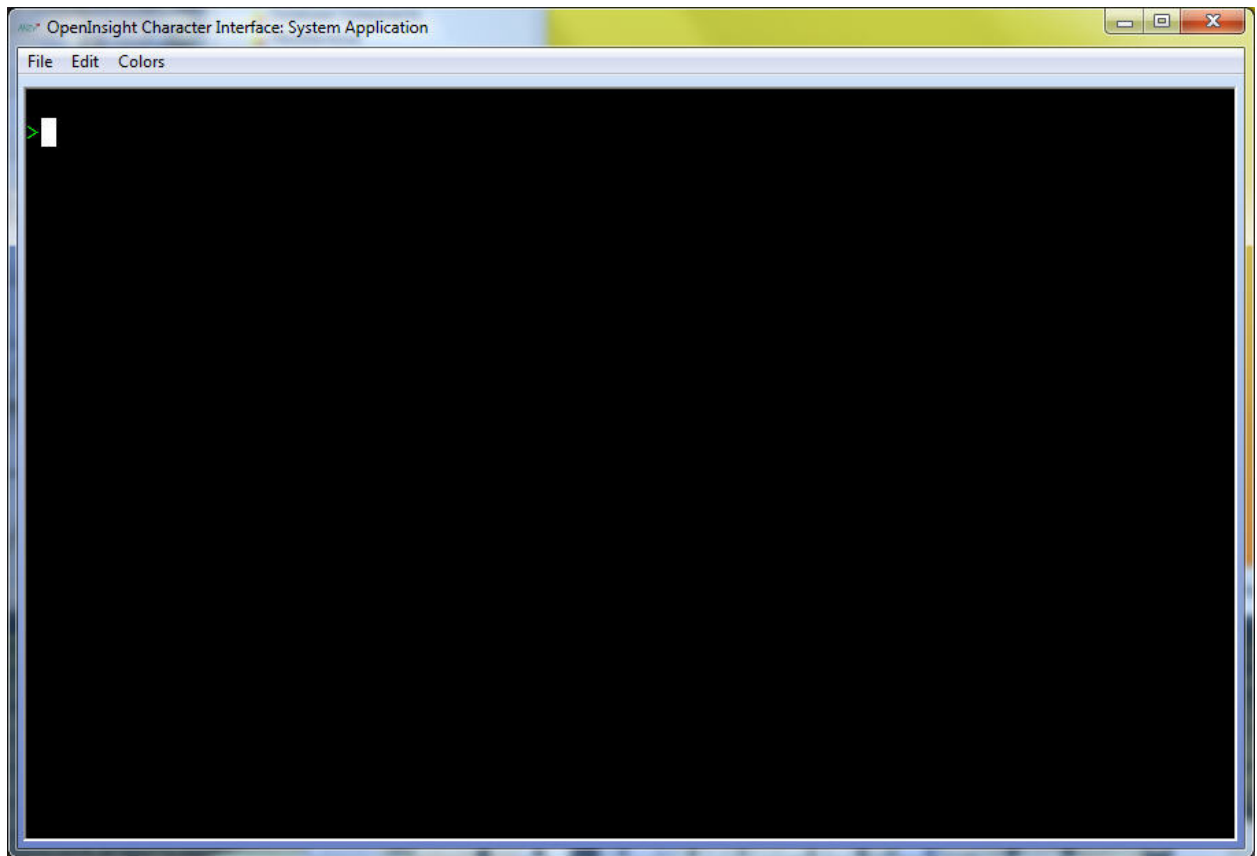
3. The OEngineServer will be installed as a service. It will need to be started.

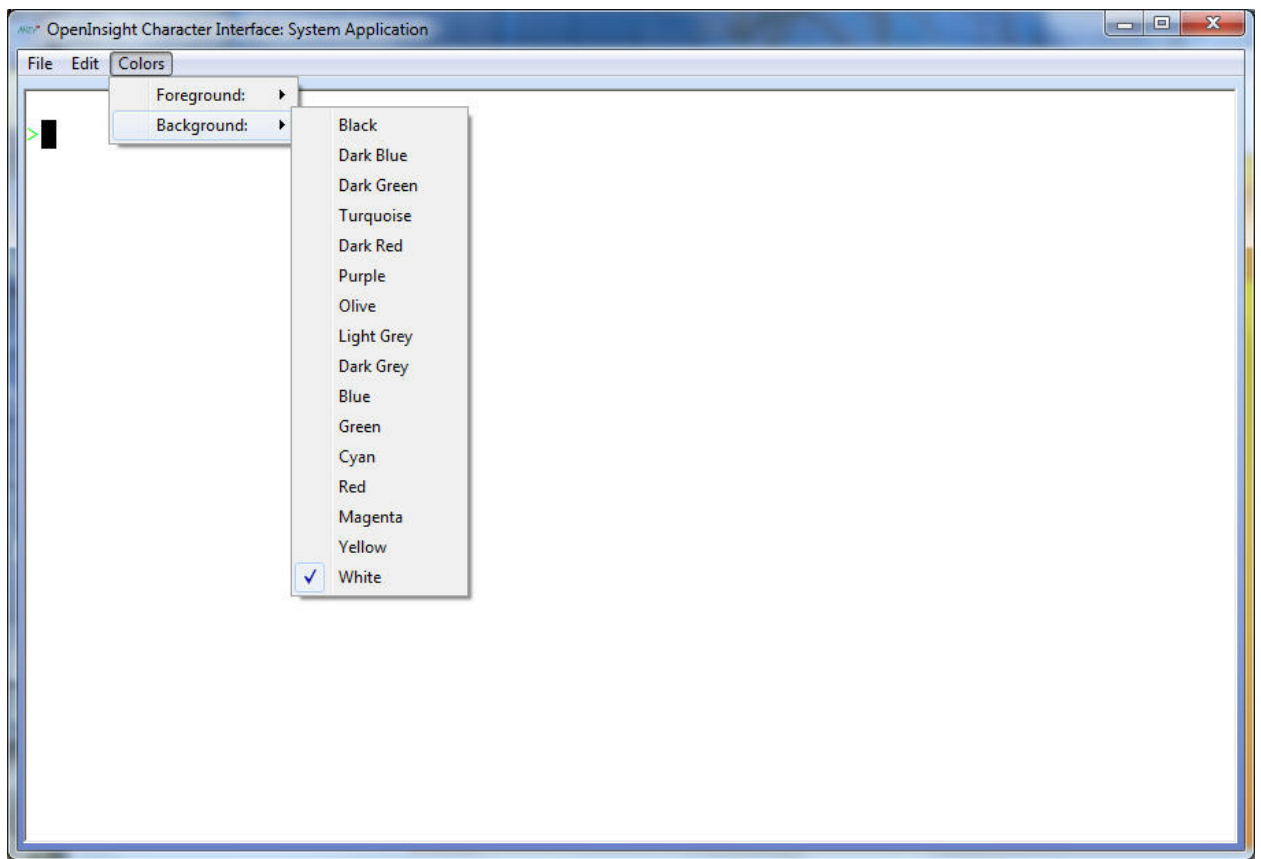


4. From the Application Tools Menu choose CTO. The CTO screen will launch to the TCL prompt. You can also set your foreground and background colors.



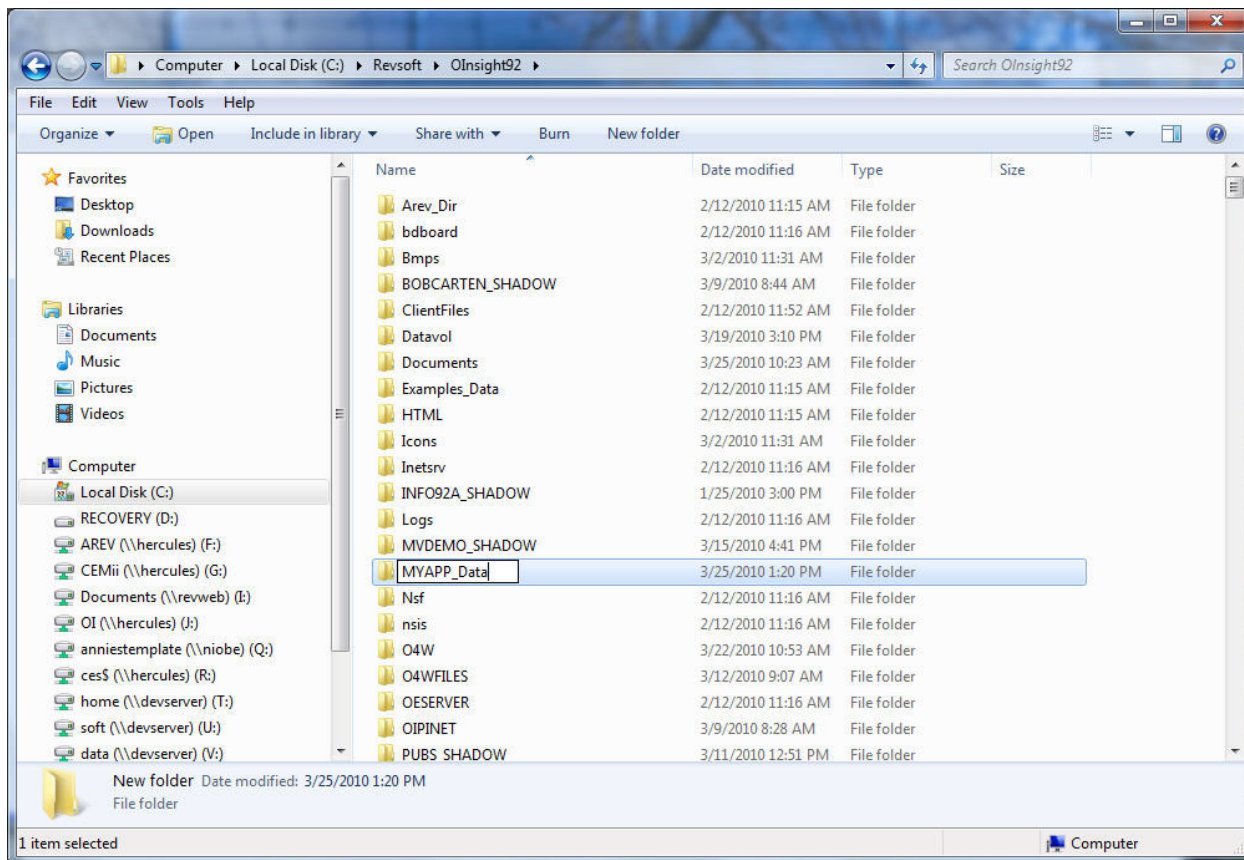
5. You can set your foreground and background colors from the Colors menu on the CTO Standalone Form.



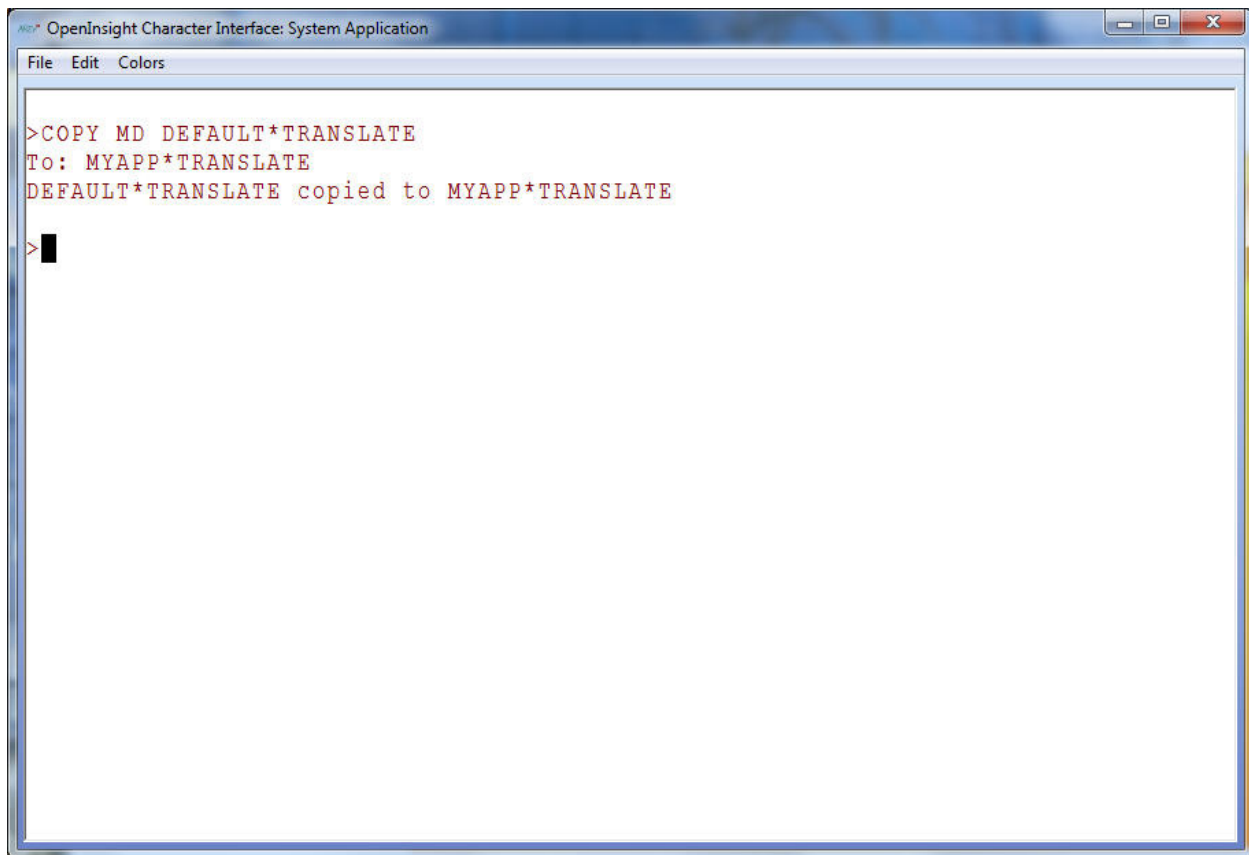


II. Creating your TRANSLATE configuration item

6. Using Windows Explorer create a new folder to store the data and dictionaries for your application. In our example we will create an application called MYAPP and store the data and dictionary tables in a folder called MYAPP_Data located within the OpenInsight directory.



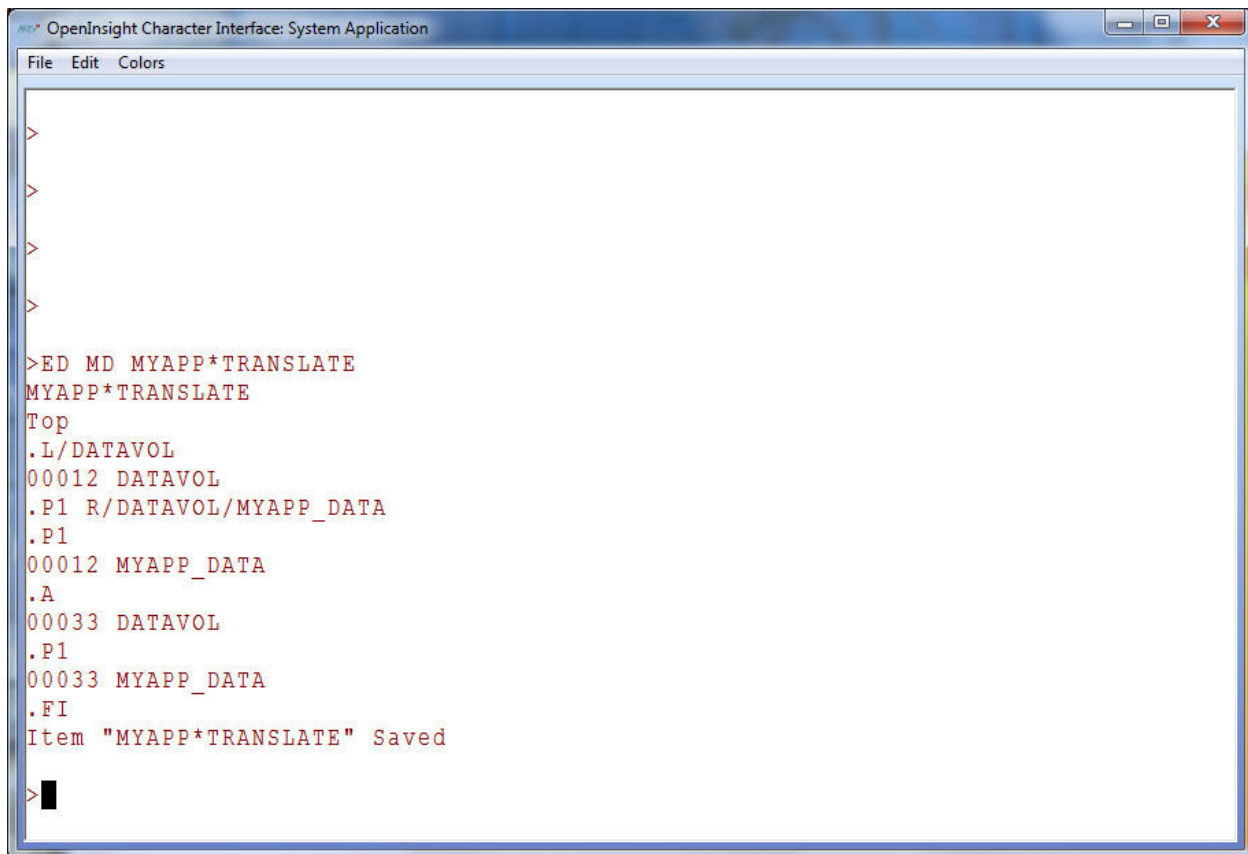
7. From the TCL prompt copy the **DEFAULT*TRANSLATE** item found in the Master Dictionary to **MYAPP*TRANSLATE** using the **COPY** command from TCL.



The screenshot shows a window titled "OpenInsight Character Interface: System Application". The window has a menu bar with "File", "Edit", and "Colors". The main area displays the following text in a monospaced font:

```
>COPY MD DEFAULT*TRANSLATE  
To: MYAPP*TRANSLATE  
DEFAULT*TRANSLATE copied to MYAPP*TRANSLATE  
> █
```

8. Using the editor (**ED**), modify the **MYAPP*TRANSLATE** item in the **MD** to tell the ACCOUNT-RESTORE process where to load your data and dictionaries.



```
OpenInsight Character Interface: System Application
File Edit Colors

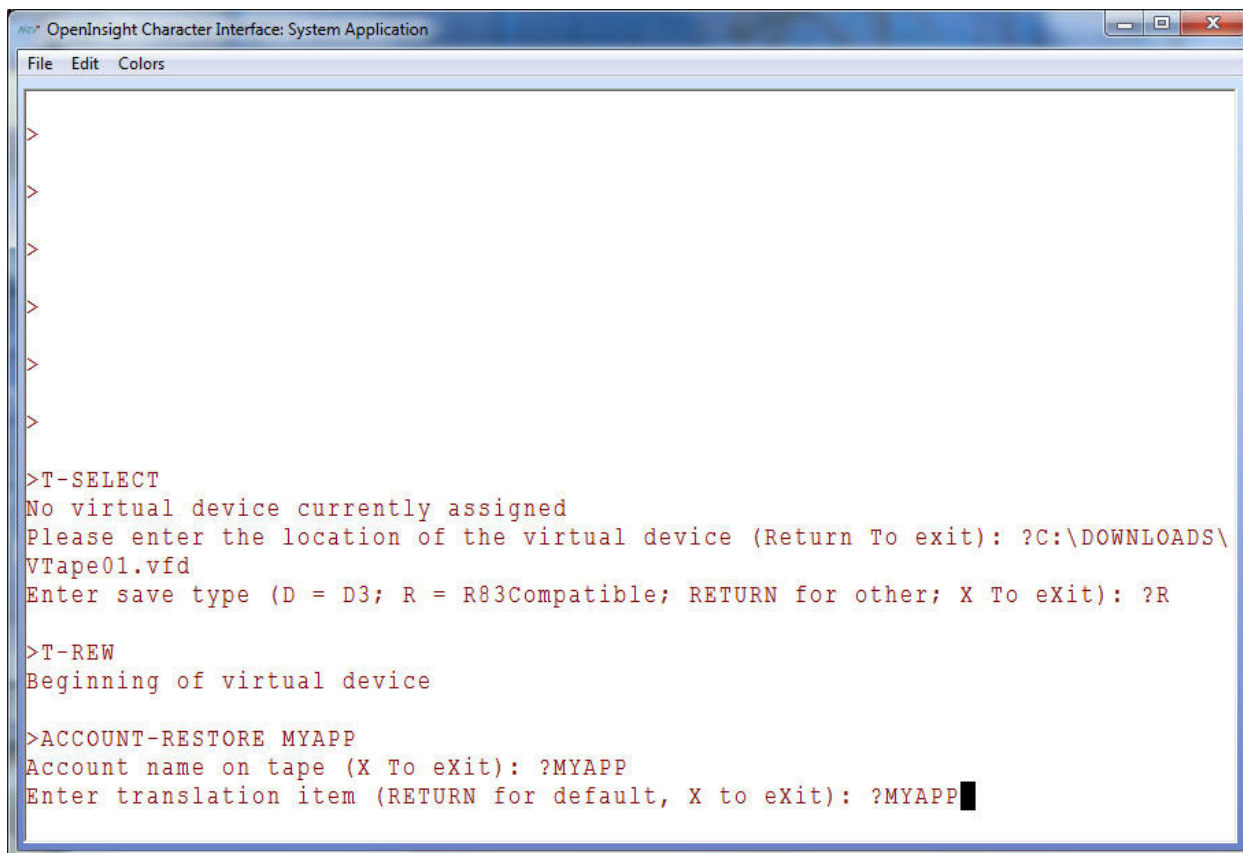
>
>
>
>

>ED MD MYAPP*TRANSLATE
MYAPP*TRANSLATE
Top
.L/DATAVOL
00012 DATAVOL
.P1 R/DATAVOL/MYAPP_DATA
.P1
00012 MYAPP_DATA
.A
00033 DATAVOL
.P1
00033 MYAPP_DATA
.FI
Item "MYAPP*TRANSLATE" Saved

>
```

III. Selecting your virtual tape

9. To select your virtual save use the **T-SELECT** command. Enter the full path of your virtual tape and save type.



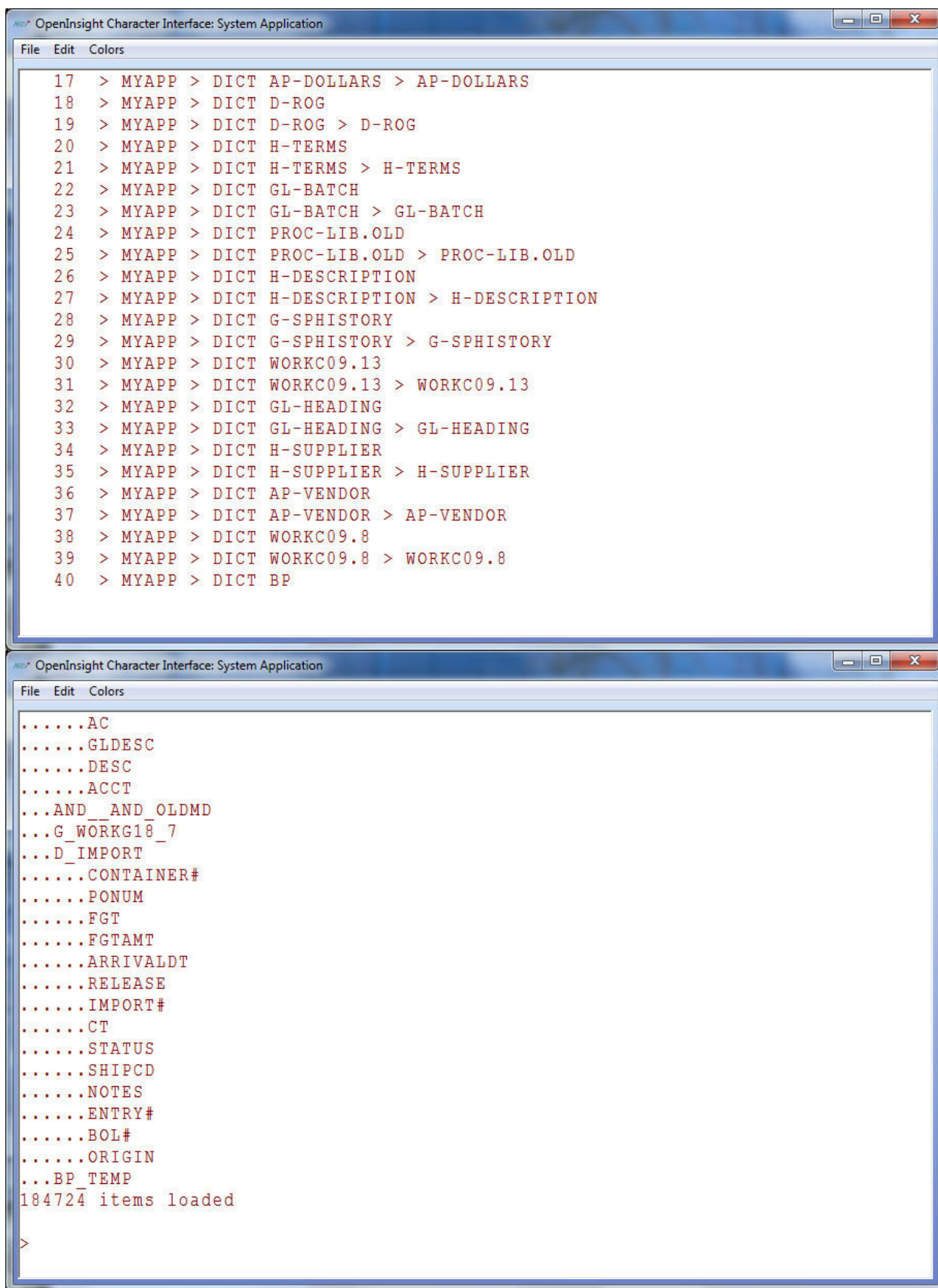
```
OpenInsight Character Interface: System Application
File Edit Colors

>
>
>
>
>
>
>
>T-SELECT
No virtual device currently assigned
Please enter the location of the virtual device (Return To exit): ?C:\DOWNLOADS\
VTape01.vfd
Enter save type (D = D3; R = R83Compatible; RETURN for other; X To eXit): ?R

>T-REW
Beginning of virtual device

>ACCOUNT-RESTORE MYAPP
Account name on tape (X To eXit): ?MYAPP
Enter translation item (RETURN for default, X to eXit): ?MYAPP
```

10. Use the **ACCOUNT-RESTORE** command to start your restore.
11. Enter APPLICATION_NAME (**MYAPP**)
12. Enter the account name on the virtual tape (**MYAPP**)
13. Enter your TRANSLATION item name (**MYAPP**)



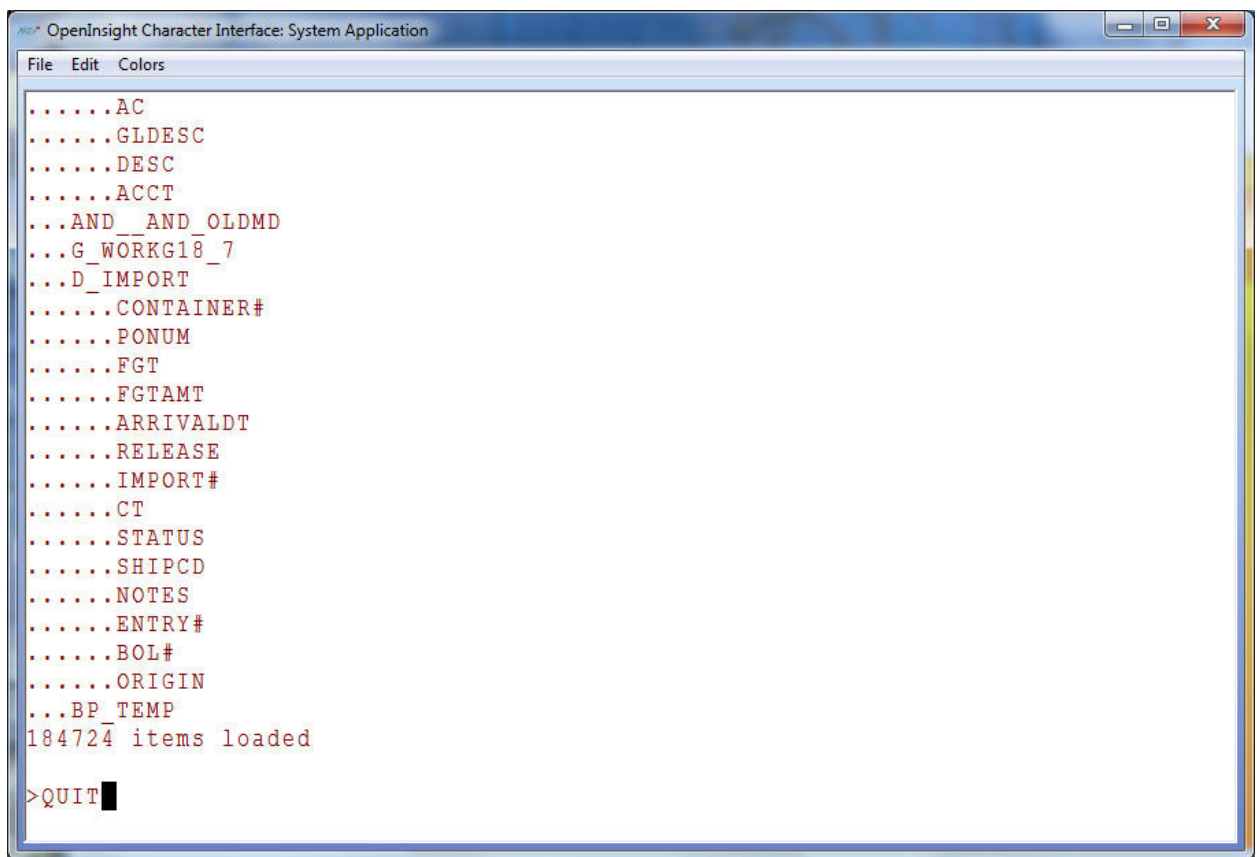
```

17 > MYAPP > DICT AP-DOLLARS > AP-DOLLARS
18 > MYAPP > DICT D-ROG
19 > MYAPP > DICT D-ROG > D-ROG
20 > MYAPP > DICT H-TERMS
21 > MYAPP > DICT H-TERMS > H-TERMS
22 > MYAPP > DICT GL-BATCH
23 > MYAPP > DICT GL-BATCH > GL-BATCH
24 > MYAPP > DICT PROC-LIB.OLD
25 > MYAPP > DICT PROC-LIB.OLD > PROC-LIB.OLD
26 > MYAPP > DICT H-DESCRIPTION
27 > MYAPP > DICT H-DESCRIPTION > H-DESCRIPTION
28 > MYAPP > DICT G-SPHISTORY
29 > MYAPP > DICT G-SPHISTORY > G-SPHISTORY
30 > MYAPP > DICT WORKC09.13
31 > MYAPP > DICT WORKC09.13 > WORKC09.13
32 > MYAPP > DICT GL-HEADING
33 > MYAPP > DICT GL-HEADING > GL-HEADING
34 > MYAPP > DICT H-SUPPLIER
35 > MYAPP > DICT H-SUPPLIER > H-SUPPLIER
36 > MYAPP > DICT AP-VENDOR
37 > MYAPP > DICT AP-VENDOR > AP-VENDOR
38 > MYAPP > DICT WORKC09.8
39 > MYAPP > DICT WORKC09.8 > WORKC09.8
40 > MYAPP > DICT BP

.....AC
.....GLDESC
.....DESC
.....ACCT
...AND__AND_OLDMD
...G_WORKG18_7
...D_IMPORT
.....CONTAINER#
.....PONUM
.....FGT
.....FGTAMT
.....ARRIVALDT
.....RELEASE
.....IMPORT#
.....CT
.....STATUS
.....SHIPCD
.....NOTES
.....ENTRY#
.....BOL#
.....ORIGIN
...BP_TEMP
184724 items loaded
>

```

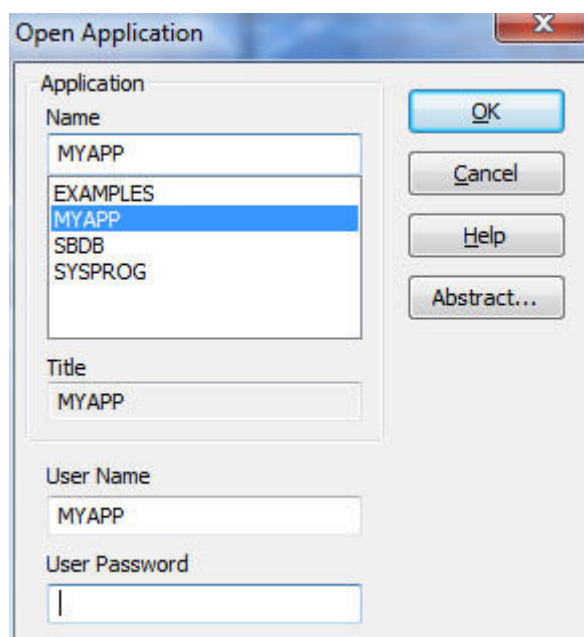
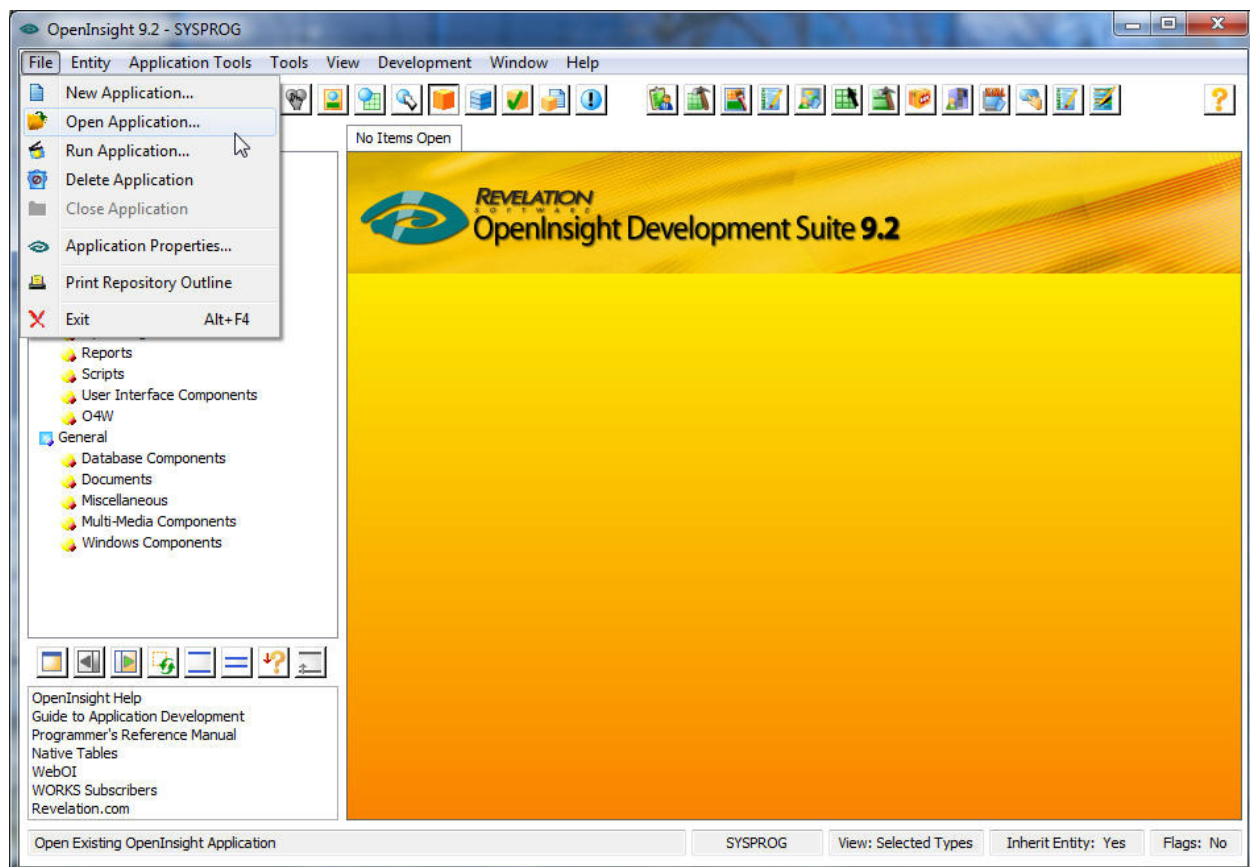
14. The ACCOUNT-RESTORE process will perform a second pass and convert the data dictionaries into OpenInsight format. Once this process is completed type **QUIT** to log out of SYSPROG and close the CTO interface.



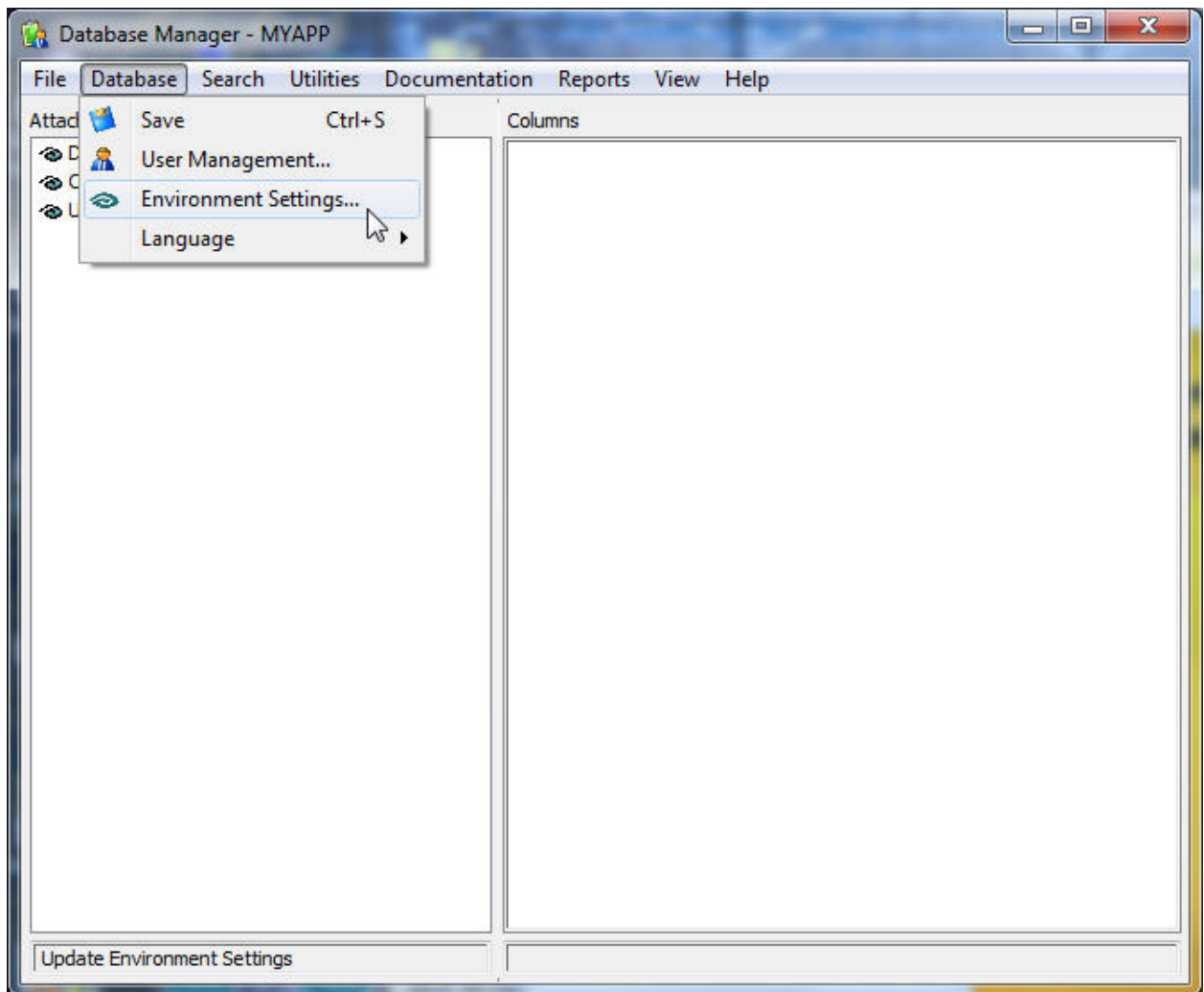
```
OpenInsight Character Interface: System Application
File Edit Colors
.....AC
.....GLDESC
.....DESC
.....ACCT
...AND__AND_OLDMD
...G_WORKG18_7
...D_IMPORT
.....CONTAINER#
.....PONUM
.....FGT
.....FGTAMT
.....ARRIVALDT
.....RELEASE
.....IMPORT#
.....CT
.....STATUS
.....SHIPCD
.....NOTES
.....ENTRY#
.....BOL#
.....ORIGIN
...BP_TEMP
184724 items loaded
>QUIT
```

IV. Logging on to your restored application

15. In our example the ACCOUNT-RESTORE process has created an application within OpenInsight called MYAPP. A default user has also been created within that application with the same name as the application itself. From File, Open Application, select application **MYAPP** and User Name **MYAPP** to log on to your newly restored application.



16. Once logged on to the MYAPP application go to the Database Manager and set your Environment Settings. You can also choose to set your foreground and background colors for the MYAPP application.



Maintain Application Environment Settings

Environment Settings

Use this screen for creating or modifying environment settings.

General | Indexes/Reports | Concurrency | Report Security - Tables | Report Security - Columns | Window/Form Settings

Default Directories
Sort Path: %TEMP%
Data Path: DATAVOL

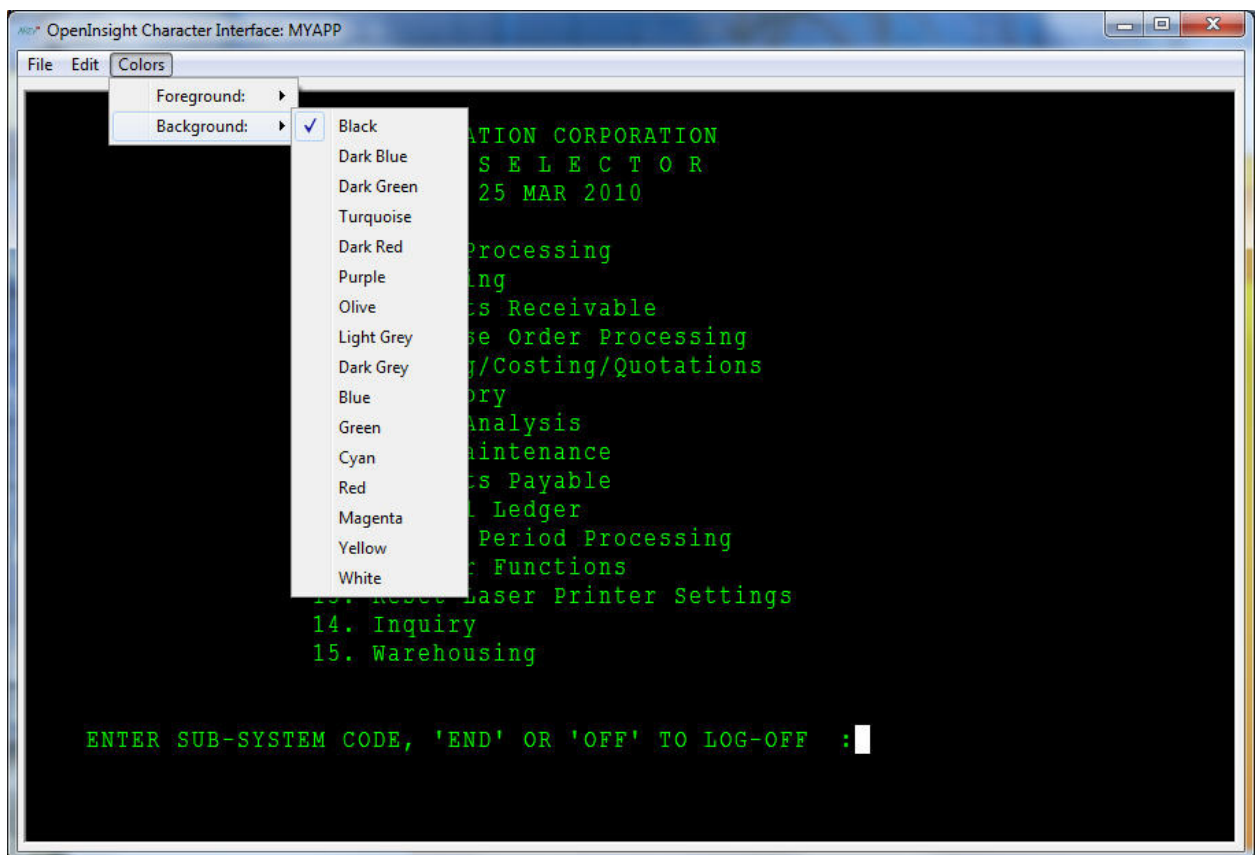
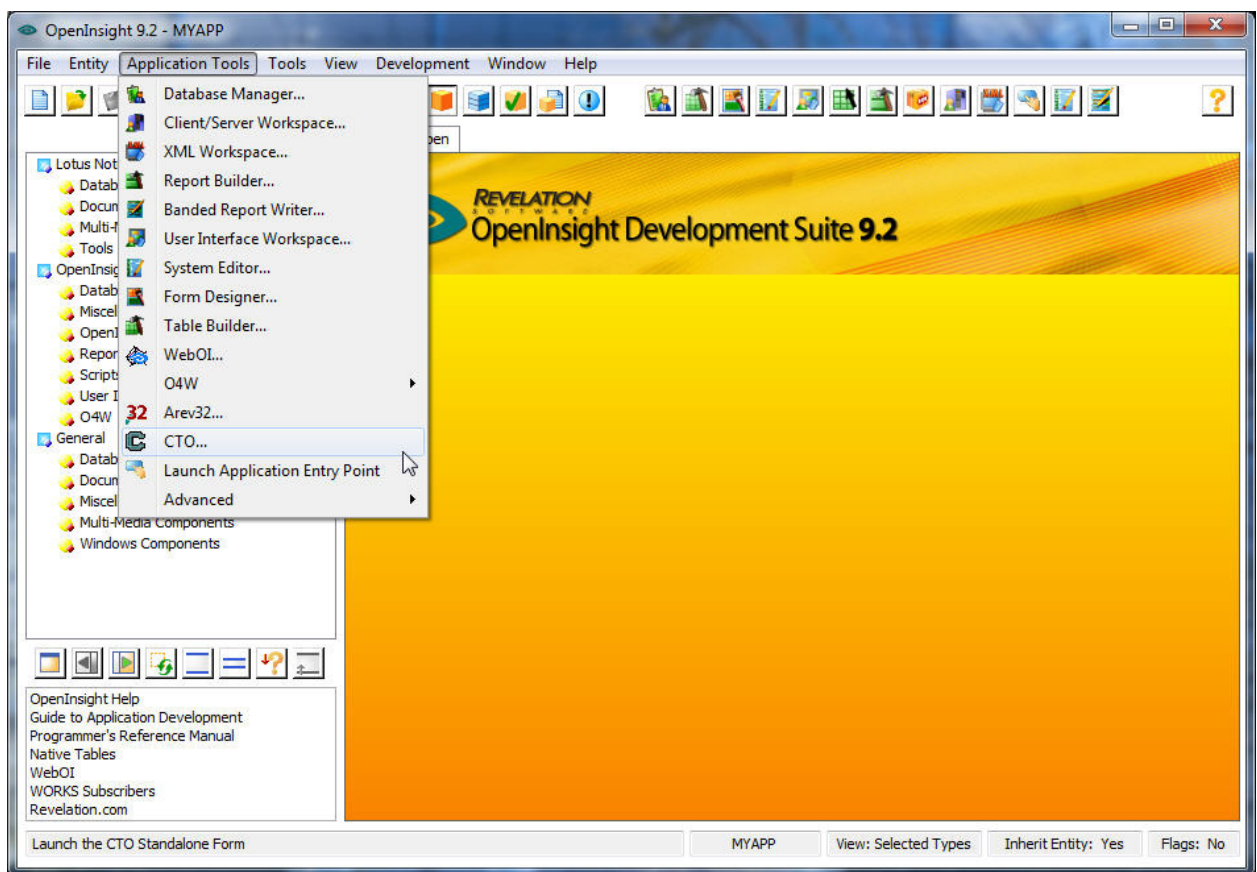
Localization
☒ None
☐ Use LND
☐ Windows Locale
Edit Language Sets

Debugger Settings
☐ Disable
☒ Enable
☐ Intercept
Intercept Routine: DEBUGGER_REPLACEMENT

General
SQL Null String:
TimeDate Format:
☐ Default IO Conversion
☐ Sort Tables by Database ID on Add Tables Dialog
Repository Log Entries: 10

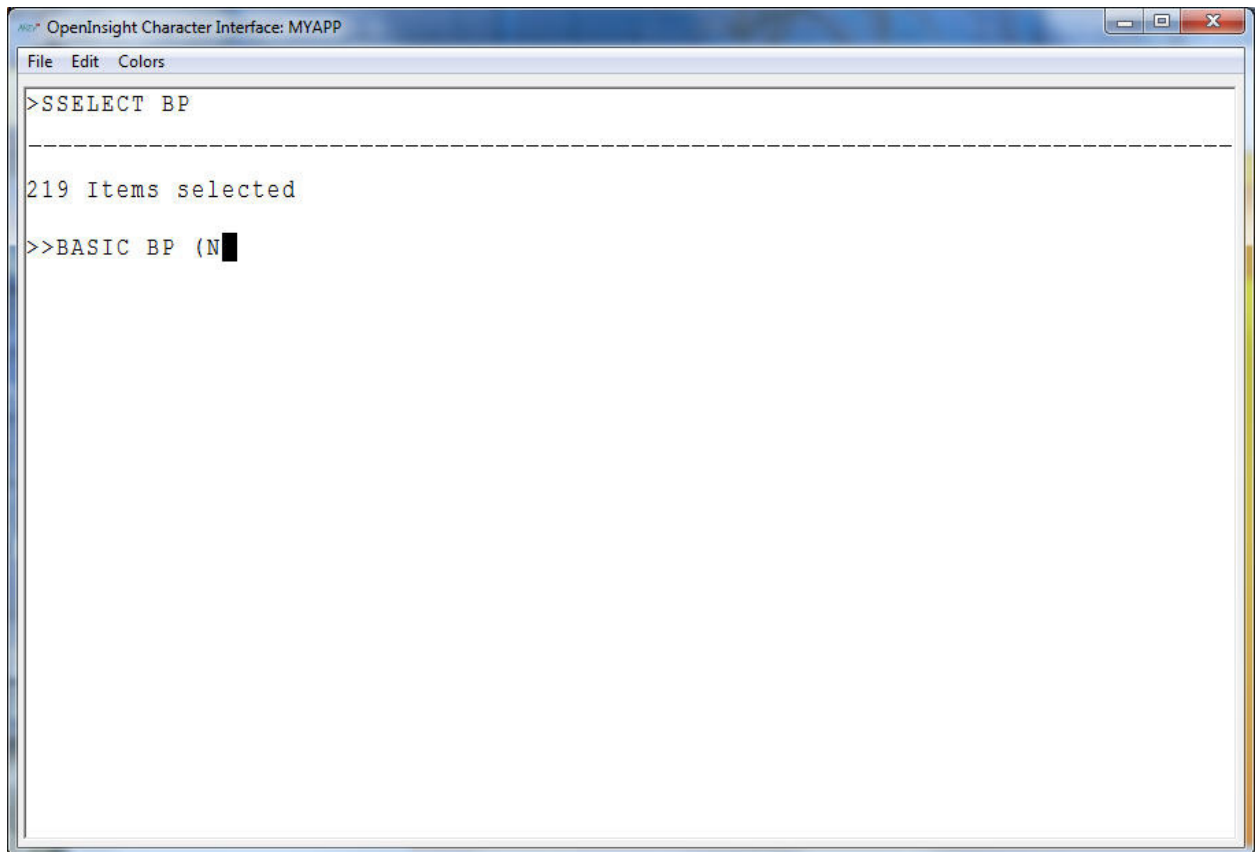
Arev32 Settings
☐ Create Arev32 User
☒ Save Precompiled Source

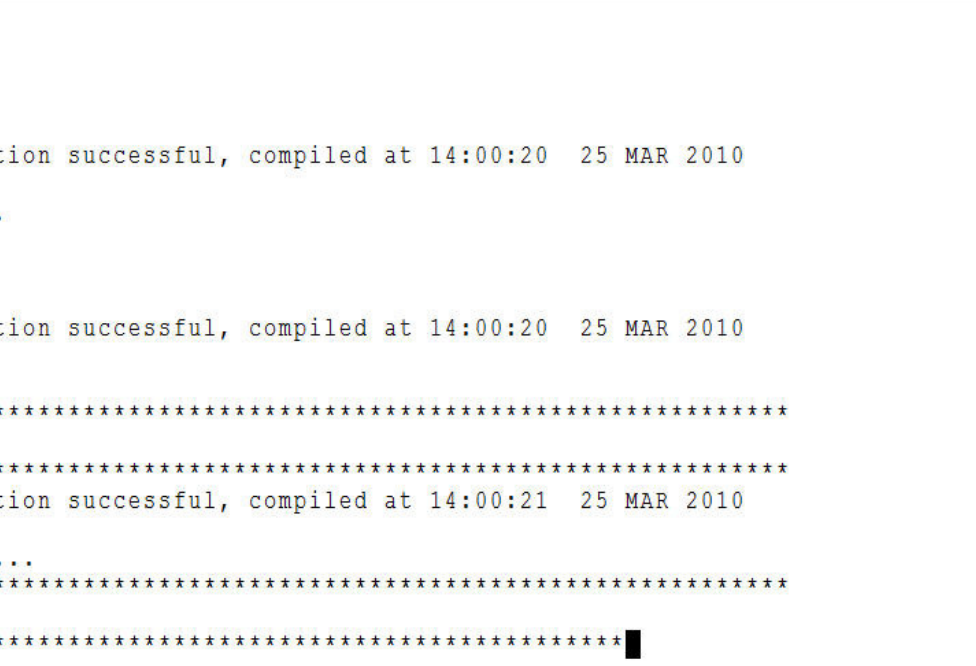
OK Cancel Help



V. Compiling and debugging your programs

17. During the ACCOUNT-RESTORE process, only your source code is restored. Source code must be compiled and subroutines must be compiled and cataloged.
18. Programs can be compiled using the BASIC command from TCL or by using FIB from the line editor.





The screenshot shows a window titled "OpenInsight Character Interface: MYAPP". The window has a menu bar with "File", "Edit", and "Colors". The main area displays the following text:

```
A07...
*****

*****
Compilation successful, compiled at 14:00:20  25 MAR 2010

A07.1...
***

***
Compilation successful, compiled at 14:00:20  25 MAR 2010

A15...
*****

*****
Compilation successful, compiled at 14:00:21  25 MAR 2010

A15.OLD...
*****

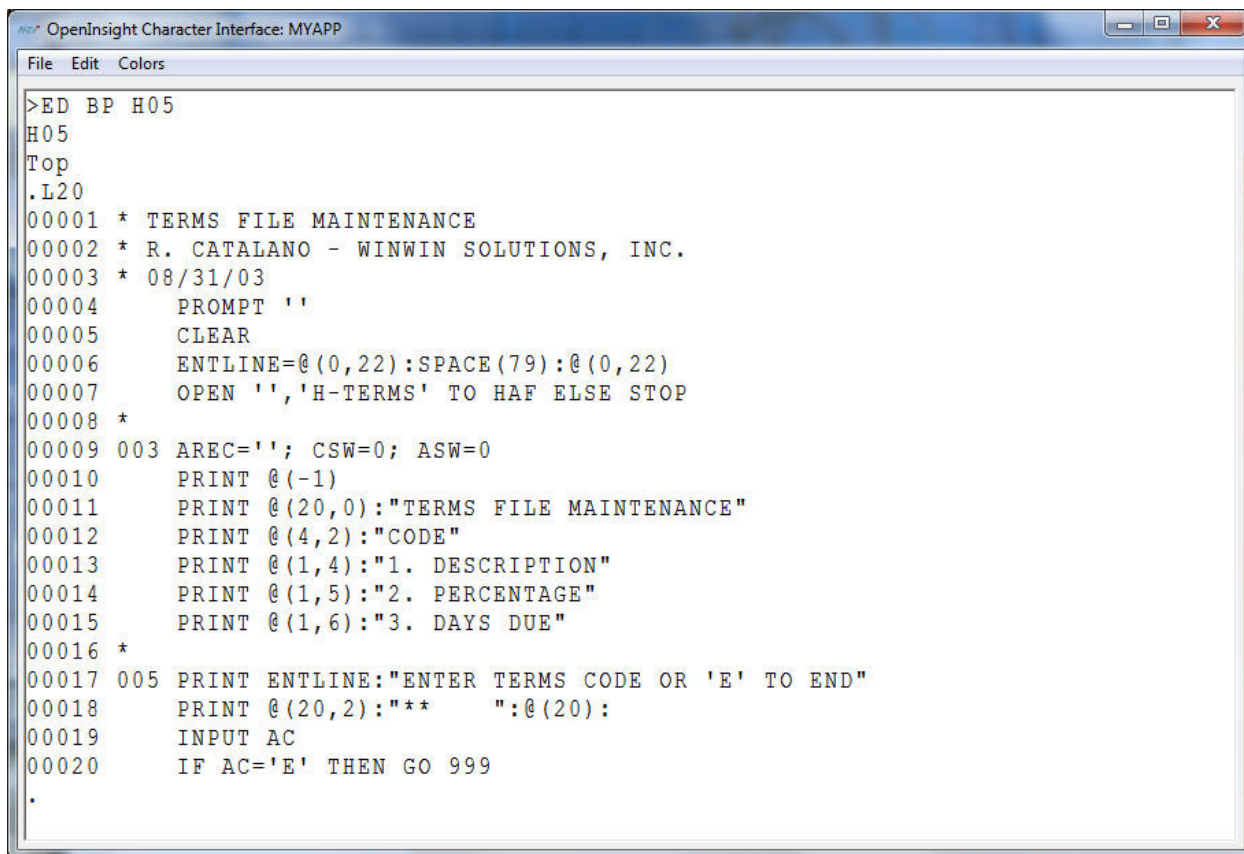
*****
```

The screenshot shows a window titled "OpenInsight Character Interface: MYAPP". The window has a menu bar with "File", "Edit", and "Colors". On the left side, there is a vertical scrollbar. The main area contains several lines of text:

```
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>FIND BP "SUBROUTINE"  
11 items selected  
  
>>CATALOG BP
```

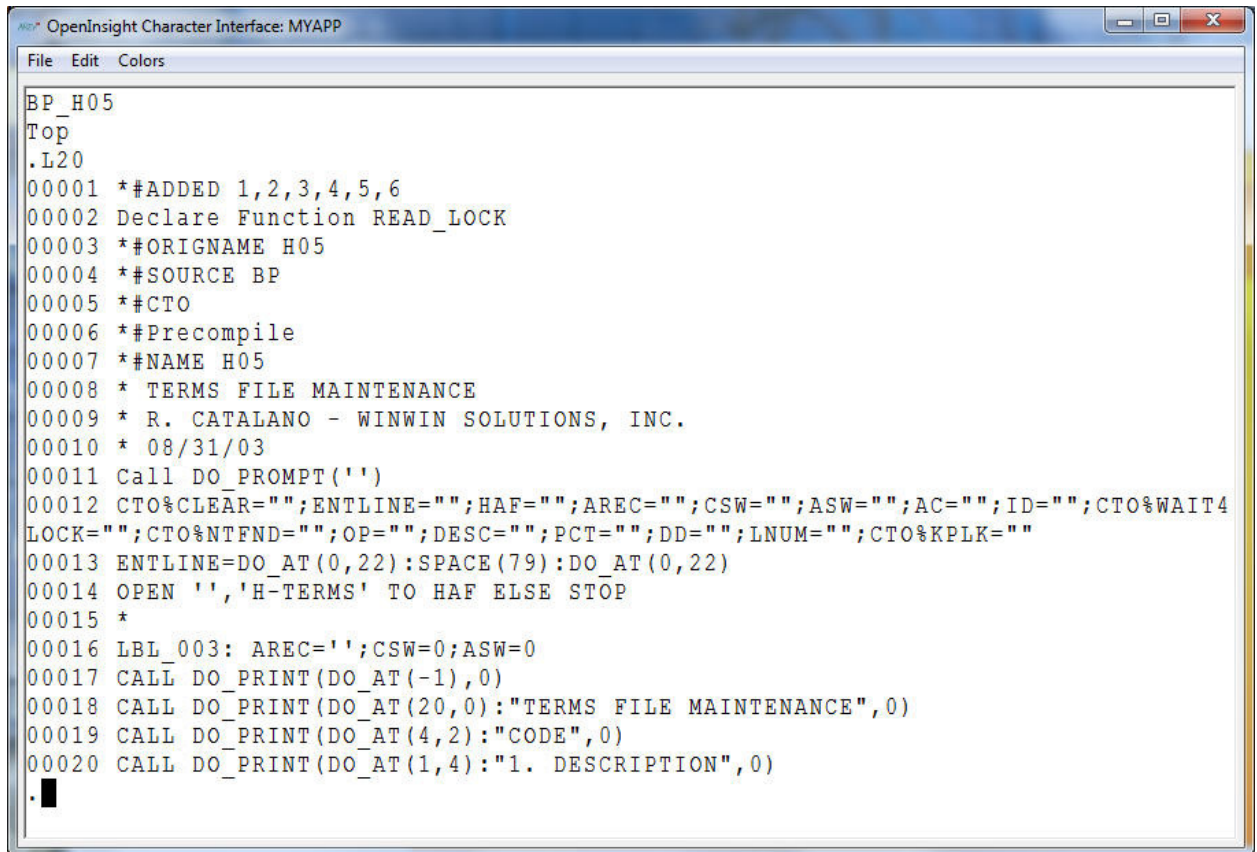

A black cursor is positioned at the end of the last line.

19. Source code can reside in any file/table and can be edited using ED from the TCL command prompt.



```
>ED BP H05
H05
Top
.L20
00001 * TERMS FILE MAINTENANCE
00002 * R. CATALANO - WINWIN SOLUTIONS, INC.
00003 * 08/31/03
00004 * PROMPT ' '
00005 * CLEAR
00006 * ENTLINE=@(0,22):SPACE(79):@(0,22)
00007 * OPEN ' ', 'H-TERMS' TO HAF ELSE STOP
00008 *
00009 003 AREC=' '; CSW=0; ASW=0
00010 PRINT @(-1)
00011 PRINT @(20,0):"TERMS FILE MAINTENANCE"
00012 PRINT @(4,2):"CODE"
00013 PRINT @(1,4):"1. DESCRIPTION"
00014 PRINT @(1,5):"2. PERCENTAGE"
00015 PRINT @(1,6):"3. DAYS DUE"
00016 *
00017 005 PRINT ENTLINE:"ENTER TERMS CODE OR 'E' TO END"
00018 PRINT @(20,2):"*** " :@(20):
00019 INPUT AC
00020 IF AC='E' THEN GO 999
.
```

20. When source code is compiled from the CTO interface, it is actually passed through a pre-compiler. A copy of the pre-compiled source code is stored in a table called `OI_PRECOMPILED_CODE` with a key of `TABLENAME_PROGRAMNAME`. To examine the pre-compiled source code for the program H05 located in the file BP type the following: `ED OI_PRECOMPILED_CODE BP_H05`



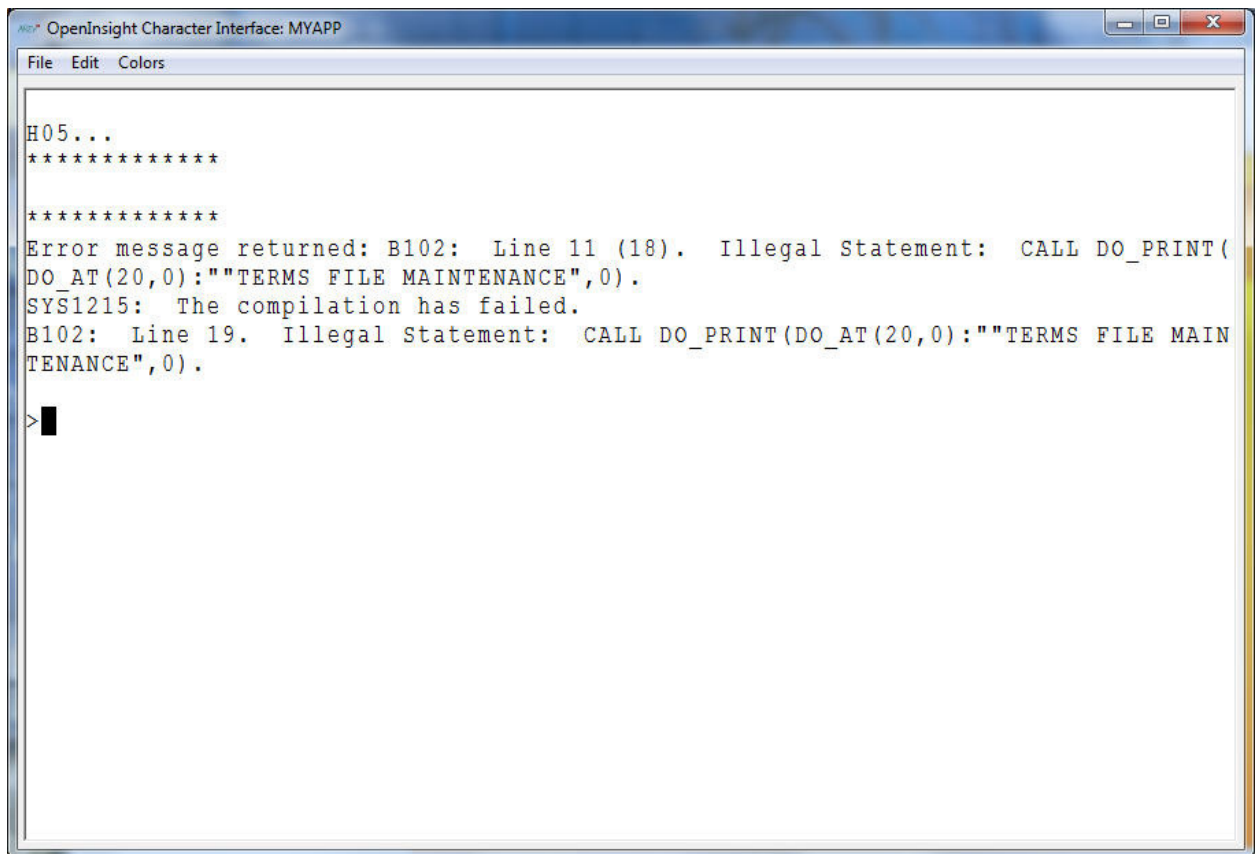
```

BP_H05
Top
.L20
00001 *#ADDED 1,2,3,4,5,6
00002 Declare Function READ_LOCK
00003 *#ORIGNAME H05
00004 *#SOURCE BP
00005 *#CTO
00006 *#Precompile
00007 *#NAME H05
00008 * TERMS FILE MAINTENANCE
00009 * R. CATALANO - WINWIN SOLUTIONS, INC.
00010 * 08/31/03
00011 Call DO_PROMPT('')
00012 CTO%CLEAR="";ENTLINE="";HAF="";AREC="";CSW="";ASW="";AC="";ID="";CTO%WAIT4
LOCK="";CTO%NTFND="";OP="";DESC="";PCT="";DD="";LNUM="";CTO%KPLK=""
00013 ENTLINE=DO_AT(0,22):SPACE(79):DO_AT(0,22)
00014 OPEN '','H-TERMS' TO HAF ELSE STOP
00015 *
00016 LBL 003: AREC='';CSW=0;ASW=0
00017 CALL DO_PRINT(DO_AT(-1),0)
00018 CALL DO_PRINT(DO_AT(20,0):"TERMS FILE MAINTENANCE",0)
00019 CALL DO_PRINT(DO_AT(4,2):"CODE",0)
00020 CALL DO_PRINT(DO_AT(1,4):"1. DESCRIPTION",0)
.

```

21. Program errors are reported by the compiler. The OpenInsight compiler compiles and reports errors on the source code found in the table `OI_PRECOMPILED_CODE`.

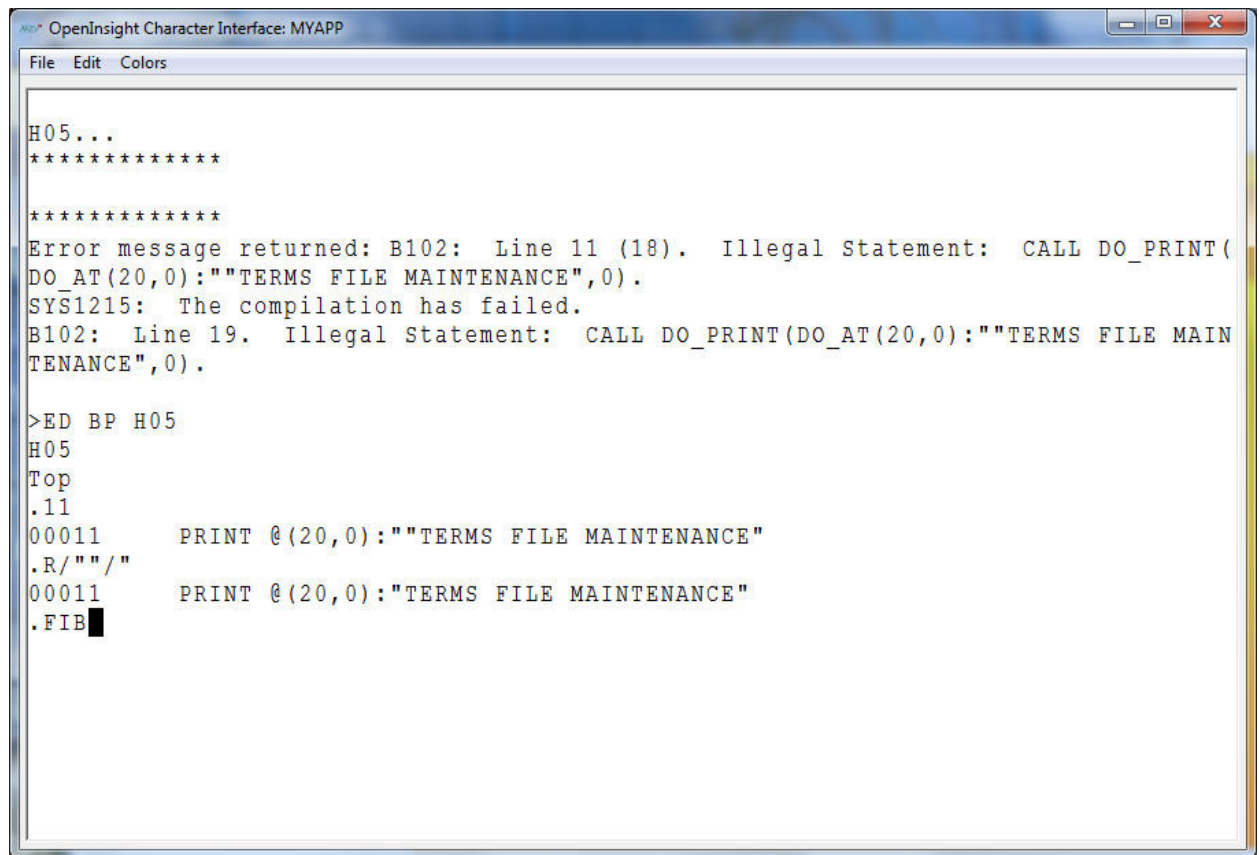
22. In the example below an error has been found on line (17) of the source code item BP_H05 in the table OI_PRECOMPILED_CODE.



The screenshot shows a window titled "OpenInsight Character Interface: MYAPP". The window has a menu bar with "File", "Edit", and "Colors". The main text area displays the following content:

```
H05...  
*****  
  
*****  
Error message returned: B102: Line 11 (18). Illegal Statement: CALL DO_PRINT(  
DO_AT(20,0):"TERMS FILE MAINTENANCE",0).  
SYS1215: The compilation has failed.  
B102: Line 19. Illegal Statement: CALL DO_PRINT(DO_AT(20,0):"TERMS FILE MAIN  
TENANCE",0).  
  
> █
```


23. Please note that, although the line number reported refers to the line in the pre-processed version of the source code, program modifications **MUST** be made to your original source code and then re-compiled.



```

OpenInsight Character Interface: MYAPP
File Edit Colors

H05..
*****

*****
Error message returned: B102: Line 11 (18). Illegal Statement: CALL DO_PRINT(
DO_AT(20,0):"TERMS FILE MAINTENANCE",0).
SYS1215: The compilation has failed.
B102: Line 19. Illegal Statement: CALL DO_PRINT(DO_AT(20,0):"TERMS FILE MAIN
TENANCE",0).

>ED BP H05
H05
Top
.11
00011 PRINT @(20,0):"TERMS FILE MAINTENANCE"
.R/""/"
00011 PRINT @(20,0):"TERMS FILE MAINTENANCE"
.FIB

```

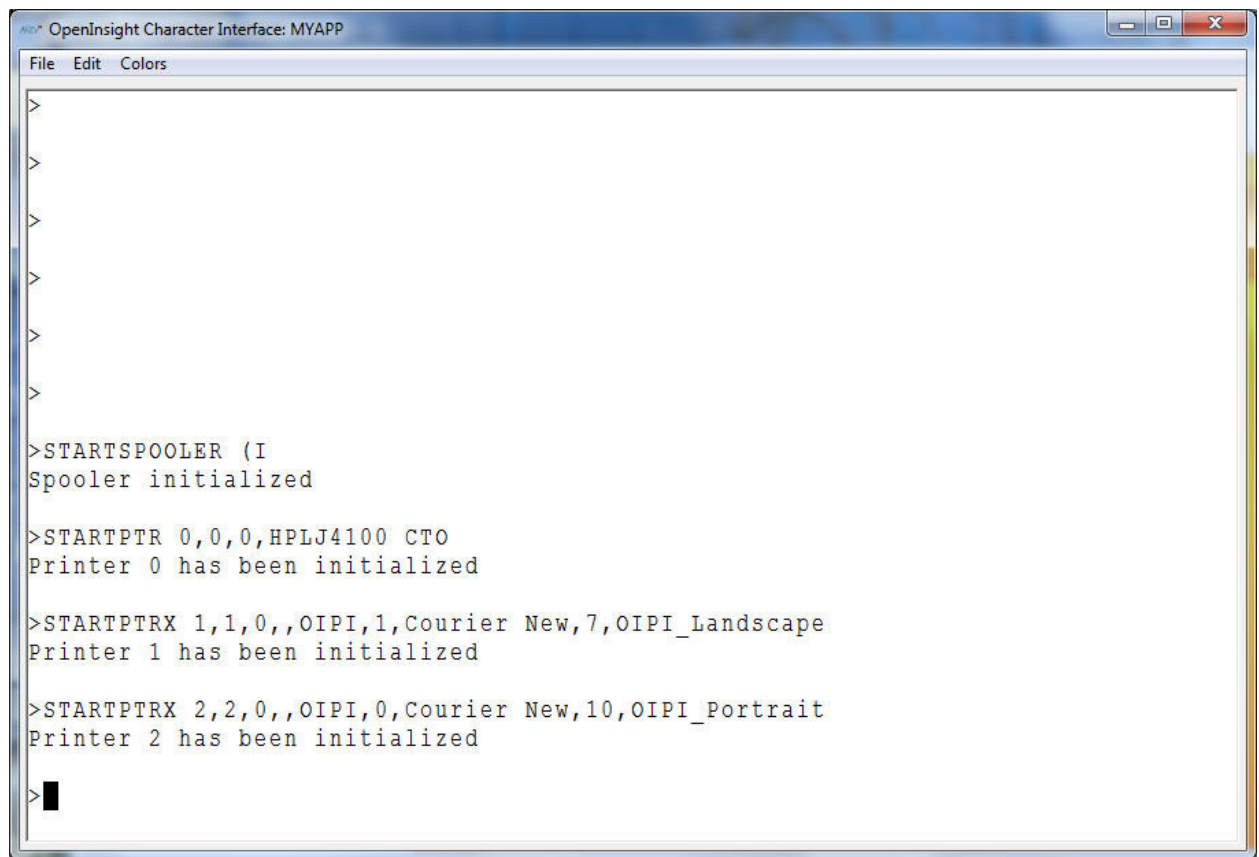
VI. Defining printers and managing the SPOOLER

24. All printers that are defined on the server where the OEngineServer resides are available for use with the CTO.
25. The **STARTPTR** & **STARTPTRX** commands are used to assign a printer to a form queue for use with the CTO. Using the **STARTPTR** command will define this printer for output using DirectPrint; all control codes, escape sequences, etc. will be passed "as is" to the printer. Using the **STARTPTRX** command allows you to define additional features, including a subroutine that will be invoked when a print job begins and ends, and/or the use of OIPI for output generation. To assign your printer to form queue 0 using DirectPrint type:

STARTPTR 0,0,0,Windows_Printer_Name

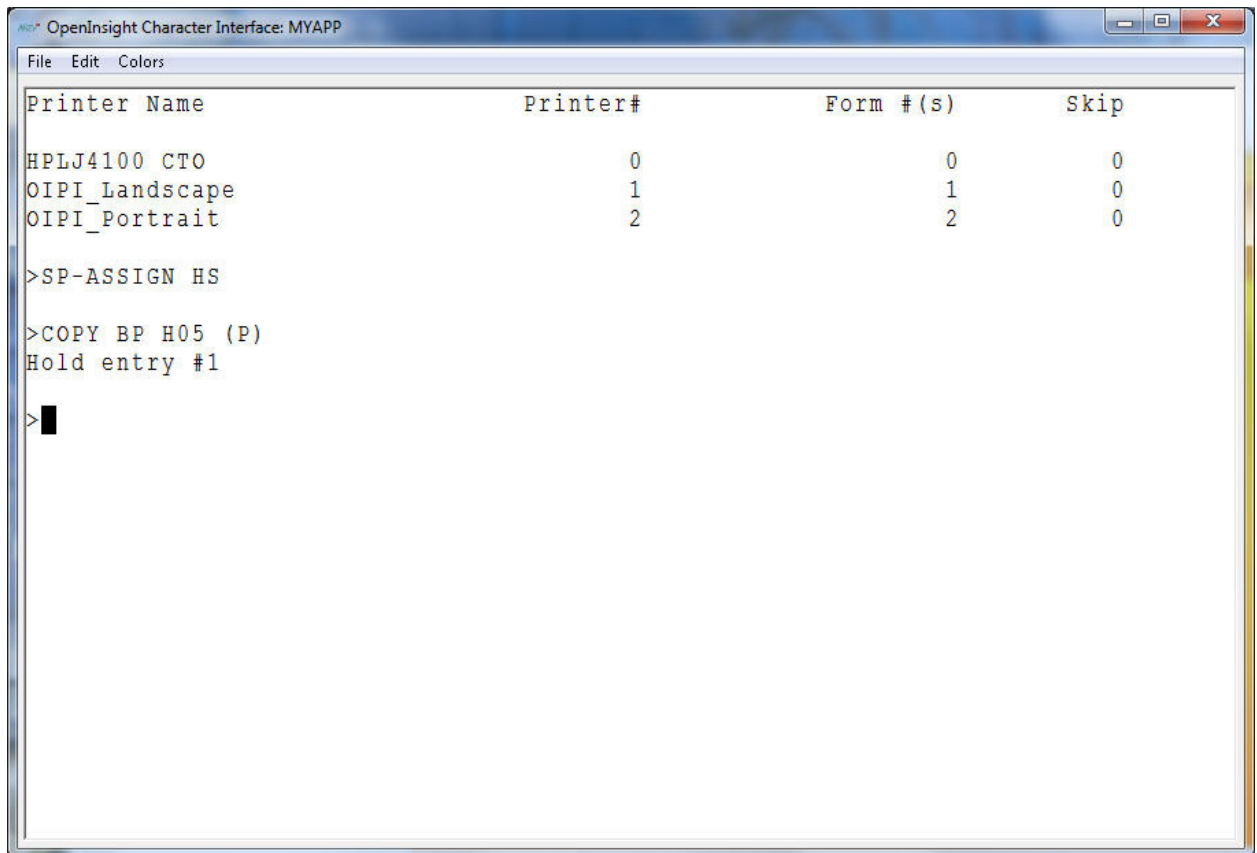
To assign your printer to form queue 1 using OIPI type:

STARTPTRX 1,1,0,,OIPI,1,Courier New,7,OIPI_Landscape

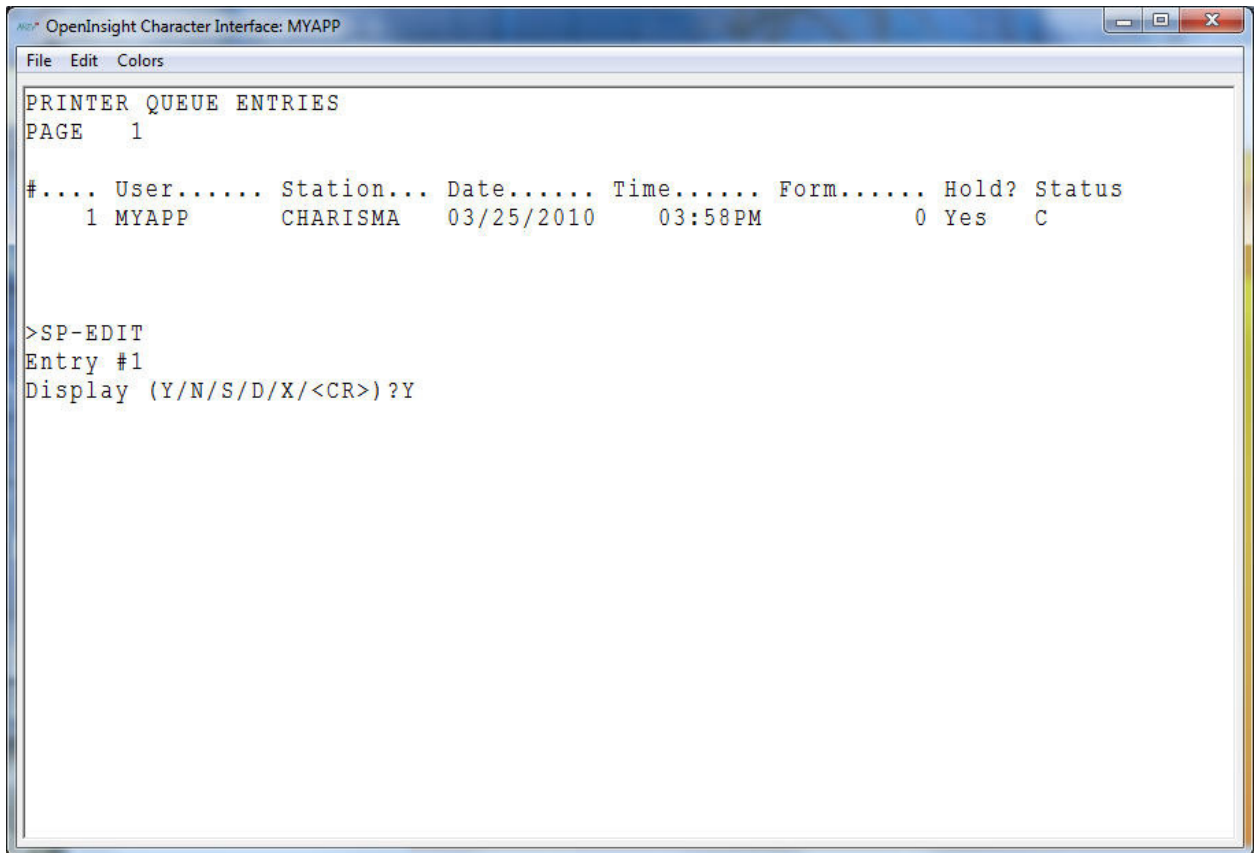


```
OpenInsight Character Interface: MYAPP
File Edit Colors
>
>
>
>
>
>
>
>STARTSPOOLER (I
Spooler initialized
>STARTPTR 0,0,0,HPLJ4100 CTO
Printer 0 has been initialized
>STARTPTRX 1,1,0,,OIPI,1,Courier New,7,OIPI_Landscape
Printer 1 has been initialized
>STARTPTRX 2,2,0,,OIPI,0,Courier New,10,OIPI_Portrait
Printer 2 has been initialized
> █
```


26. The **LISTPTR** command is used to list all printers defined for use with the CTO.
27. Output can be assigned to either a physical form queue or a hold file. To generate output for a hold file, specify “HS” as the output destination in the **SP-ASSIGN** command.



28. Output sent to a hold file can be retrieved using the **SP-EDIT** command. **LISTPEQS** will display all items in the CTO spooler. Output to an OIPI defined printer will be displayed to the print preview window.



The screenshot shows a window titled "OpenInsight Character Interface: MYAPP". The window contains a text area with the following text:

```
PRINTER QUEUE ENTRIES
PAGE      1

#.... User..... Station... Date..... Time..... Form..... Hold? Status
   1 MYAPP      CHARISMA   03/25/2010    03:58PM          0 Yes    C

>SP-EDIT
Entry #1
Display (Y/N/S/D/X/<CR>) ?Y
```

OpenInSight Character Interface: MYAPP

File Edit Colors

>SP-EDIT

Entry #1

Display (Y/N/S/D/X/<CR>) ? Y

BP H05

Page 1

H05

```

00001 * TERMS FILE MAINTENANCE
00002 * R. CATALANO - WINWIN SOLUTIONS, INC.
00003 * 08/31/03
00004     PROMPT ''
00005     CLEAR
00006     ENTLINE=@(0,22):SPACE(79):@(0,22)
00007     OPEN '', 'H-TERMS' TO HAF ELSE STOP
00008 *
00009 003 AREC=''; CSW=0; ASW=0
00010     PRINT @(-1)
00011     PRINT @(20,0):"TERMS FILE MAINTENANCE"
00012     PRINT @(4,2):"CODE"
00013     PRINT @(1,4):"1. DESCRIPTION"
00014     PRINT
Spool (Y/N=<cr>/T/TN/F) ?
Delete (Y/N=<cr>) : ?
End of requested files
>

```

OpenInSight Character Interface: MYAP Captured Output - 16:10:39 25 MAR 2010

File Edit Colors

Pages

Page 1

Page 2

Page 3

>TERM 80,25,132,60

>SP-ASSIGN F2

>COPY BP H05 (P)

Entry #2

BP H05

H05

00001 * TERMS FILE MAINTENANCE

00002 * R. CATALANO - WINWIN SOLUTIONS, INC.

00003 * 08/31/03

00004 PROMPT ''

00005 CLEAR

00006 ENTLINE=@(0,22):SPACE(79):@(0,22)

00007 OPEN '', 'H-TERMS' TO HAF ELSE STOP

00008 *

00009 003 AREC=''; CSW=0; ASW=0

00010 PRINT @(-1)

00011 PRINT @(20,0):"TERMS FILE MAINTENANCE"

00012 PRINT @(4,2):"CODE"

00013 PRINT @(1,4):"1. DESCRIPTION"

00014 PRINT @(1,5):"2. PERCENTAGE"

00015 PRINT @(1,6):"3. DAYS DUE"

00016 *

00017 005 PRINT ENTLINE:"ENTER TERMS CODE OR 'E' 1

00018 PRINT @(20,2):"*** ":@(20):

00019 INPUT AC

00020 IF AC='E' THEN GO 999

00021 IF LEN(AC)>2 THEN GO 005

00022 AC=STR('0',2-LEN(AC)):AC

00023 IF AC MATCHES '2N' THEN NULL ELSE GO 005

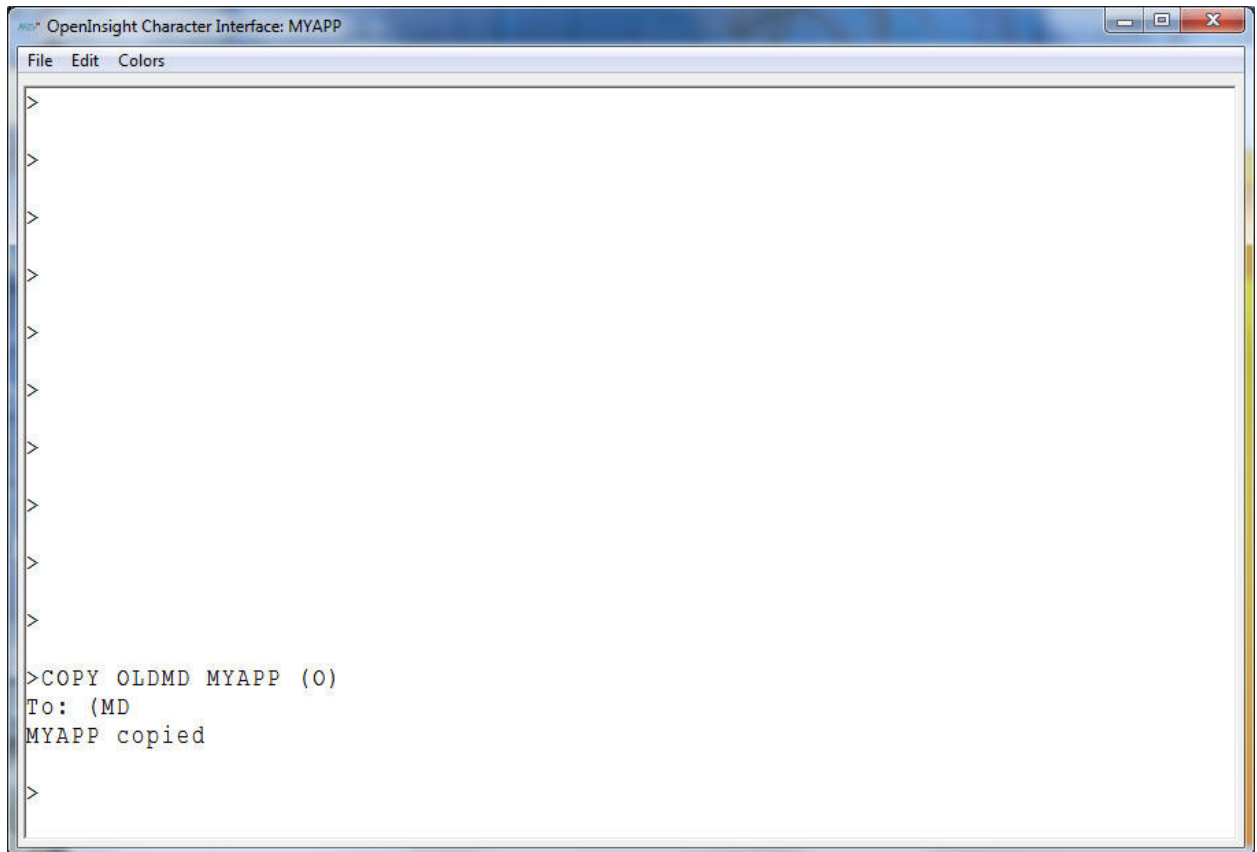
00024 PRINT @(20,2):AC

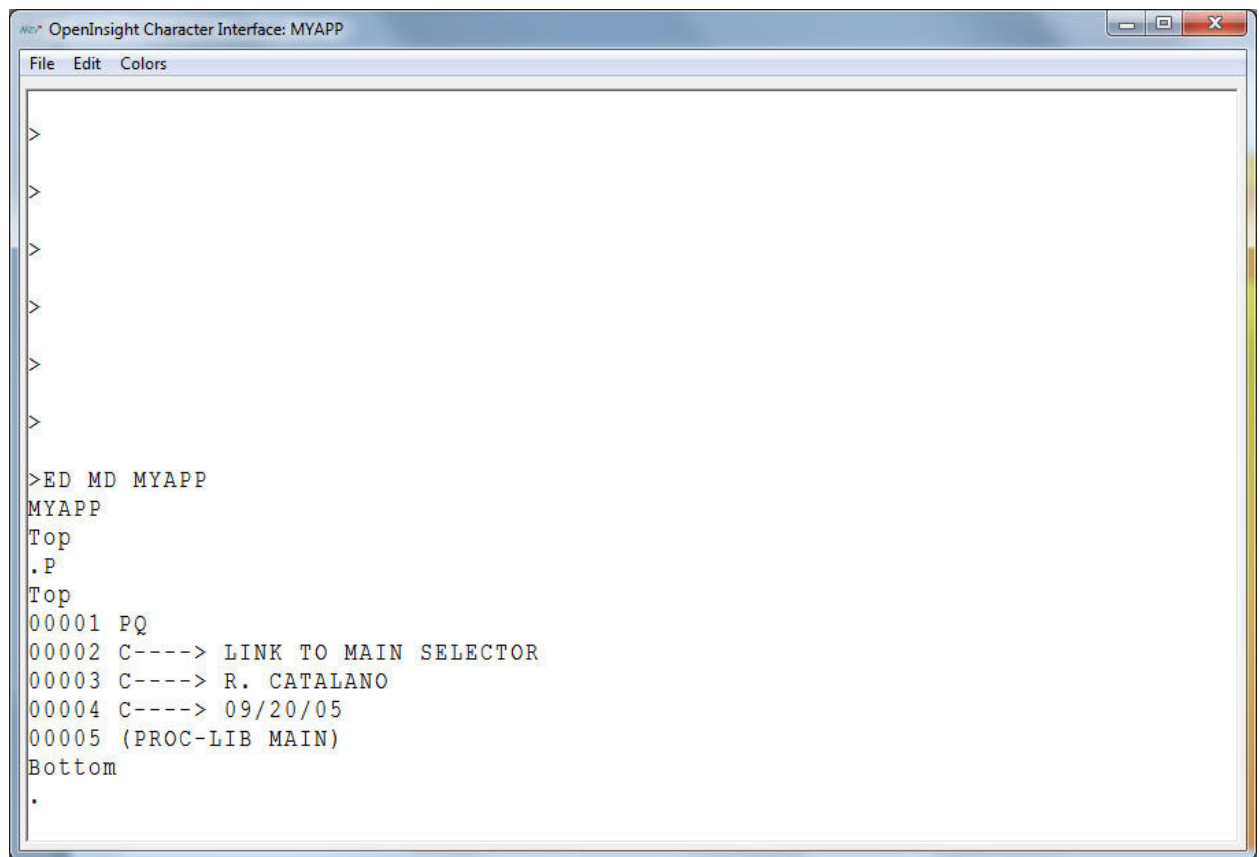
00025 *

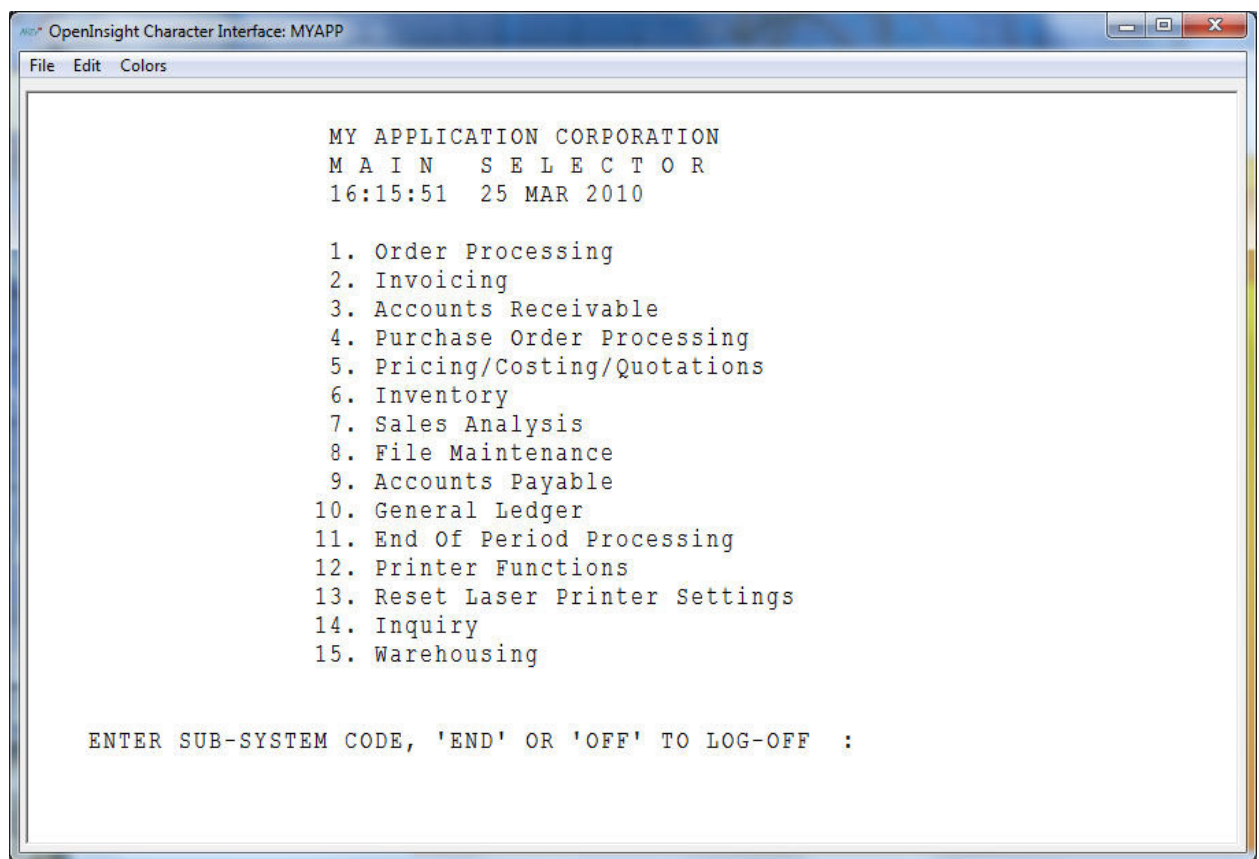
00026 015 ID=AC

VII. Launch your application in CTO

29. To automatically launch your application from logon, create a **PROC** in the **MD** with your **APPLICATION_NAME** or **USER_NAME** as the key. This PROC may execute another PROC or BASIC program.





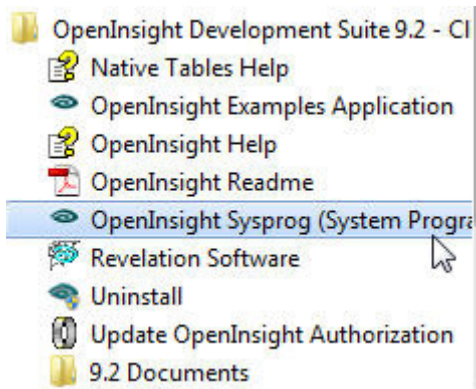


30. To exit an application and the CTO interface, you must log **OFF** the application and **QUIT** the CTO.

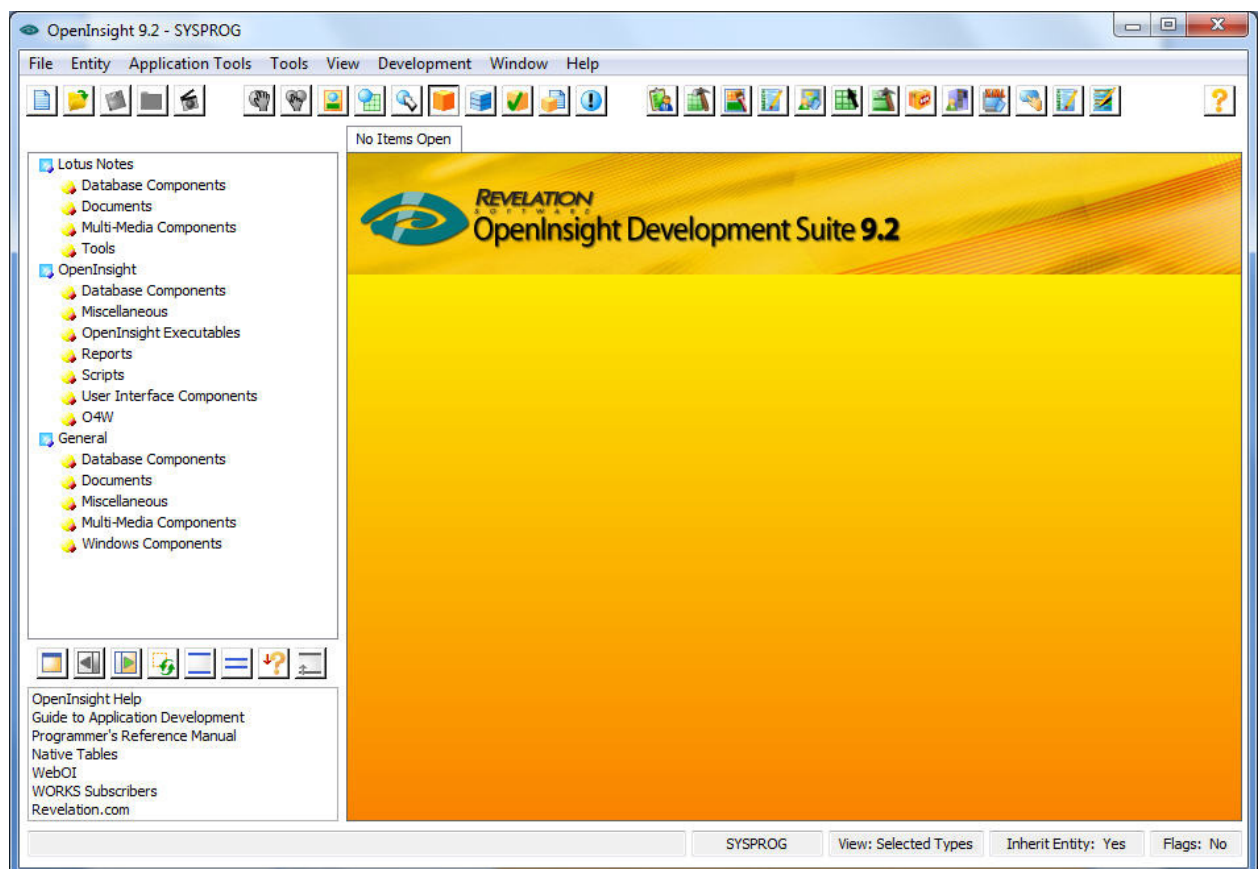
Starting OpenInsight

I. Starting OpenInsight

1. Launch OpenInsight.
2. Start, Programs, OpenInsight Development Suite, OpenInsight Sysprog.

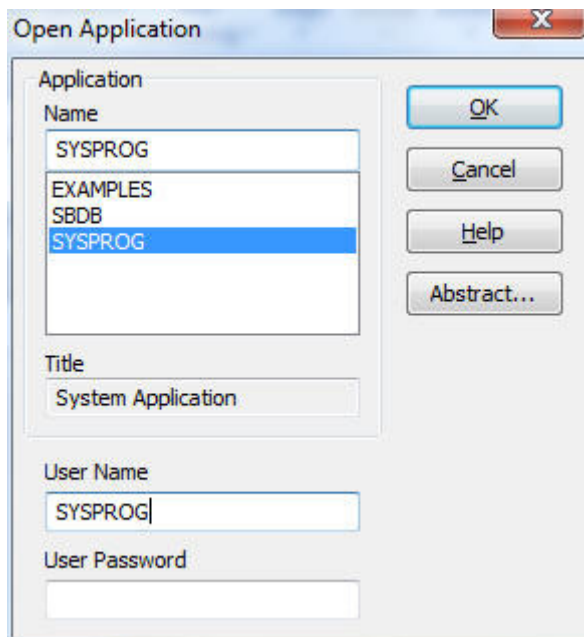


4. The Application Manager window for the SYSPROG application will be displayed as shown below.



II. Opening an Existing Application

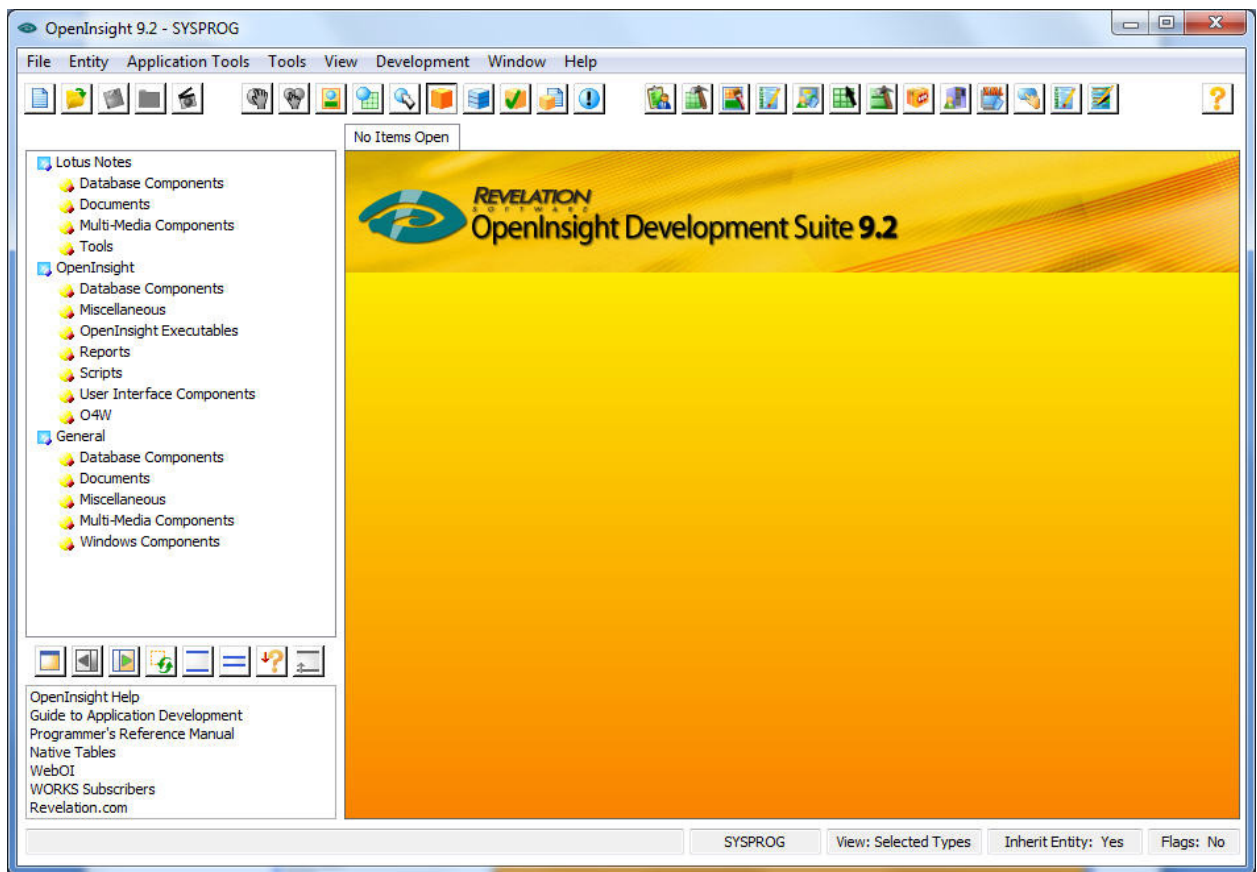
5. Choose *Open Application* from the *File* menu. The *Open Application* dialog box will be displayed.



5. Choose the SYSPROG application from the *Name* list box.
6. Click in the *User Name* edit line and type SYSPROG.
7. Click the OK button to open the SYSPROG application.

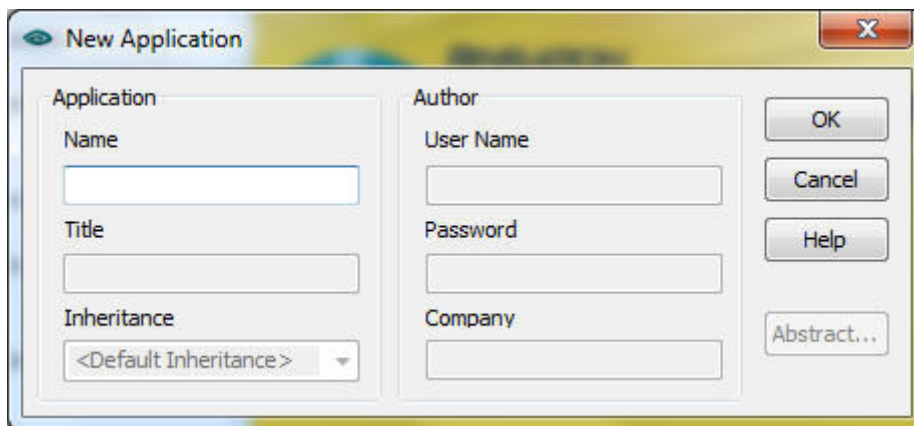
Note: To open any application provided with OpenInsight, enter the *Application Name* in the *User Name* edit line. Passwords are not required for sample applications.

The *Application Manager* window for the SYSPROG application will be displayed, as shown below.



III. Creating a New Application

8. Choose *Application, New Application* from the *File* menu to create a new application. The *New Application* dialog box is displayed.

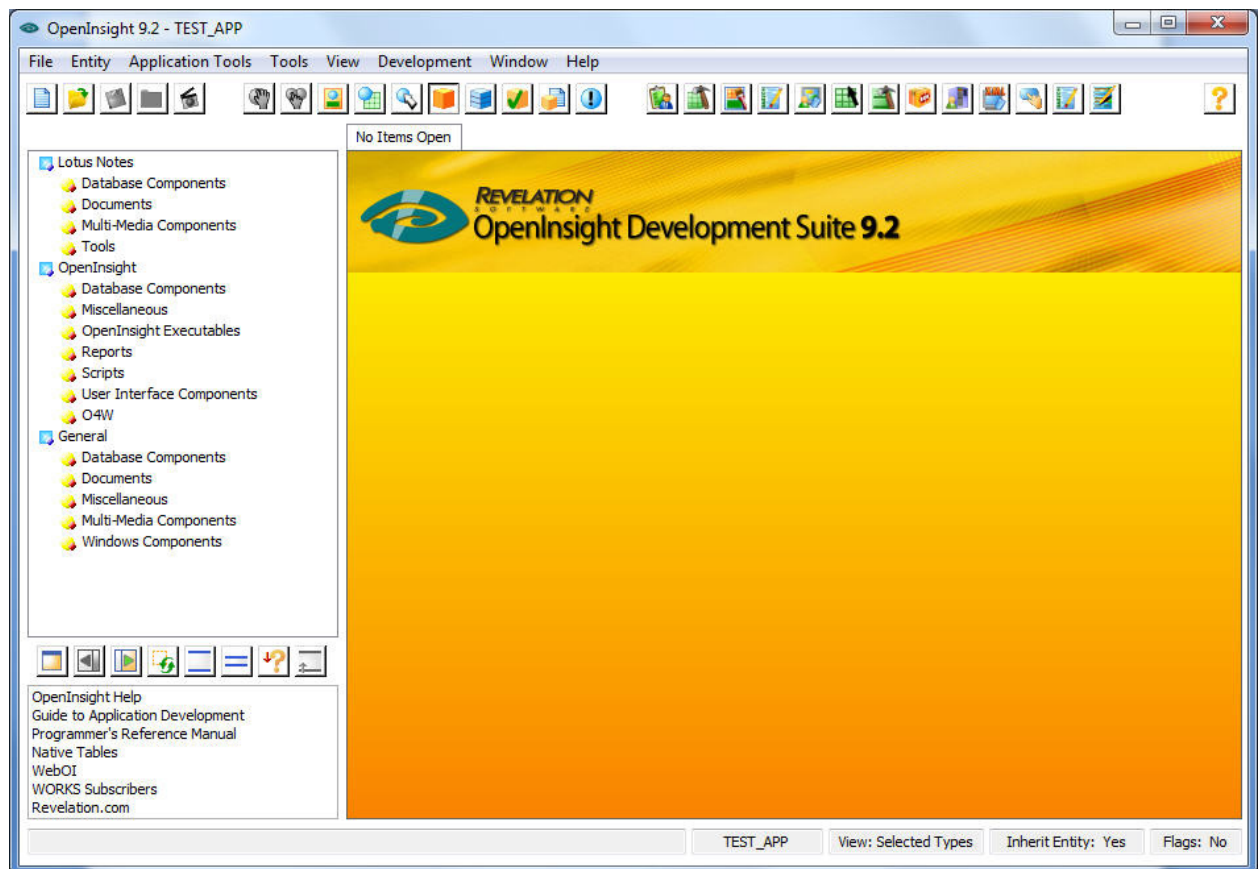


The *Application Name* and *Author User Name* are the only required items in this dialog box.

9. Click in the *Name* edit line and type TEST_APP. (Spaces are not allowed in an *Application Name* or *User Name*.)
10. Click in the *User Name* edit line and type TEST_APP.

Note: The *Inheritance* drop down indicates <Default Inheritance>. This means that this newly created application inherits the components of the base application, SYSPROG, mentioned above.

11. Click the OK button to create the new TEST_APP application.
The *Application Manager* window for the TEST_APP application will be displayed, as shown below.



Working with Linear Hash Data

I. Linear Hash Data

OpenInsight comes complete with its own proprietary database, known as Linear Hash. It is a Multivalue, variable-length database. Multivalue means that there may be more than one value in a field, but you don't need to have separate fields.

For example, in many databases you would have a field named **Address1**, and probably **Address2**, and maybe **Address3**. So, if someone had an address of *123 Main Street, Apt 2A*, you would put *123 Main Street* into the **Address1** field, and the *Apt 2A* into the **Address2** field. **Address3** would be empty.

In a multivalue database, such as used by OpenInsight, you would put all the data into the same field, called **Address**. If there was one value, it would be stored, and there would be no wasted space with empty fields. If there were two or more values, they would be stored, and there would be no wasted space or fields. We do this by using delimiters behind the scenes called Value Marks.

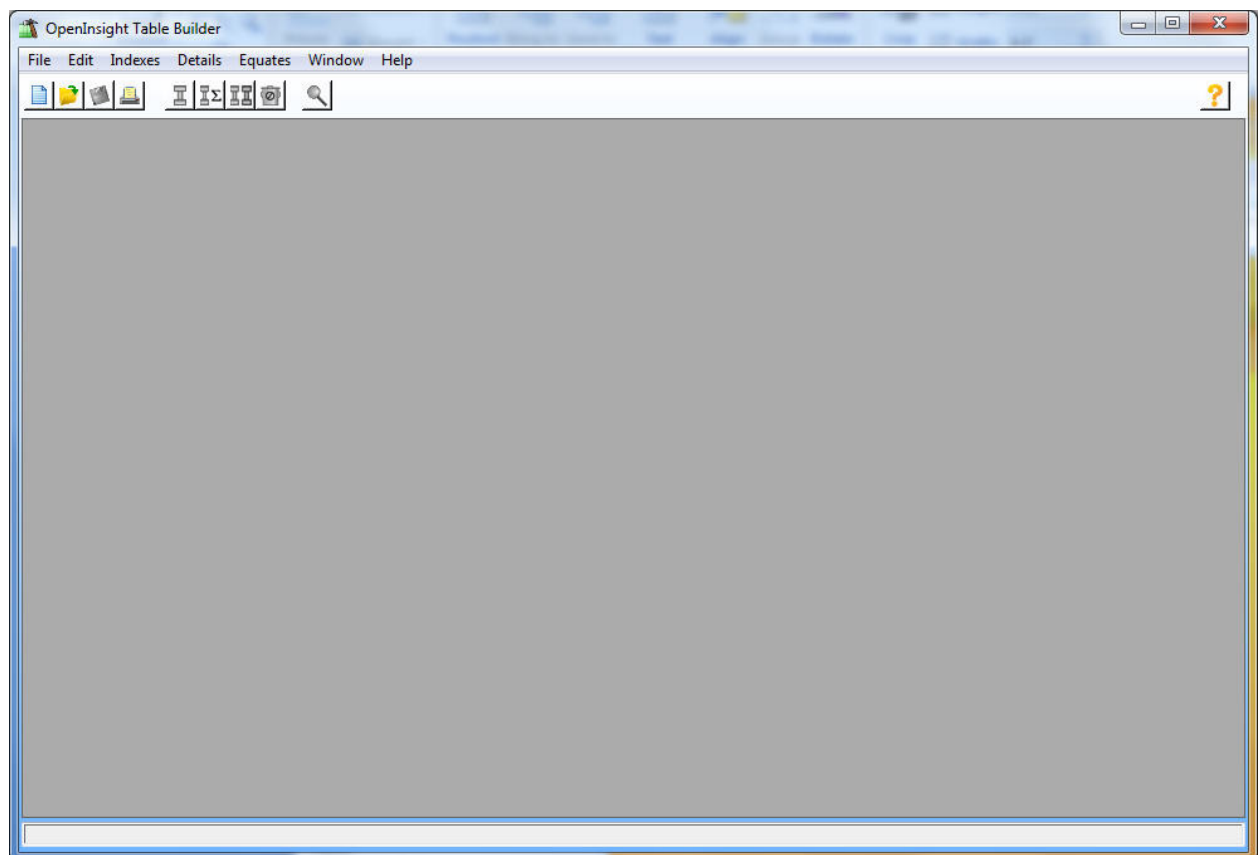
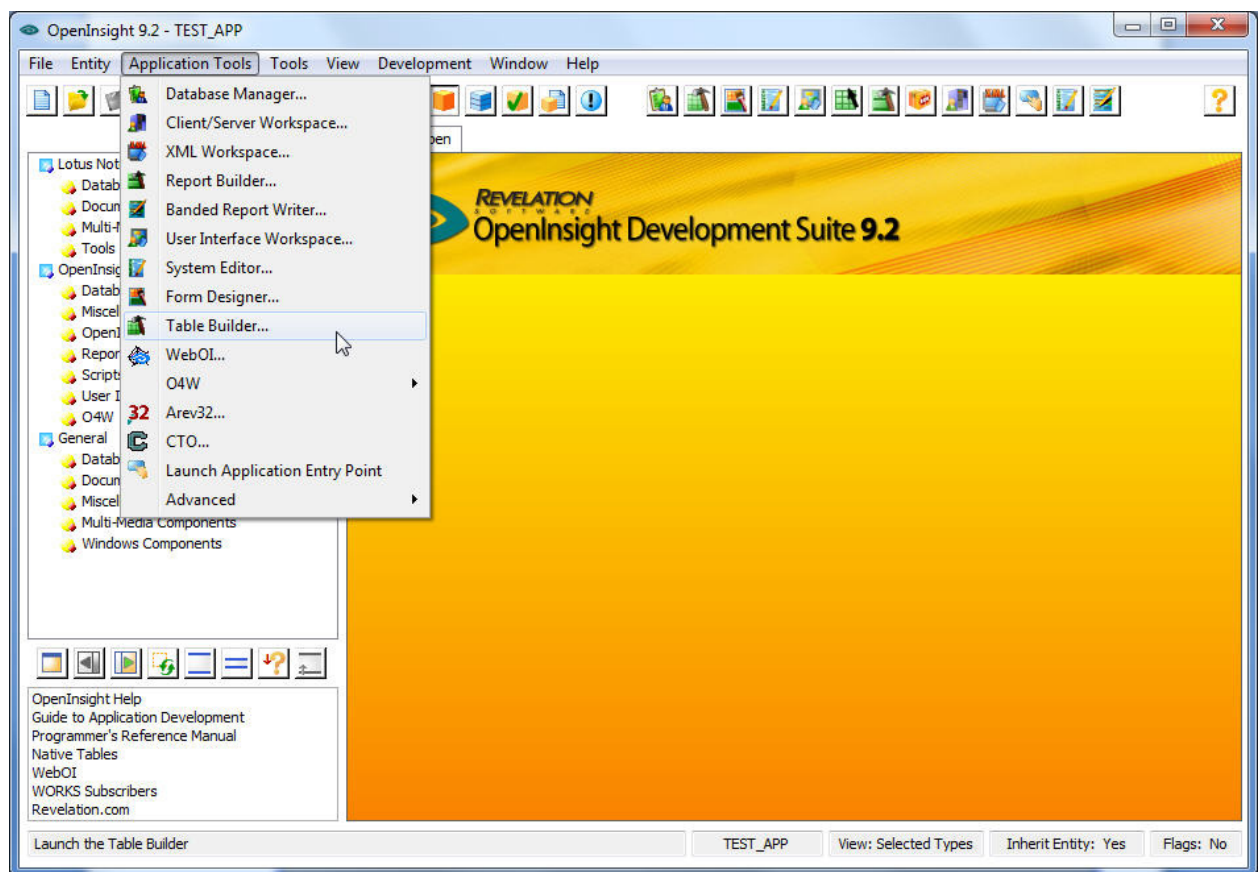
Variable-length means that OpenInsight doesn't have fixed data widths for its data. If a field contains data that is 17 characters long, we use only 17 characters plus a delimiter called a Field Mark. If a field contains data that is only 1 character long, we use only 1 character plus the field mark.

These attributes make OpenInsight a very flexible, efficient database. Once a table is defined, you can add as many new fields as you want without redefining the database. The tables created are always self-resizing, so they are always efficient as possible.

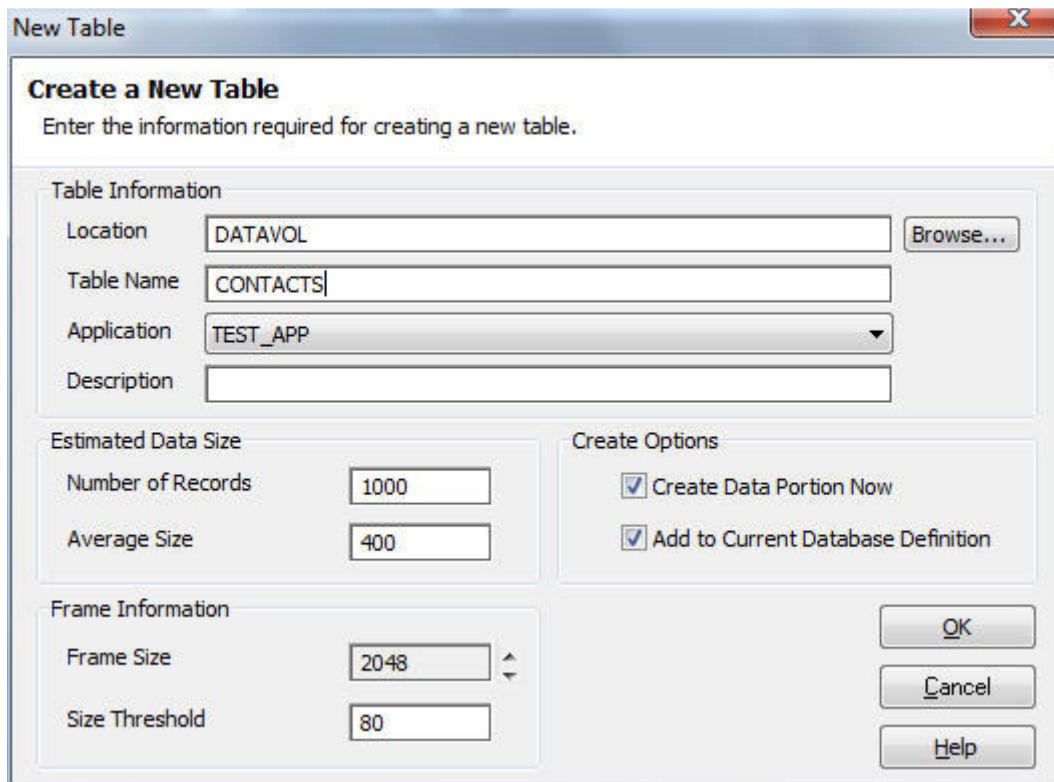
II. Creating a Data Table

1. Choose *Table Builder* from the *Application Tools* menu or press the Table Builder button





2. Choose New... from the File menu or click on the New button. The New Table dialog box will be displayed.



The image shows a 'New Table' dialog box with a title bar and a close button. The main area is titled 'Create a New Table' and contains the instruction 'Enter the information required for creating a new table.' The dialog is organized into several sections: 'Table Information' with fields for Location (DATAVOL), Table Name (CONTACTS), Application (TEST_APP), and Description; 'Estimated Data Size' with fields for Number of Records (1000) and Average Size (400); 'Frame Information' with fields for Frame Size (2048) and Size Threshold (80); and 'Create Options' with two checked checkboxes: 'Create Data Portion Now' and 'Add to Current Database Definition'. A 'Browse...' button is next to the Location field. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

New Table

Create a New Table
Enter the information required for creating a new table.

Table Information

Location: DATAVOL

Table Name: CONTACTS

Application: TEST_APP

Description:

Estimated Data Size

Number of Records: 1000

Average Size: 400

Frame Information

Frame Size: 2048

Size Threshold: 80

Create Options

☒ Create Data Portion Now

☒ Add to Current Database Definition

3. Choose a Location. The Browse button will allow you to choose the location.
4. Enter the name of the new table in the *Table Name* edit line.
5. Click on the OK button. The Table Builder window will be displayed as shown below.

OpenInsight Table Builder

File Edit Indexes Details Equates Window Help

Table Builder <CONTACTS>

Table Builder
Use this screen for creating or modifying dictionary column definitions.xx

Data Columns

Position	Column Name	Data Type	Key	Not Null	Default	S/M	Len	Just	AMV

Calculated Columns

Column Name	Data Type	Formula	Just	AMV

Group Columns

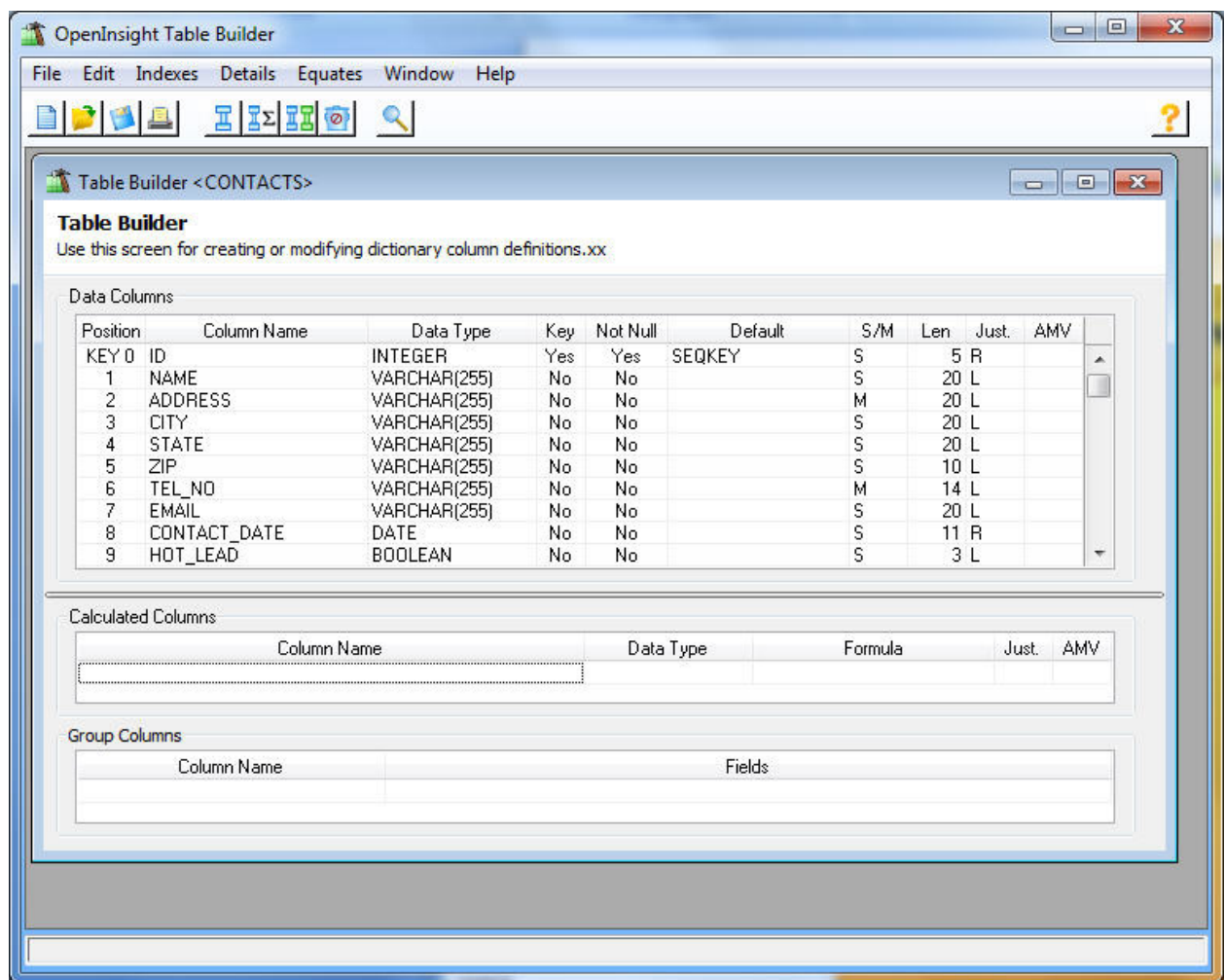
Column Name	Fields

III. Adding Data Columns to a Data Table

6. Notice that the first row of the Data Columns edit table is selected and the cursor is positioned in the Column Name column. This is usually your key column, but it is not required to be so. Type **ID** in the first row under Column Name. Case is not important as the Table Builder converts all column names to uppercase.
7. Tab to the next column, Data Type. A list of available data types is available by right-clicking on the Data Type column. Type or select **INTEGER**.
8. Tab to the next column, Key. As this is the key column for CONTACTS, type **Y** or **Yes** and tab to the next column. The Key column defaults to No. when tabbing off the column.
9. The Not Null column accepts of value of Yes or No. As this is the key column for CONTACTS, type **Y** or **Yes** and tab to the next column. The Not Null column defaults to No. when tabbing off the column.
10. The Default column allows entering of the default value for this column. Type **SEQKEY** for the ID column's default. SEQKEY provides a sequential counter which is incremented upon saving a record.
11. The SM Column indicates whether this column is a Single valued or MultiValue field. For a Key this should always be **S**.
12. The Len column indicates the default length that will be displayed for this column. It does not limit the amount of data stored in the field. Enter a **5** in this field.
13. Tab again to enter the row in the dictionary. All text is converted to uppercase and Table Builder assigns a position for this column in the record. The ID column has been assigned the position **KEY 1** meaning that it is the first record field's
14. Continue using Table Builder to enter the following data columns in the data dictionary. Notice that no additional columns specify a Default value.

Column Name	Data Type	Key	Not Null	S/M	Len
NAME	VARCHAR(255)	N	N	S	20
ADDRESS	VARCHAR(255)	N	N	M	20
CITY	VARCHAR(255)	N	N	S	20
STATE	VARCHAR(255)	N	N	S	2
ZIP	VARCHAR(255)	N	N	S	10
TEL_NO	VARCHAR(255)	N	N	S	14
EMAIL	VARCHAR(255)	N	N	S	20
CONTACT_DATE	DATE	N	N	S	11
HOT_LEAD	BOOLEAN	N	N	S	3

Your Table Builder should look like the figure below.



- Double-click on the TEL_NO row to display the Column Attributes dialog box where you specify additional dictionary information.

Edit Data Column Attributes - <TEL_NO>

Data Column Attributes

Use this screen for editing the attributes of a particular Data Column.

Domain DataType: VARCHAR(255) Default: <input type="checkbox"/> Not Null <input checked="" type="checkbox"/> Multivalued <input type="checkbox"/> Master MV Groupname:	Position <input type="checkbox"/> Key Position: 6 <input checked="" type="checkbox"/> Master
Data Validation <i>Double-Click for Options</i> Input: Output:	Display <input checked="" type="radio"/> Left <input type="radio"/> Center <input type="radio"/> Right <input type="radio"/> Text Length: 14 Indexes <input type="checkbox"/> Btree <input type="checkbox"/> Xref <input type="checkbox"/> Relational <input type="checkbox"/> Bitmap
Description of Column Usage <div style="border: 1px solid black; height: 100px;"></div>	

The Column Attributes dialog box is used to provide additional information regarding the data field, such as data validation, display format and column description. The Column Attributes dialog box is also where you set a column to be multivalued. Check the Multivalued check box for the TEL_NO column.

Edit Data Column Attributes - <TEL_NO>

Data Column Attributes
Use this screen for editing the attributes of a particular Data Column.

Domain
DataType: VARCHAR(255)
Default:
☐ Not Null ☒ Multivalued ☐ Master
MV Groupname:
Data Validation
Double-Click for Options
Input: [PHONE_FORMAT]
Output: [PHONE_FORMAT]
Position
☐ Key Position: 6 ☒ Master
Display
☒ Left ☐ Center ☐ Right ☐ Text Length: 14
Indexes
☒ Btree ☒ Xref ☐ Relational ☐ Bitmap
Column Heading (For Reports)
Tel No
OK Cancel Help
Description of Column Usage

Double-click on the Input area and choose [PHONE_FORMAT].

Also click the drop down list on the Data Validation, Output and select PHONE_FORMAT as the Output pattern. This format causes telephone numbers to be displayed as (NNN) NNN-NNNN. Click the OK button.

IV. Adding Calculated Columns to a Data Table

16. Double-click in the first row of the Calculated Columns edit table at the bottom of the Table Builder window. This will display the Calculated Column Attributes window.
17. Type NEXT_CONTACT in Name field.
18. Select DATE from the DataType drop down list.
19. Select D from the Output drop down list. Type D4/ to make it more readable for the user.
20. In the Edit Formula text box type: @ANS = {CONTACT_DATE} + 7. Clicking on the Test button will check for syntax errors within the formula.

Note: @ANS is a system variable used to store the value of a calculated column. Dates are stored internally as integers. In the above formula we add 7 days to the Contact_Date. By applying the D output type the result will display as a date and not an integer.

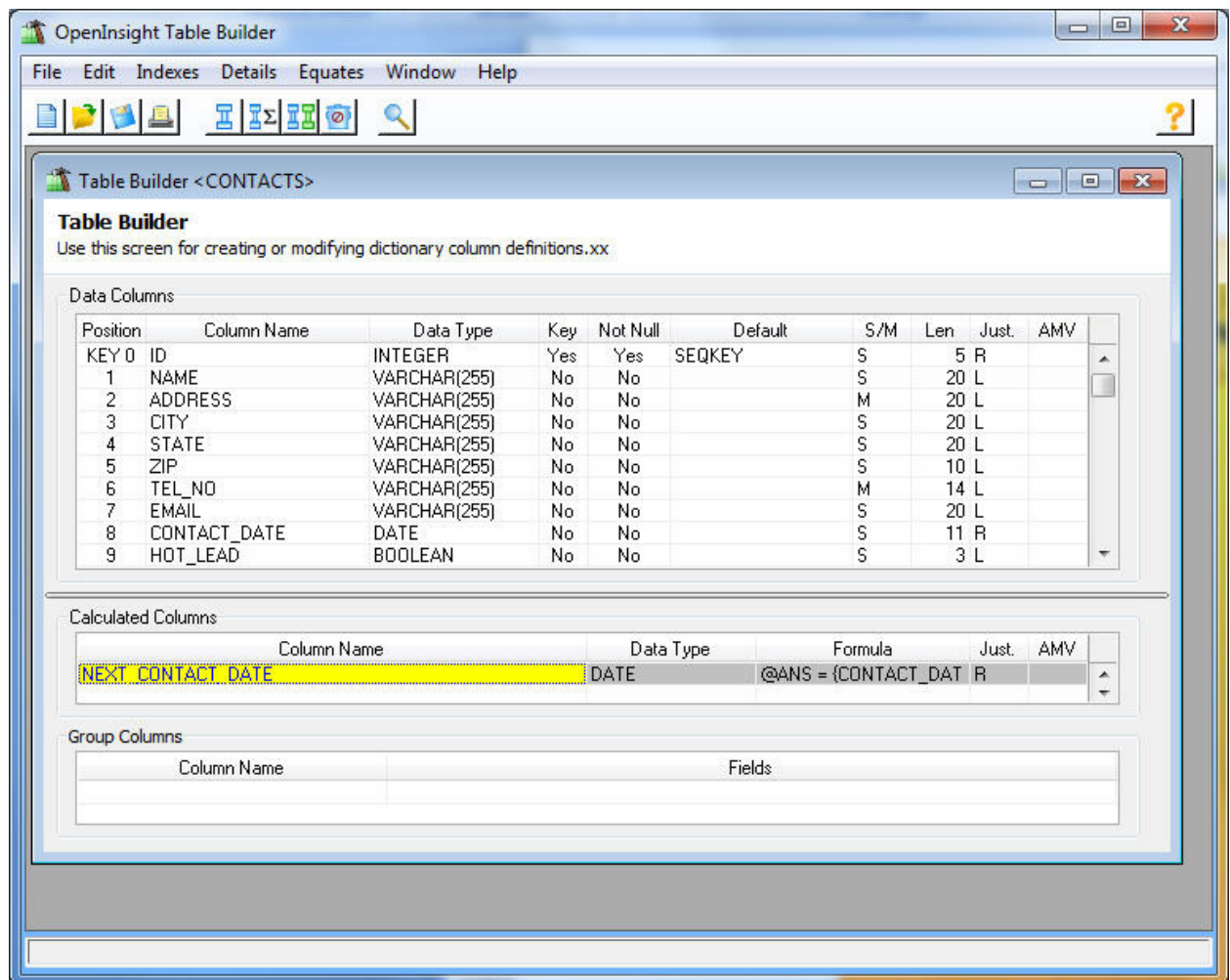
The screenshot shows the 'Edit Calculated Column Attributes' dialog box for the column 'NEXT_CONTACT_DATE'. The dialog has a title bar with a close button (X). The main area is divided into several sections:

- Details:** Contains fields for Name (NEXT_CONTACT_DATE), DataType (DATE), Output (D4/), and checkboxes for Multivalued and Master. There is also an MV Groupname field.
- Display:** Contains radio buttons for Left, Center, Right (selected), and Text. A Length field is set to 11.
- Column Heading:** A list box containing the column name 'NEXT_CONTACT_DATE'.
- Indexes:** Checkboxes for Btree, Xref, Relational, and Bitmap.
- Edit Formula:** A large text area containing the formula '@ANS = {CONTACT_DATE} + 7'.
- Description of Column Usage:** A large empty text area at the bottom.

Buttons for OK, Cancel, and Help are located on the right side of the dialog. There are also buttons for 'Test Symbolic' and 'Formula Wizard'.


21. Click the OK button.

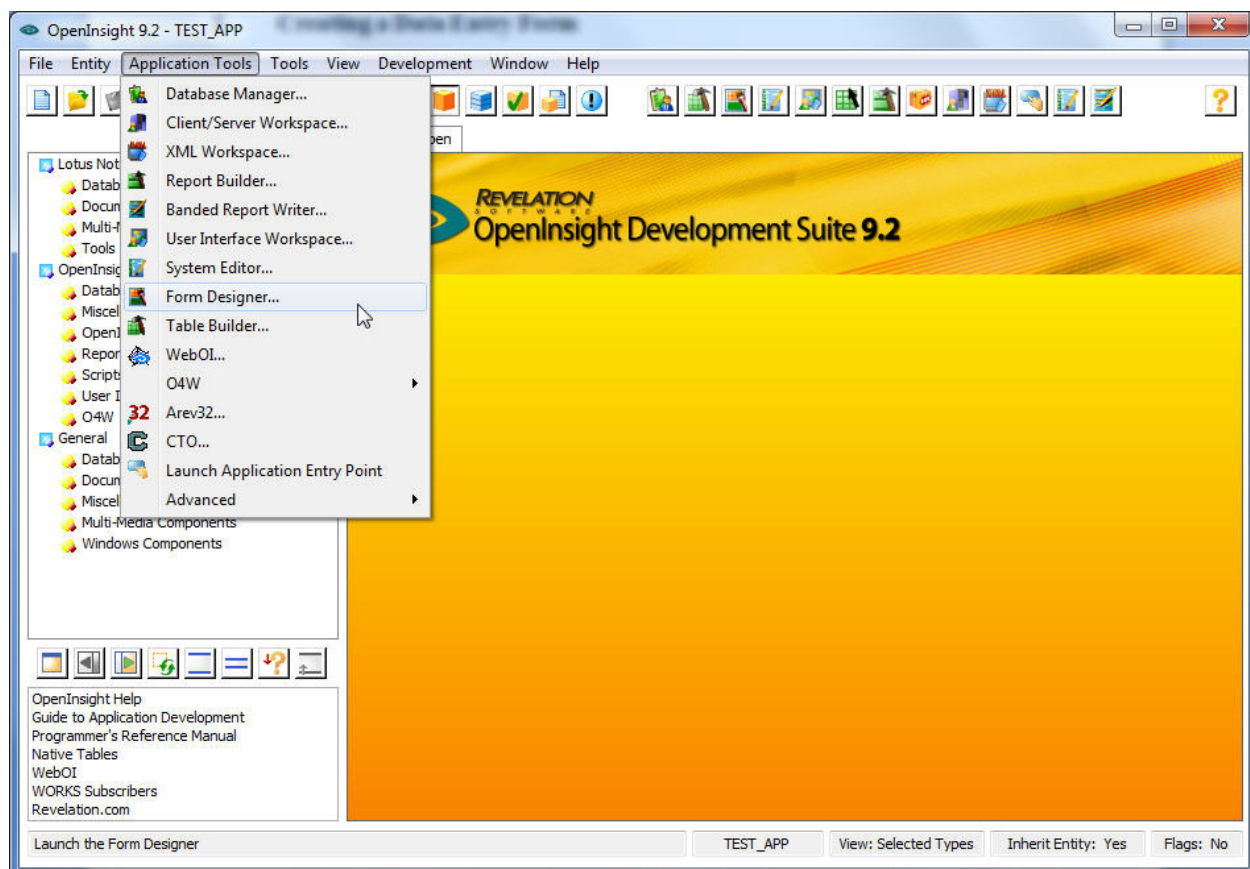
Your Table Builder should now look like the figure below.

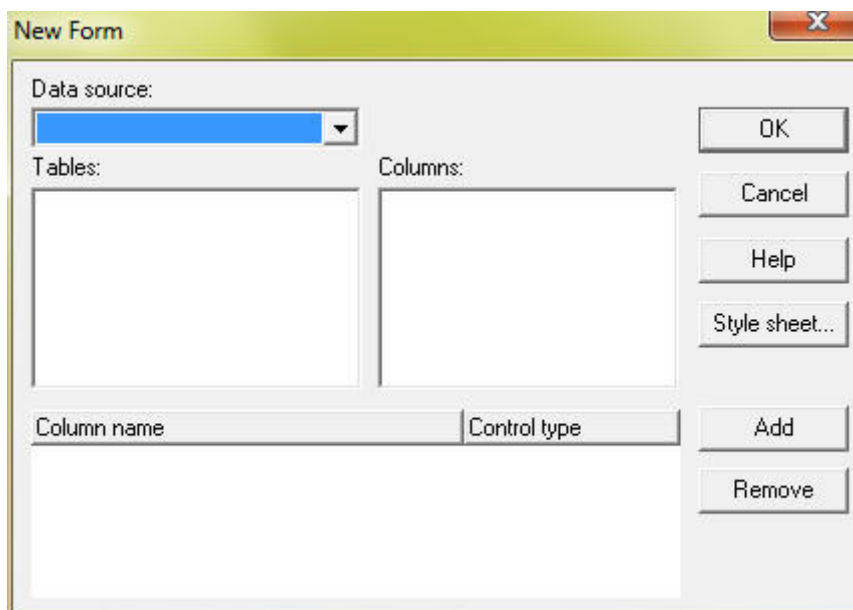


Working with Forms

I. Creating a Data Entry Form

1. Open the Form Designer. Click on *Form Designer* from the *Application Tools* menu or press the Form Designer button .
2. Choose New Form from the File menu in the Form Designer to display the New Form dialog box.





3. Select OpenInsight Tables from the *Data Source* drop-down list. A list of native tables is displayed in the left list box.
4. Select the OpenInsight table from the list in the left list box (by double-clicking the name). Choose **CONTACTS**.
5. The columns from the selected table are displayed in the *Columns* list box.
6. Select columns from the *Columns* list box to use in creating the OpenInsight form. To select a column, double-click on the column name in the *Columns* list box or select the column and click the ADD button. The selected columns are displayed in the selected columns edit table. The order in which columns are selected sets the tab order for the controls in the OpenInsight form being created. Select all the fields.
7. The default control type for columns is an edit line. The edit table control is the default control type for a multi-value column. If you want to specify another control type, select the appropriate column row in the selected columns edit table and double-click to display a list of control types. Double-click the required control type in the control type list for the specified column. The new control type is indicated in the selected columns edit table.
8. If you want to remove a column from the selected columns edit table, select the column row and click the **Remove** button. The row is deleted from the edit table and therefore not displayed on the OpenInsight form.
9. When all columns have been selected and the required controls specified, click the OK button to create the default OpenInsight form. The form should resemble the following:

The screenshot shows a 'Form Designer' window titled 'Untitled'. The form layout includes the following fields and controls:

- Id:** A single-line text input field.
- Name:** A single-line text input field.
- Address:** A table with 3 rows and 1 column.
- City:** A single-line text input field.
- State:** A single-line text input field.
- Zip:** A single-line text input field.
- Tel no:** A table with 3 rows and 1 column.
- Email:** A single-line text input field.
- Contact date:** A single-line text input field.
- Hot lead:** A checkbox.
- Next contact date:** A single-line text input field.
- Controls:** A vertical toolbar on the right side of the window containing various UI controls like buttons, text boxes, and tables.

10. Save the form by choosing Save from the Form Designer File menu. The Save As dialog box will be displayed.

The screenshot shows a 'Save As...' dialog box. It contains a 'Form name' label and a text input field. Below the input field is a list box containing the following items:

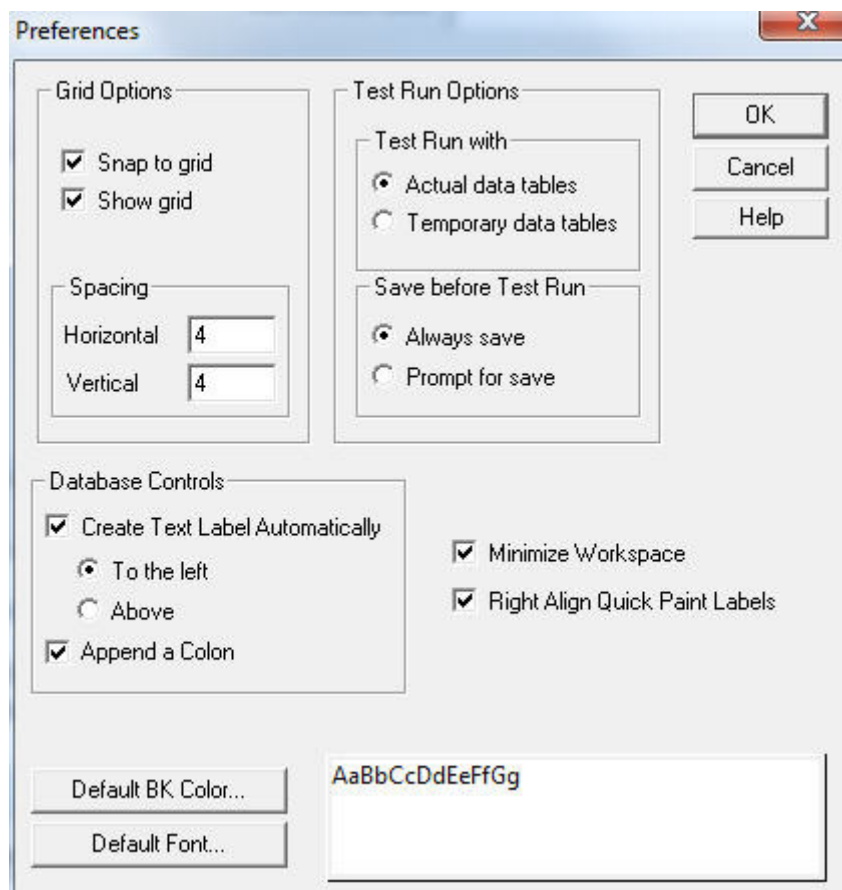
- CS_MAIN
- CS_SPY
- DEFDTEMPLATE
- DEFMDITEMPLATE
- DEFNTEMPLATE


At the bottom of the dialog are 'OK' and 'Cancel' buttons.

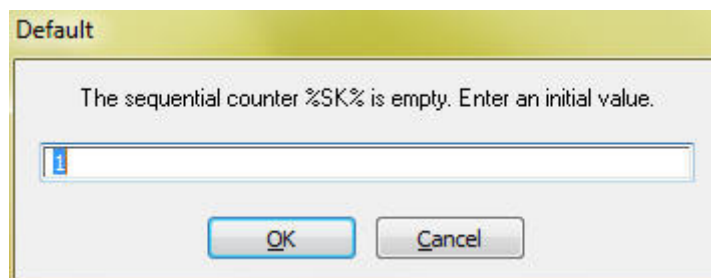
11. Type CONTACTS in the Form Name field and click the OK button.

II. Adding Data to the Table

12. Choose Preferences from the Form Designer Tools menu and verify that *Actual data tables* is selected in the Testrun with: group then click the OK button. This assures that the data you create is saved in the CONTACTS table.



13. Click the **Test Run** button  or choose Test Run from the Form Designer File menu.
14. The first time that the CONTACTS form is executed you are prompted for an initial value for the sequential key counter. This is because you defined the ID column in the table as having a default of SEQKEY. %SK% is a system variable that keeps track of the sequential counter.



15. Type an initial value then click the OK button. The form will display with the ID field containing the value entered in the sequential key dialog box.
16. Enter the following data for this record:

The screenshot shows a software form titled "Untitled - <New Entry>". The form has a menu bar with "File", "Edit", "View", "QBF", and "Help". The fields are as follows:

- Id:** 1
- Name:** Sean FitzSimons
- Email:** sean@revelation.com
- Contact date:** 18 MAR 2010
- Next contact date:** 03/25/2010
- City:** Westwood
- State:** NJ
- Zip:** 07675
- Address:** A list with three rows: 1. 99 Kinderlammack Road, 2. Suite 109, 3. (highlighted).
- Tel no:** A list with three rows: 1. (800) 262-4747, 2. (201) 594-1422, 3. (highlighted).
- Hot lead:** A checkbox that is checked.

Note: Notice that when you tab from the ID field, a date is displayed in the Next Contact field. This is a calculated data field which automatically recalculates. When data is entered into the Contact Date field, the Next Contact will be updated with the correct date.

17. Press the F9 key or choose Save Row from the File menu to save this record. The record is saved, the form is cleared and the ID is updated to the next sequential key.
18. Continue entering the next three records:

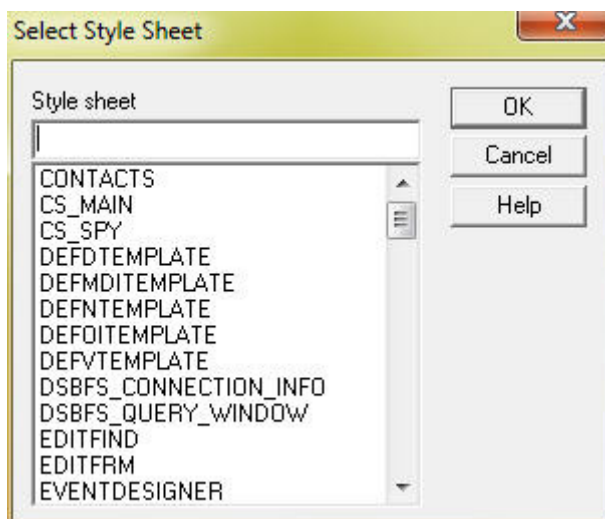
Name	Jill Smith	Robert Jones	Lawrence Granville
Address	22 Waldorf Dr.	16 Scott Terrace	18 Riverview Ave.
City	Youngstown	Asbury	Hibbing
State	OH	MI	MN
Zip	06322	07299	05744
Tel No	888 121-9955	877 333-4444	800 532-9876
Email	jsmith@smith.com	Rjones99@verizone.com	Lg052@roam.net
Contact Date	03/21/10	02/28/10	03/31/10
Hot Lead	True	False	True

19. After adding the last record, choose Close from the File menu to exit the form. This will return you to the Form Designer.

III. Creating an MDI Frame

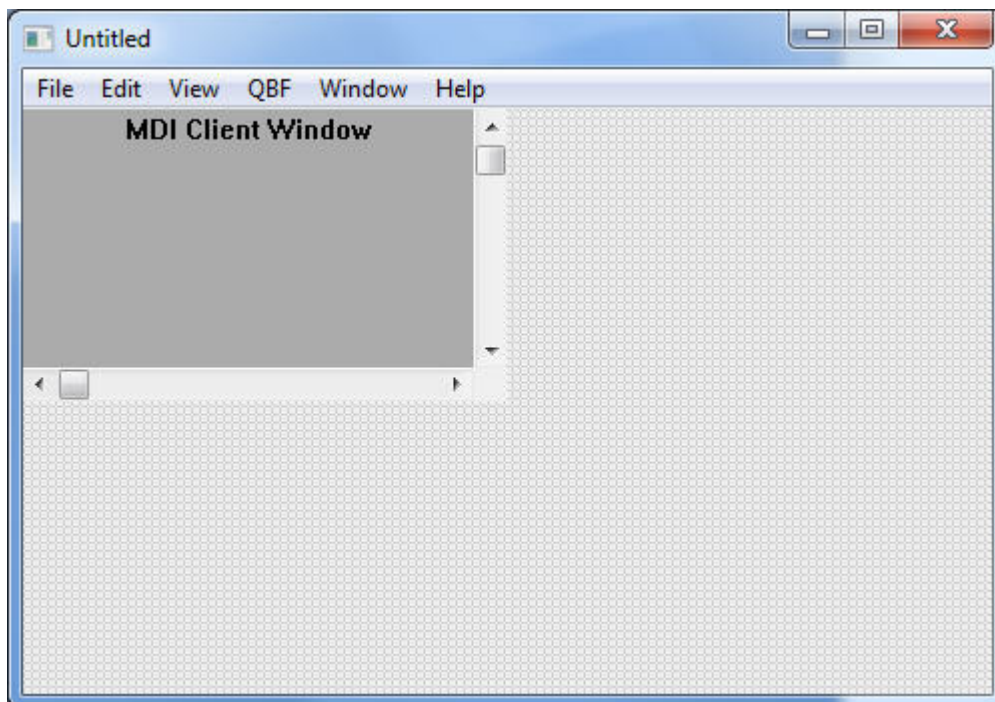
Multiple Document Interface (MDI) windows are child windows or OpenInsight forms that belong to a parent frame and are displayed in the window client area. The MDI Frame is the main window and normally contains a menu, a client area and a status line. The MDI Frame controls all of the child windows. The MDI children are the OpenInsight forms that appear within the client area of the MDI frame.

1. Select New MDI frame from the Form designer File menu to display the Select Style Sheet dialog box which lists the existing OpenInsight forms in the current application.



2. Click the OK button to exit the Select Style Sheet dialog box without selecting a style sheet.

An MDI frame with an MDI client window is displayed. The MDI client window is part of the frame window and cannot be deleted or copied. MDI child windows are confined to the MDI client area. The text “MDI Client Window” is visible only during OpenInSight form design so that you can distinguish the MDI client area from other controls. The Client Window should display as follows:



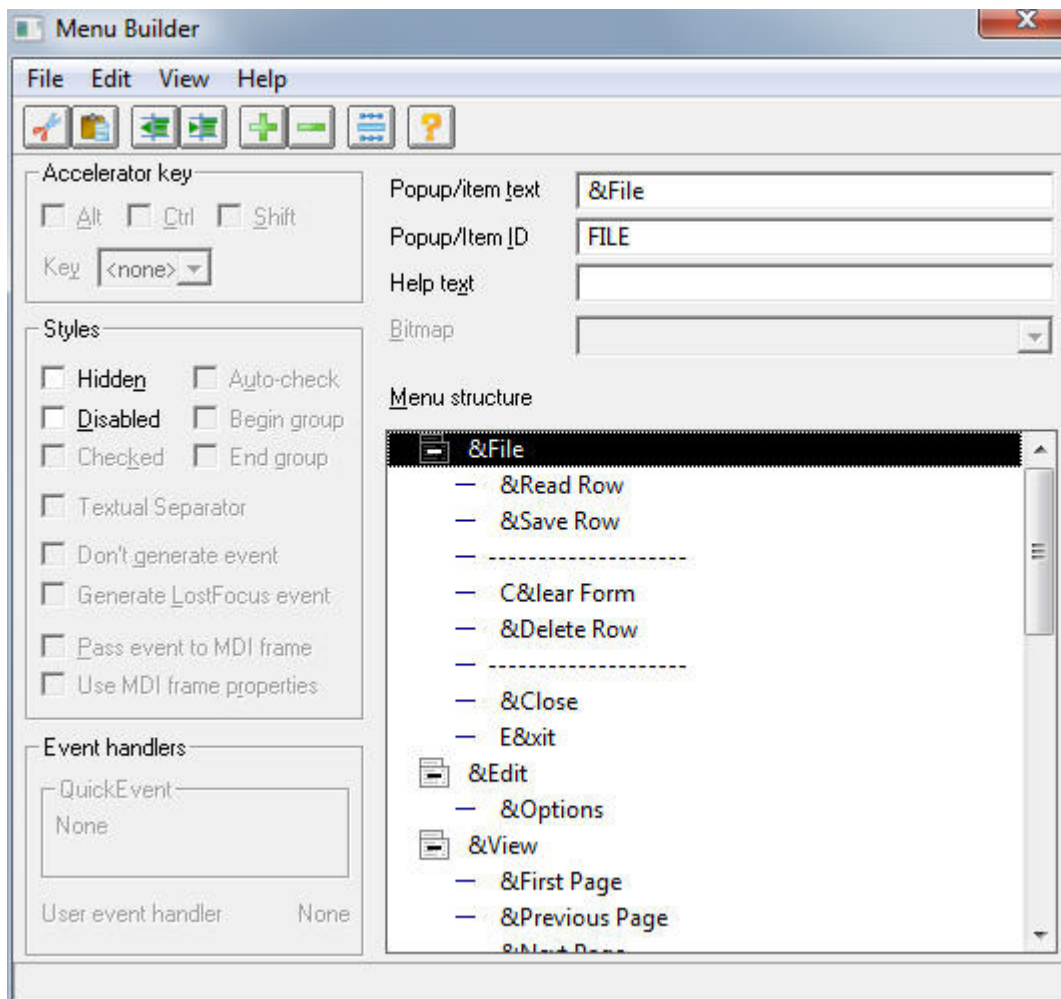
The MDI Client Window can be resized.

IV. Creating an MDI Child

MDI Child windows are those that reside within the MDI client area. MDI Child windows can be any OpenInsight form created for the current application. MDI Child windows are often created prior to creating the MDI frame. A feature of MDI child windows is that they do not contain menus. Menu functionality is provided by the frame.

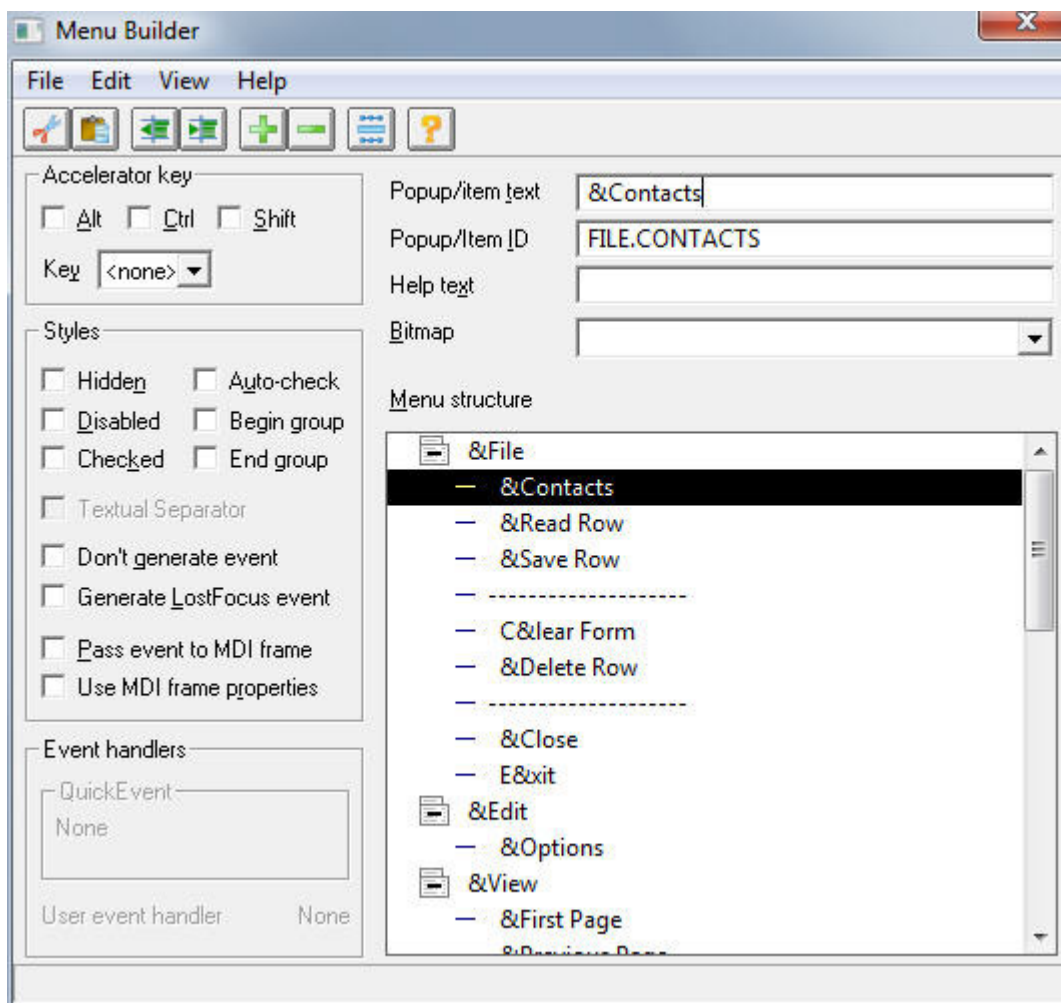
An MDI child is executed from the MDI frame with the BASIC+ function `START_MDICHILD`.

1. Choose *Design* from the *Menu* menu. The *Menu Builder* dialog box is displayed.

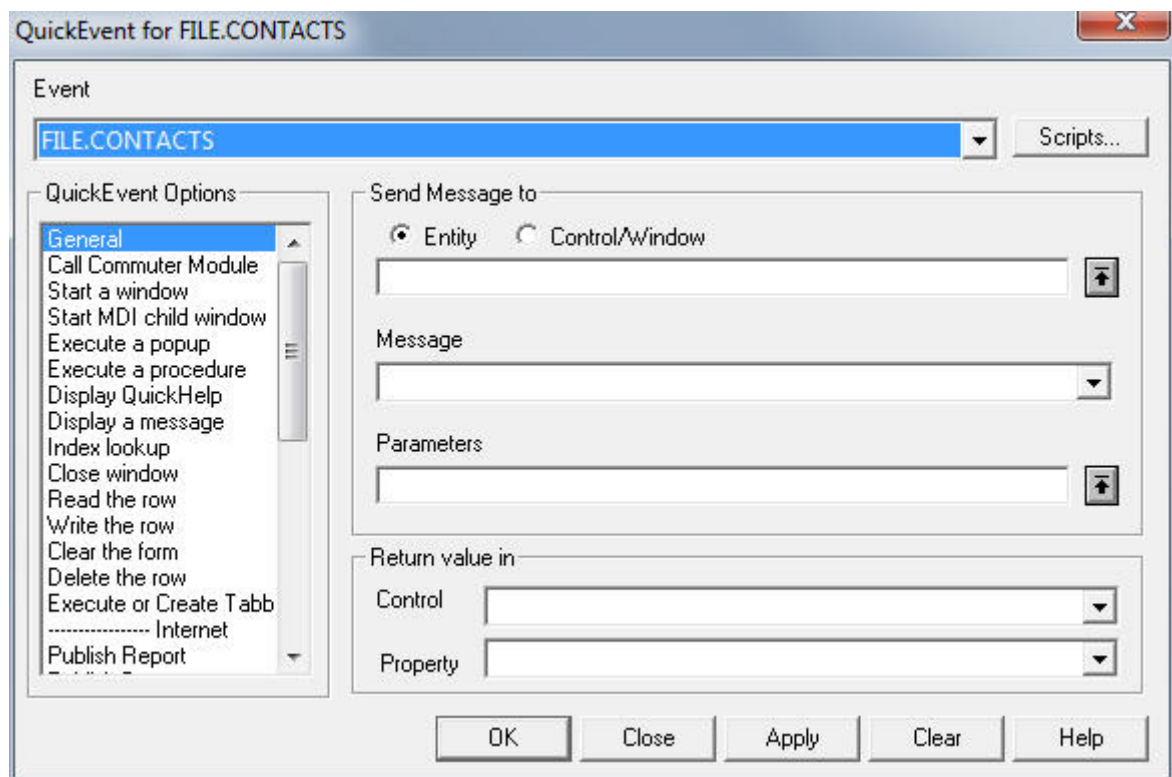


The first menu item is highlighted.

2. Select *Insert item after selection* from the *Edit* menu.
3. Type '&Contacts' in the *Popup/item text* field. The *Popup/Item ID:* field is automatically populated.
4. Choose *Exit/Update* from the *File* menu.



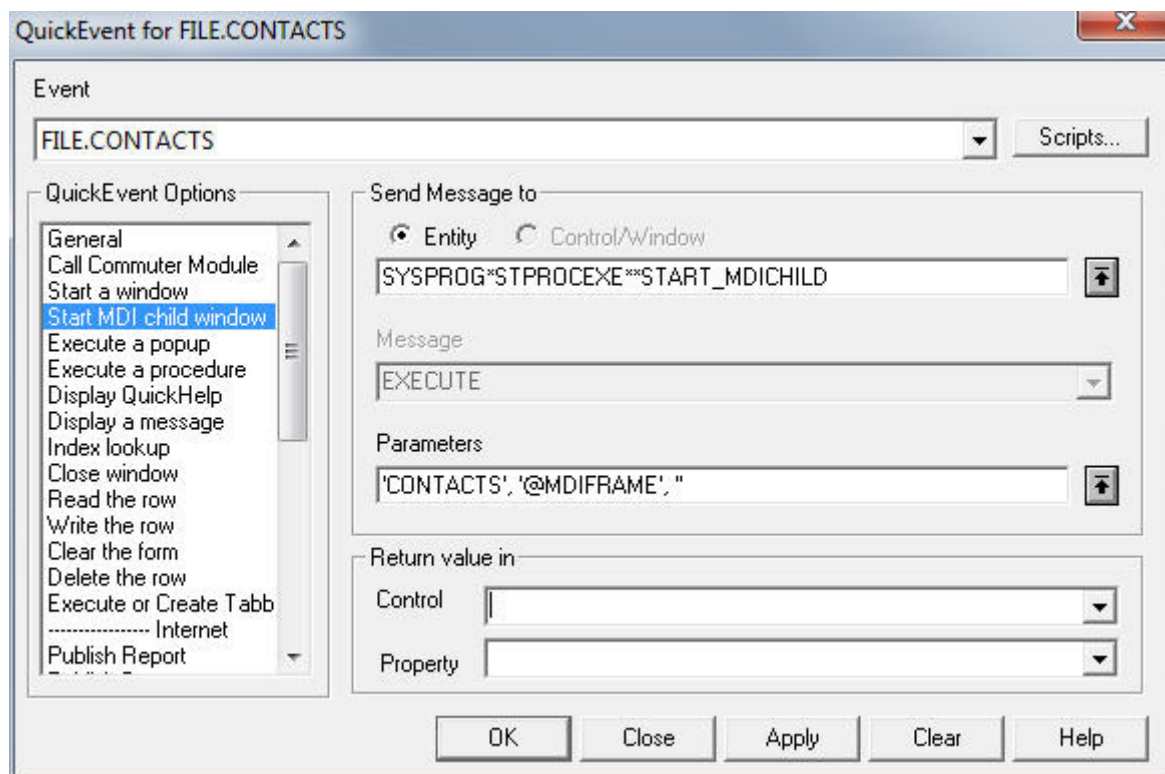
5. Choose *QuickEvents* from the *Menu* menu. The *QuickEvents* dialog box is displayed.



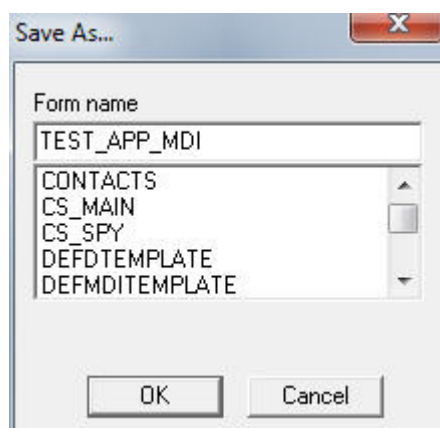
6. Choose *FILE.CONTACTS* from the *Event* drop down list.
7. Choose *Start MDI child window* from the *QuickEvent Options: list*. The QuickEvent dialog box will be populated with default settings for the Start_MDICHild function.



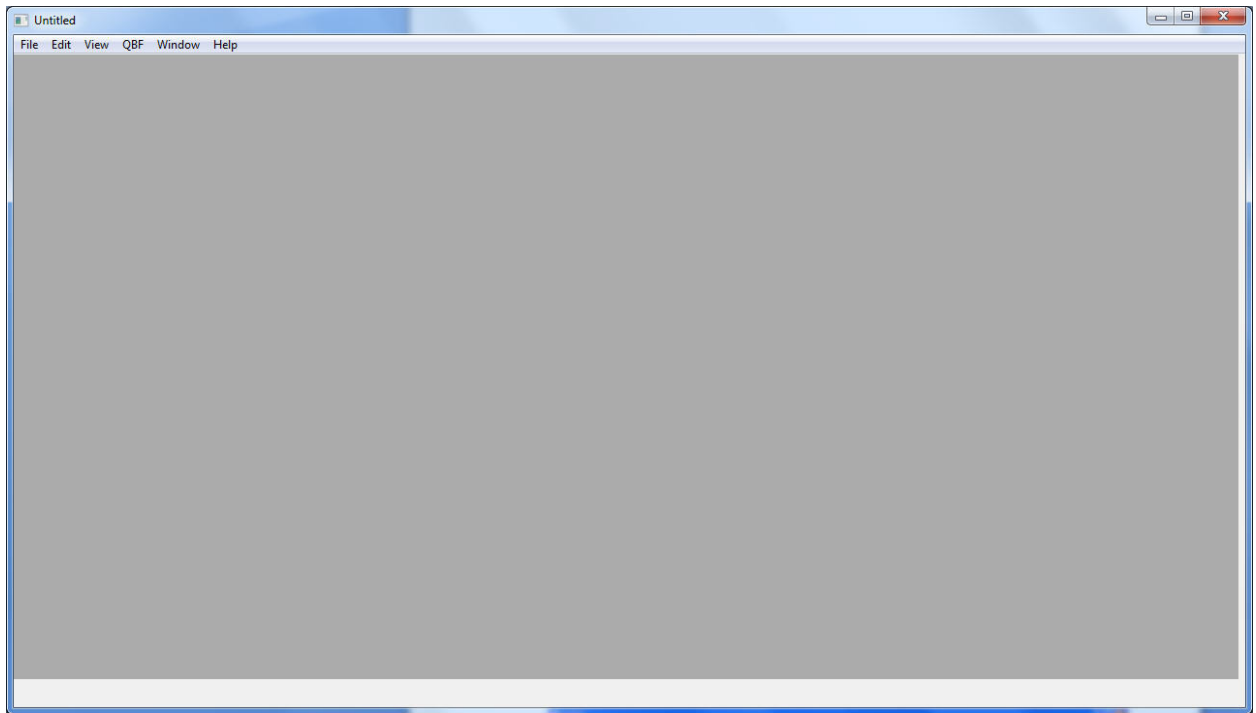
8. On the *Parameters*: change *ChildName* to 'CONTACTS' and *CreateParam* to null (''). Include the single quotes.
9. Press the *Apply* button. This will place a <q> next to FILE.CONTACTS in the Event field.



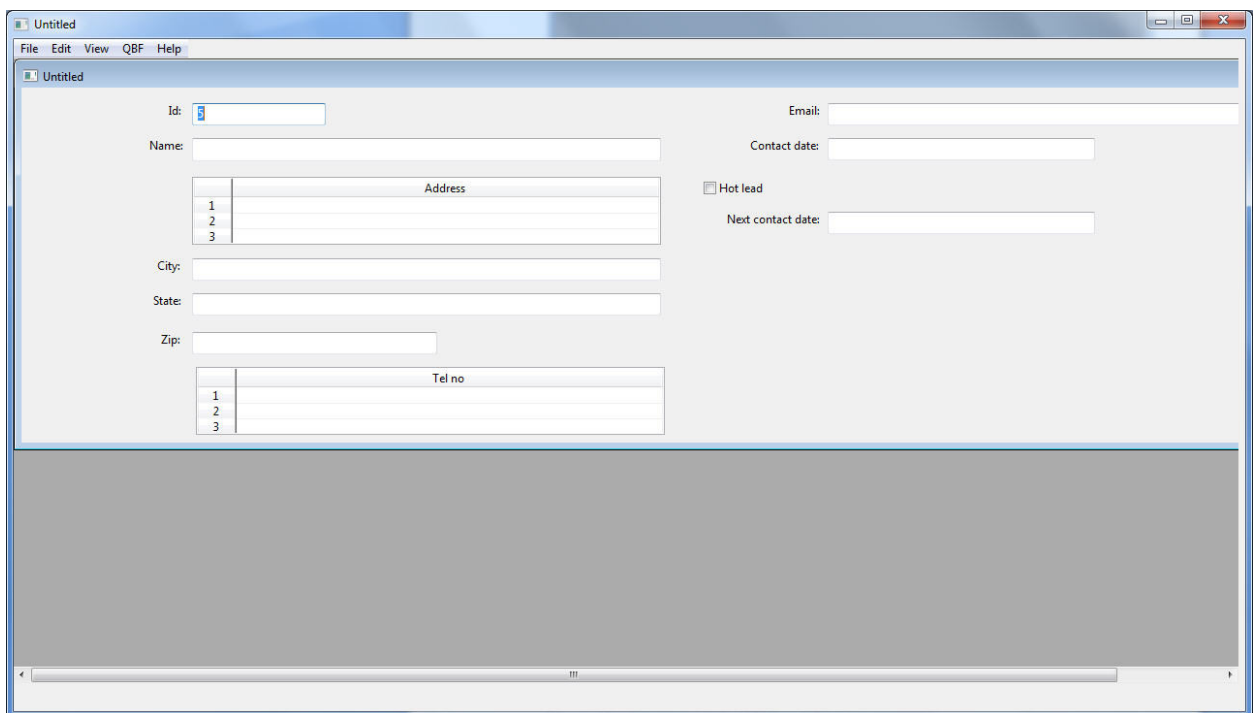
10. Press the *Close* button. This will return you to the MDI Frame.
11. Select *Save* from the *File* menu. Enter TEST_APP_MDI for the name of the form.



12. Test run the form. Choose *Test Run* from the *File* menu in the Form Designer. The following will display:



13. Choose Contacts from the File menu. The following screen will display:



Id:

Name:

	Address
1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>

City:

State:

Zip:

	Tel no
1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>

Email:

Contact date:


☐ Hot lead

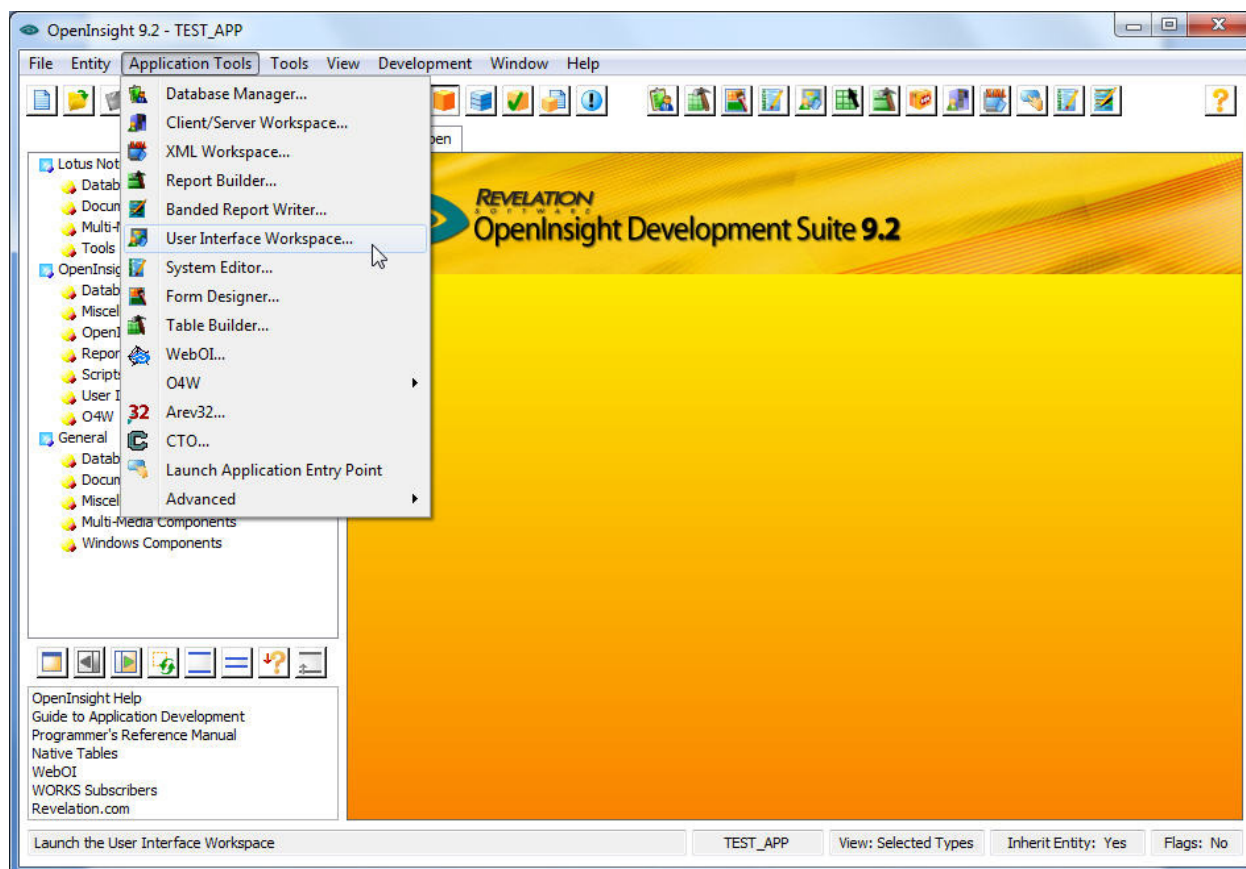
Next contact date:

Your application now contains MDI capability.

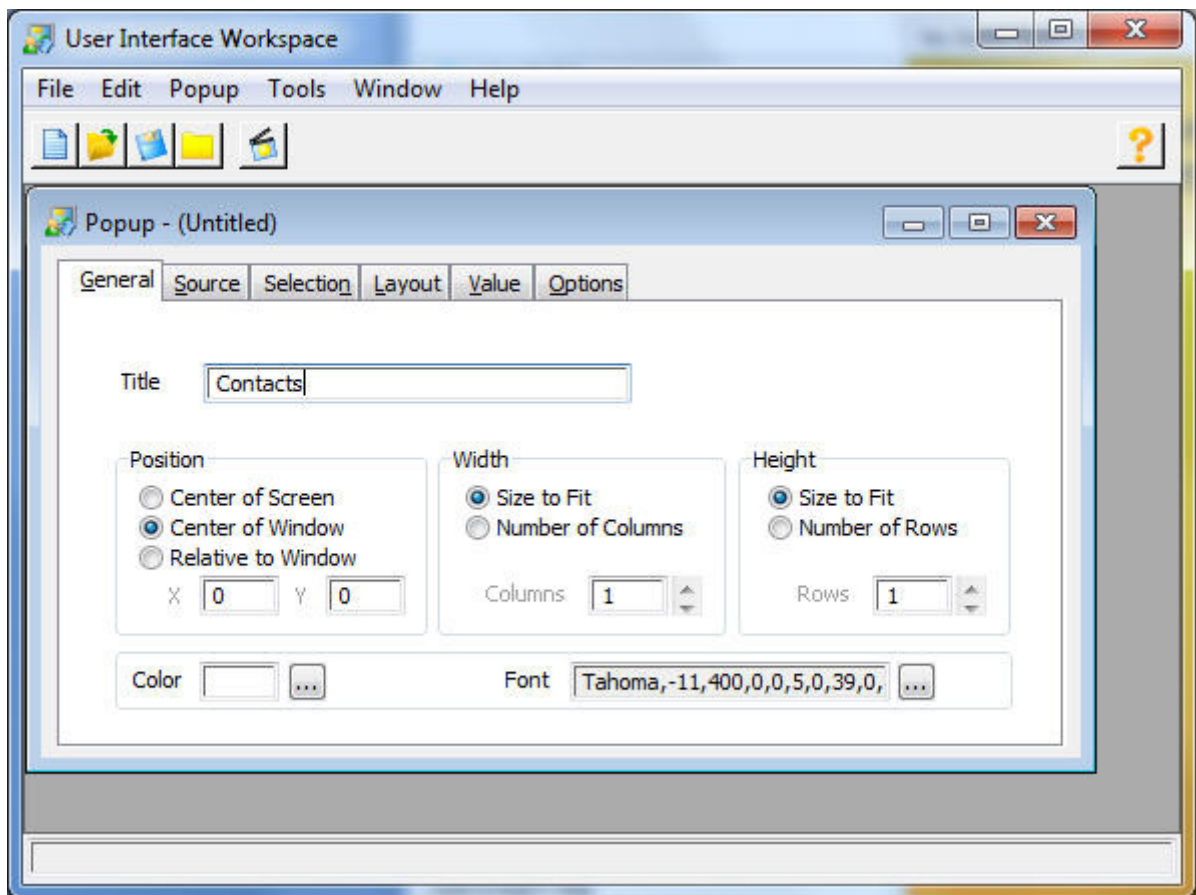
Working with Popups

I. Creating a Popup

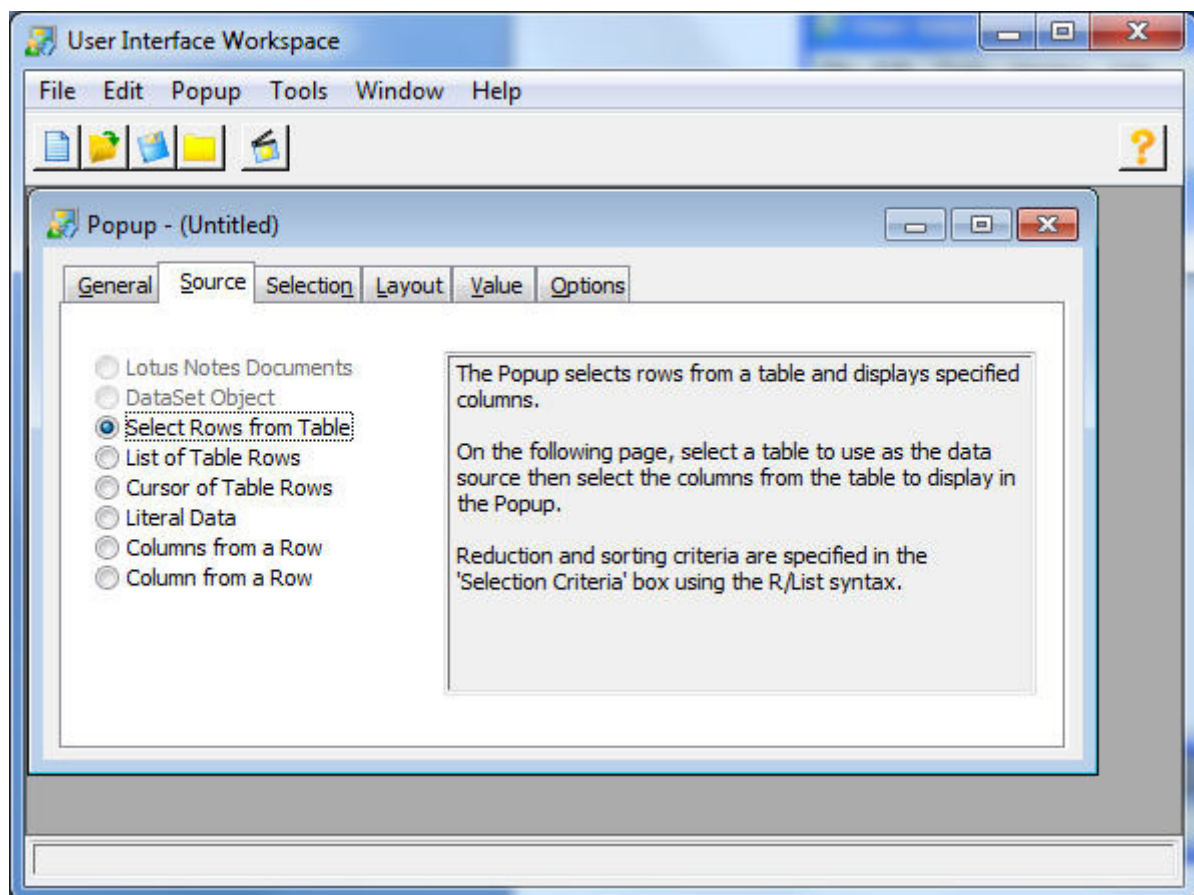
1. Launch the User Interface Workspace by clicking on *UI Workspace* from the *Application Tools* menu or press the UI Workspace button .



2. Choose New from the File menu; choose Popup from the New Entity dialog box to display the Popup Designer below.
3. Type: Contacts in the title edit line on the General tab. This is displayed in the title bar of the popup. Review the other options on this tab and accept the defaults, then click the Source tab.

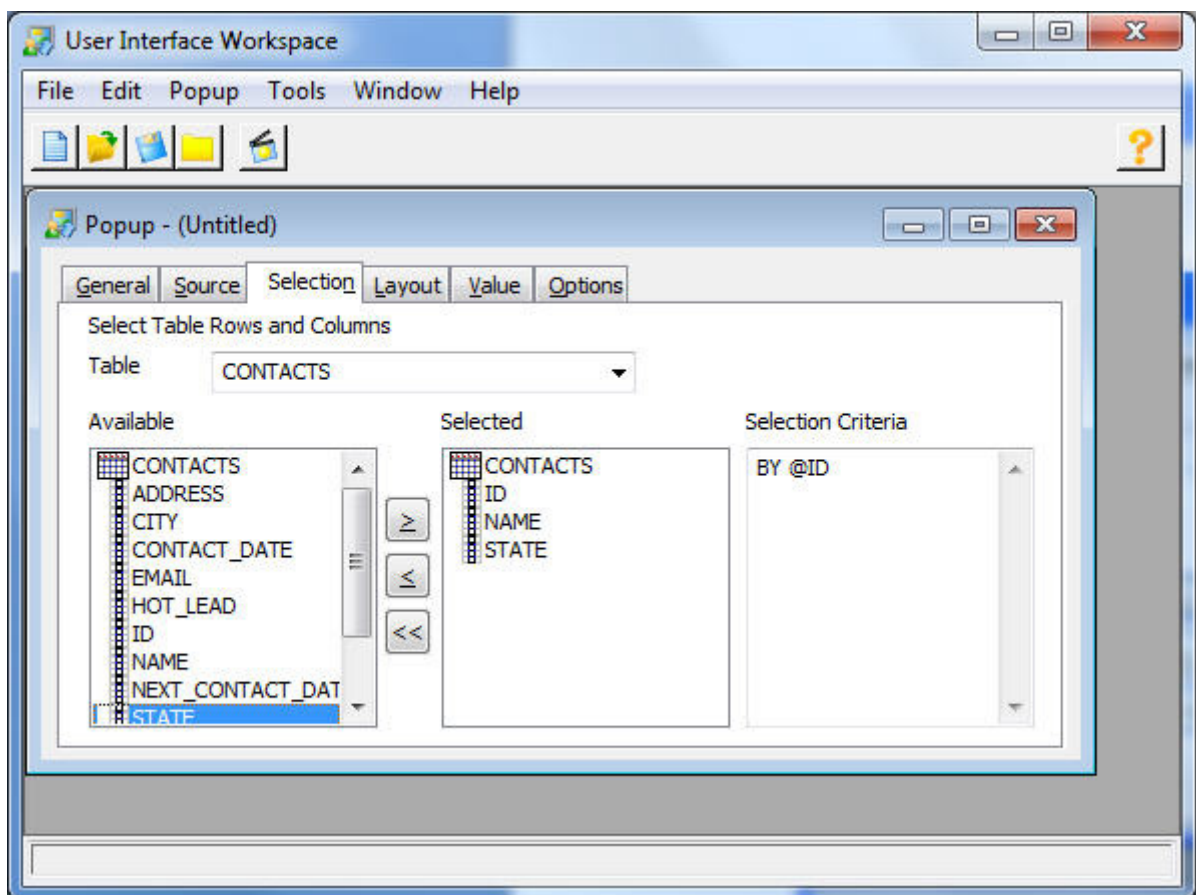


4. Click the *Source* tab where you specify a data source used to populate the Popup. Review all options and choose Select Rows from Table.

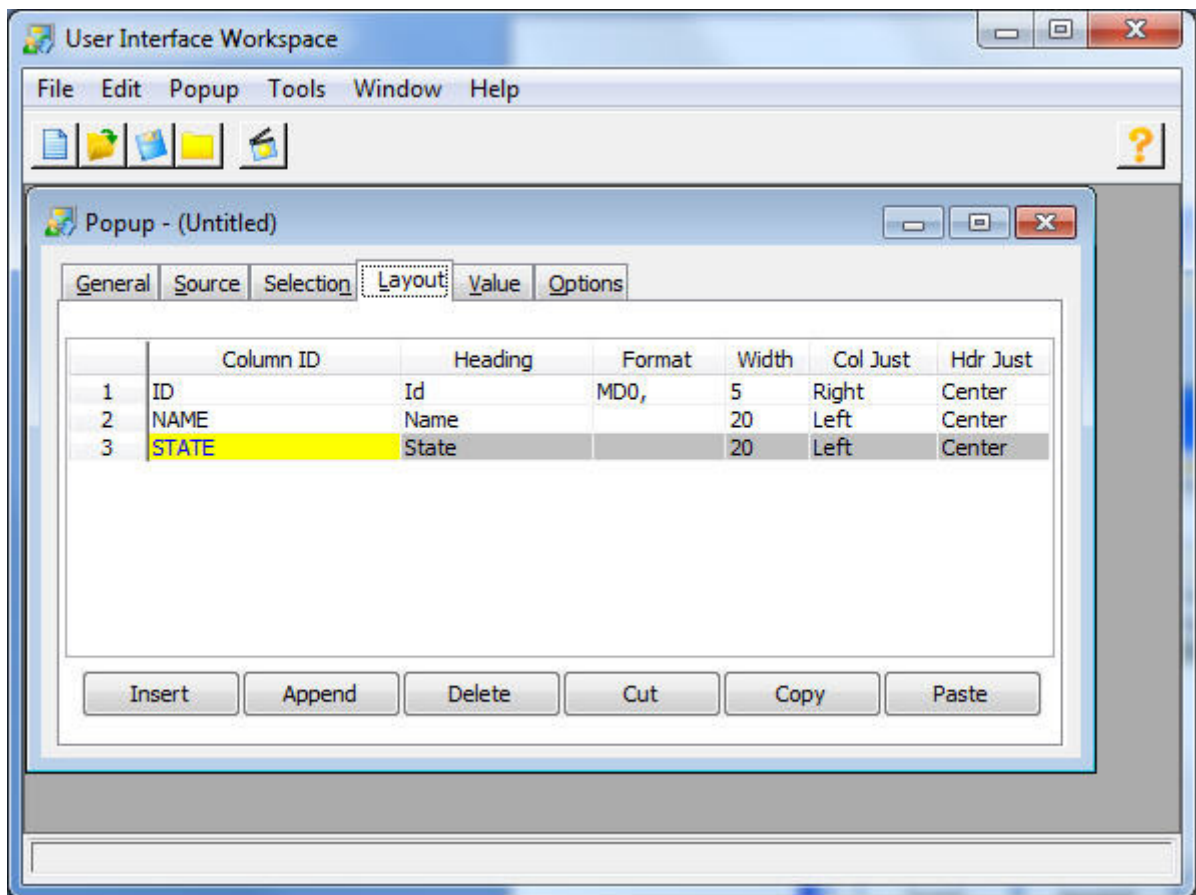


5. Click the *Selection* tab (which is dependent on the Source you selected). Regardless of the appearance of the *Selection* page, the process to be performed is that of selecting the specific data to be displayed in the Popup. The definition of a selection formula may be required.

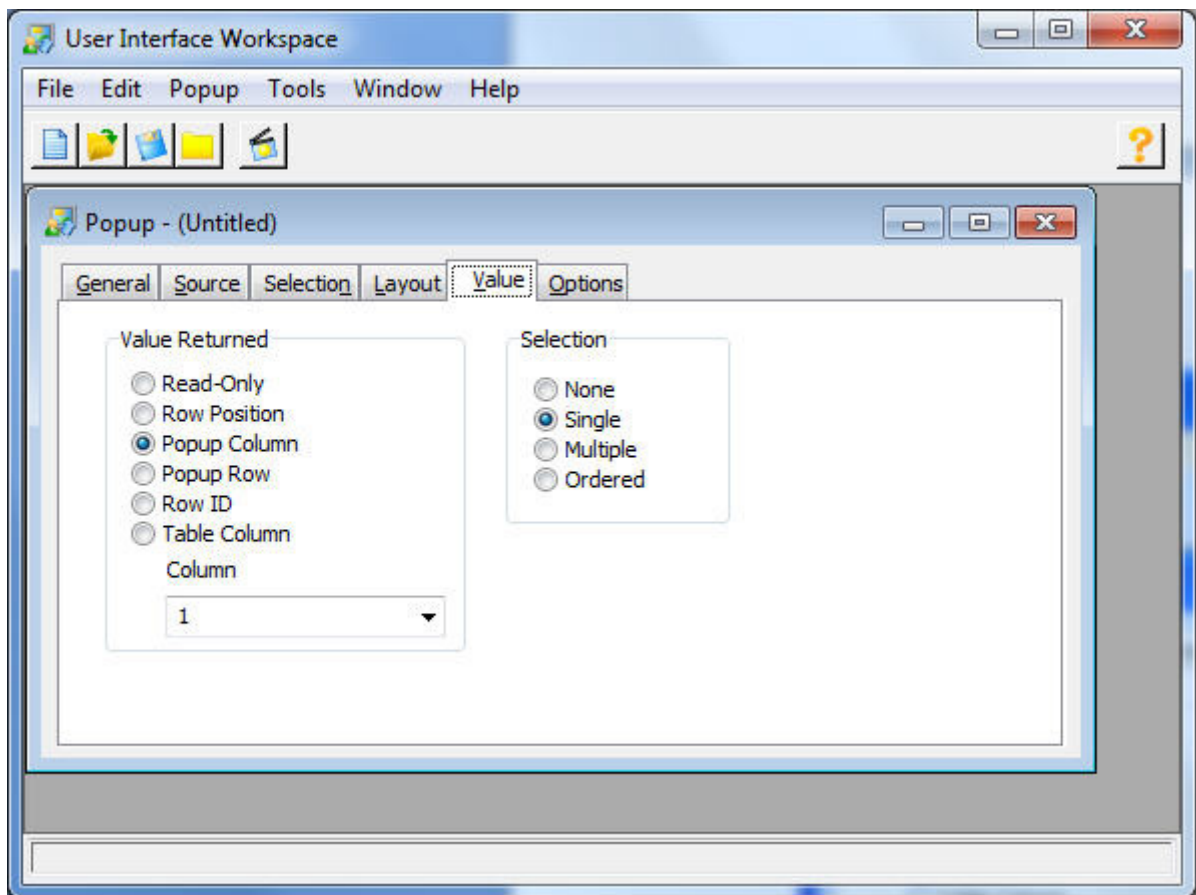
Select CONTACTS from the Table drop down. Choose the fields you want displayed from the Available list and click the right arrow button. This will place the fields selected in the selected list.



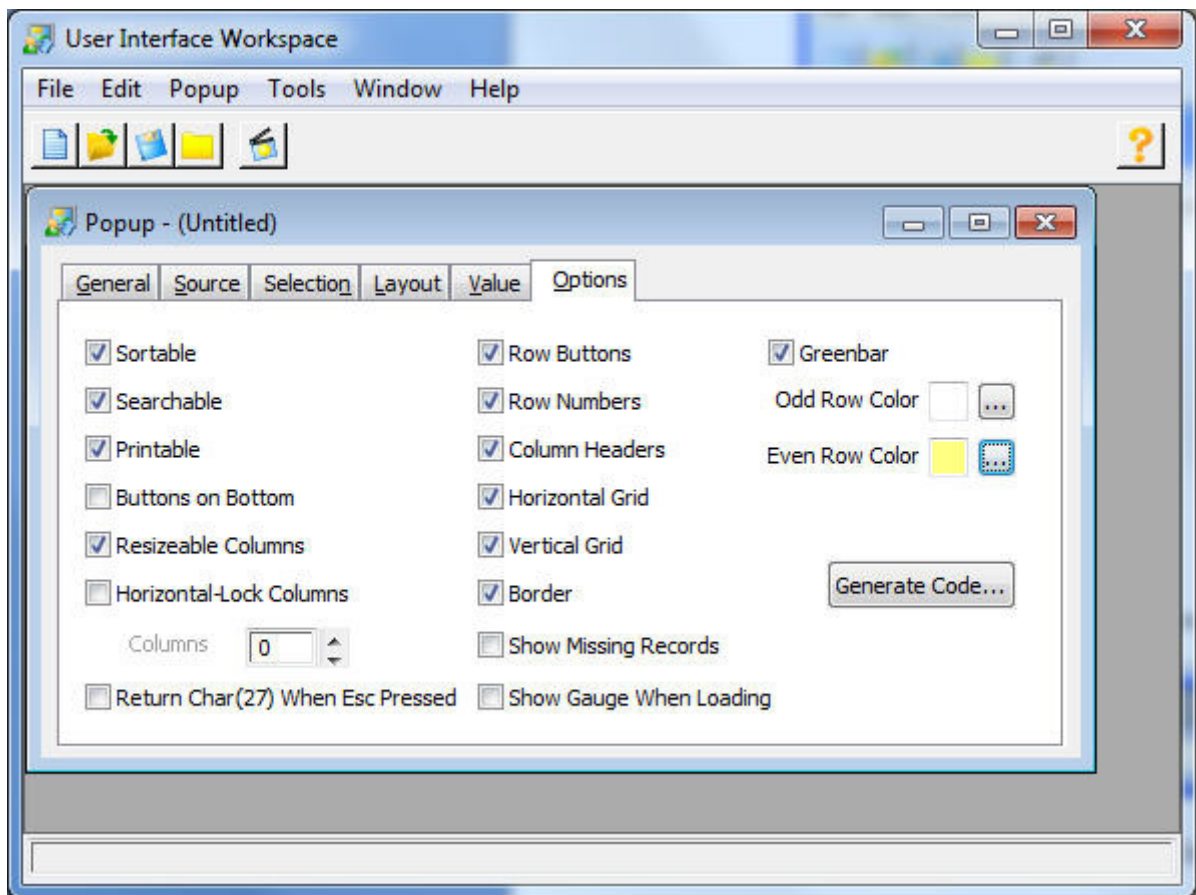
6. Click the *Layout* tab to describe how you want the Popup to look including column headings and alignment, width, etc. This is pre-populated with settings from the data dictionary.



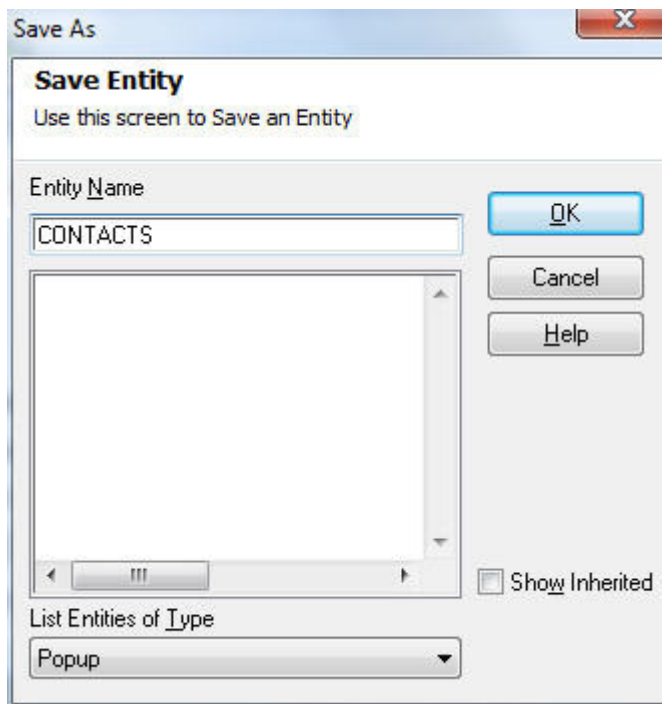
7. Click the *Value* tab where you specify what information you want the Popup to return, and specify the number of selections that can be made in the Popup.



8. Click the *Options* tab where a number of miscellaneous characteristics of the Popup are specified.



9. Save the Popup, and then test it by choosing File, Test Run.



Contacts X

	Id	Name	State
1	1	Sean Fitzsimons	NJ
2	2	Jill Smith	OH
3	3	Robert Jones	MI
4	4	Lawrence Granville	MN

OK

Cancel

Search...

Print

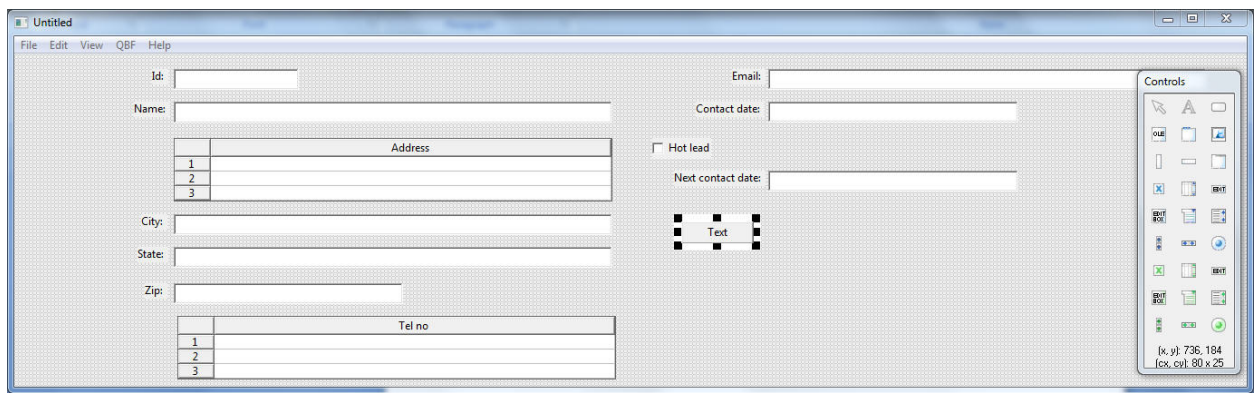
II. Adding Popup Functionality to a Window

The most common use of a Popup is to provide users with a list of choices during data entry. The following procedure will demonstrate how to add a button to an existing form and the Event Handler necessary to execute the Popup and place the results within a data field.

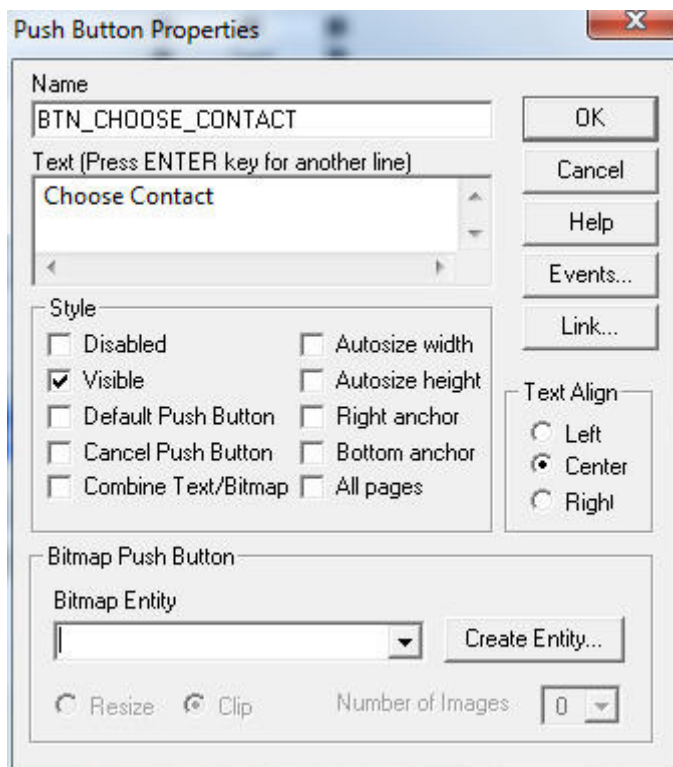
1. Open the Form Designer.
2. Select *Open* from the *File* menu. Choose the CONTACTS form.
3. From the Controls tool bar select a button by clicking on the OK button icon, in the upper right corner



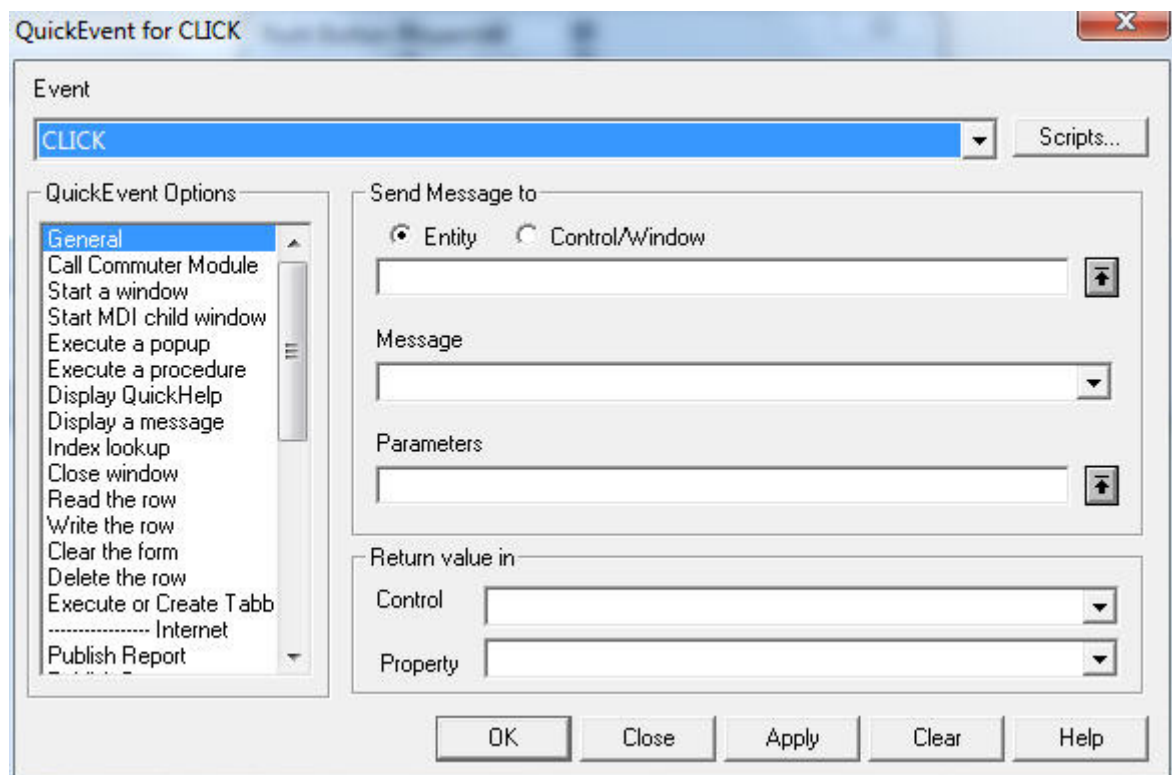
4. Place the control on the form by moving the mouse cursor to a location on the form where you want to place the control. The mouse cursor changes to the control's icon with a + appearing to the left and above the icon as soon as the mouse cursor enters the form. The + is used to help position the control. Click the left mouse button where you want to place the control. The control is displayed in the form in the selected state (with handles visible).



- Double-click on the button. The Push Button Properties window will appear. Change the *Name* of the control to “BTN_CHOOSE_CONTACT” and the *Text* that is displayed on the control to “Choose Contact”.



- Click on the *Events* button to display the QuickEvent dialog box.

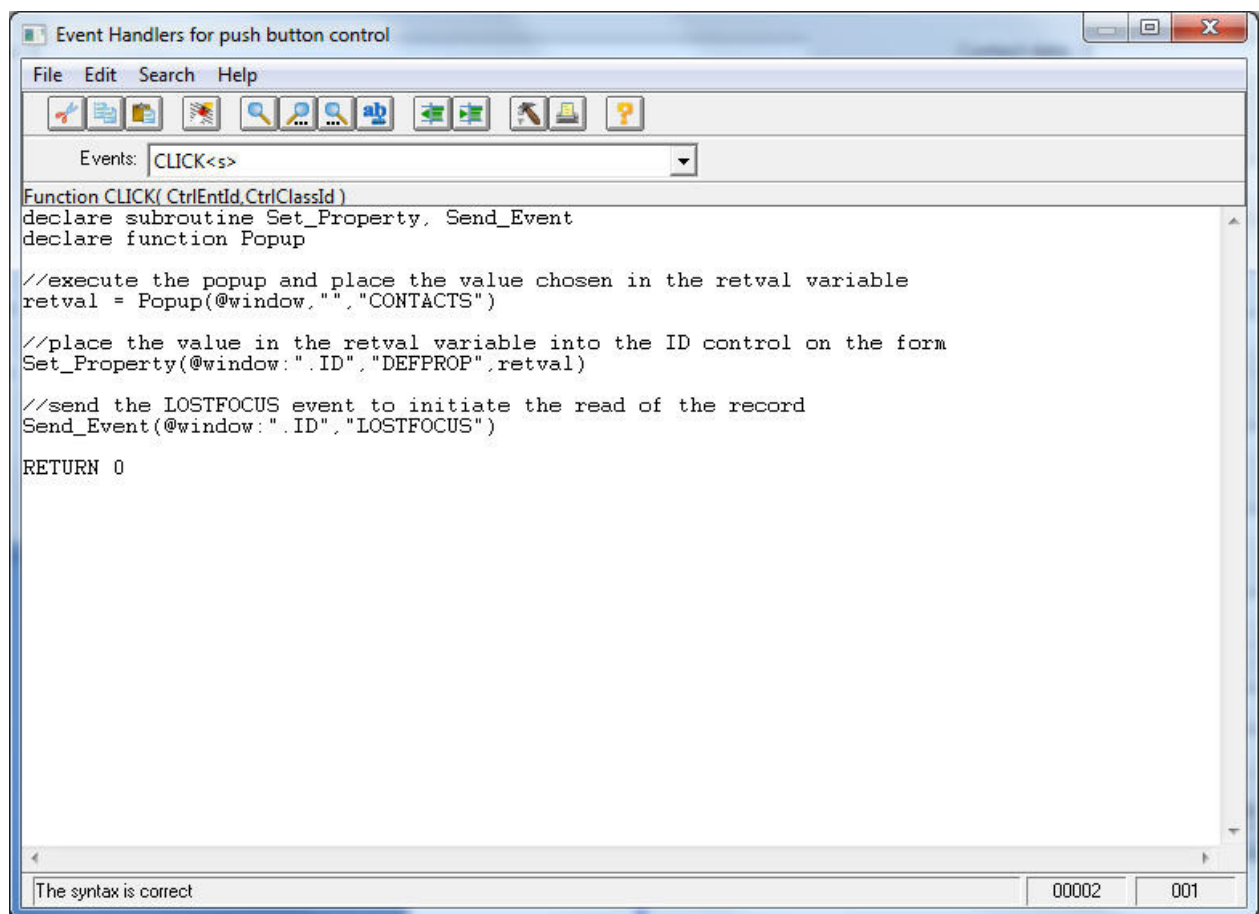


7. The CLICK event will be chosen by default.
8. Click on the *Scripts* button. This will launch the Event Editor. In the Editor type the following code:

```
declare subroutine Set_Property, Send_Event
declare function Popup

retval = Popup(@window,"","CONTACTS")
Set_Property(@window:".ID","DEFPROP",retval)
Send_Event(@window:".ID","LOSTFOCUS")
```

9. Select *Check Syntax* from the *File* menu. Any errors will be displayed at the bottom of the Event Editor window. If the syntax is correct the status line will display "The syntax is correct".



This will cause a popup to display when the Choose Contacts button is clicked. The user may choose a Contact. The value chosen will be placed into the ID control on the Contacts form. When the ID field loses focus a read of the ID placed into the field will occur.

10. Select *Exit/Update* from the *File* menu. This will save the code and return to QuickEvent dialog box.
11. Press the *Apply* button. This will place a <s> next to CLICK in the Event field.
12. Press Close. This will return you to the Push Button Properties window. Select OK on the properties window. This will return you to form.

The screenshot shows a software window titled "Untitled" with a menu bar (File, Edit, View, QBF, Help). The form contains the following fields and controls:

- Id:** A single-line text input field.
- Name:** A single-line text input field.
- Address:** A table with 3 rows and 1 column.
- City:** A single-line text input field.
- State:** A single-line text input field.
- Zip:** A single-line text input field.
- Tel no:** A table with 3 rows and 1 column.
- Email:** A single-line text input field.
- Contact date:** A single-line text input field.
- Next contact date:** A single-line text input field.
- Hot lead:** A checkbox.
- Choose Contact:** A button.

A "Controls" panel is visible on the right side of the window, containing various icons for editing and viewing. The status bar at the bottom right shows the coordinates (x, y): 8, 112 and the size (cx, cy): 1376 x 424.

13. Choose *Save* from the *File* menu.
14. To test run the MDI Frame, access the Contacts form from the File menu and click on the Choose Contact button.


Working with Reports

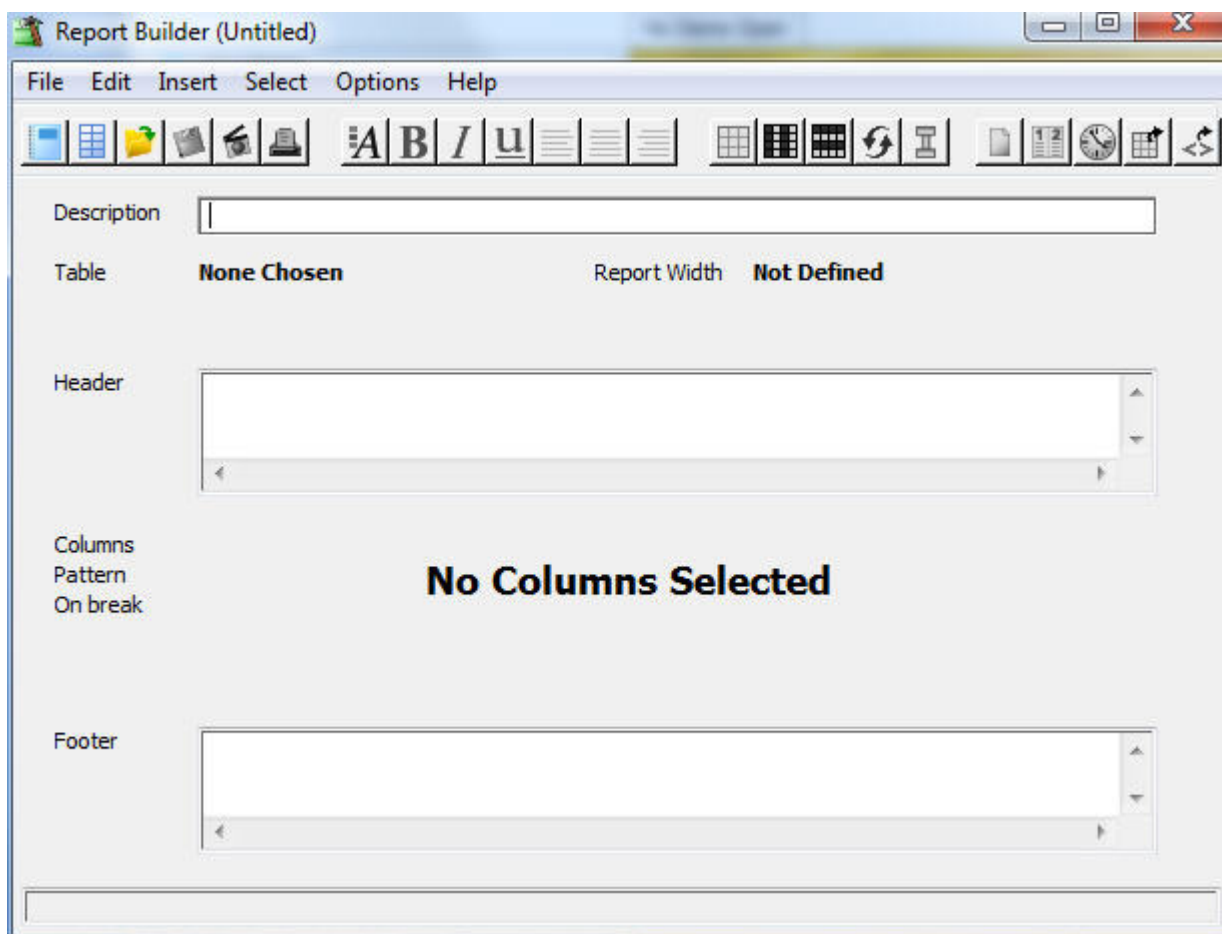
I. Report Builder+

The OpenInsight Report Builder is a tool used to create R/List type reports. Using the Report Builder+ the developer or end-user can create columnar reports on any attached table with a minimum of effort.

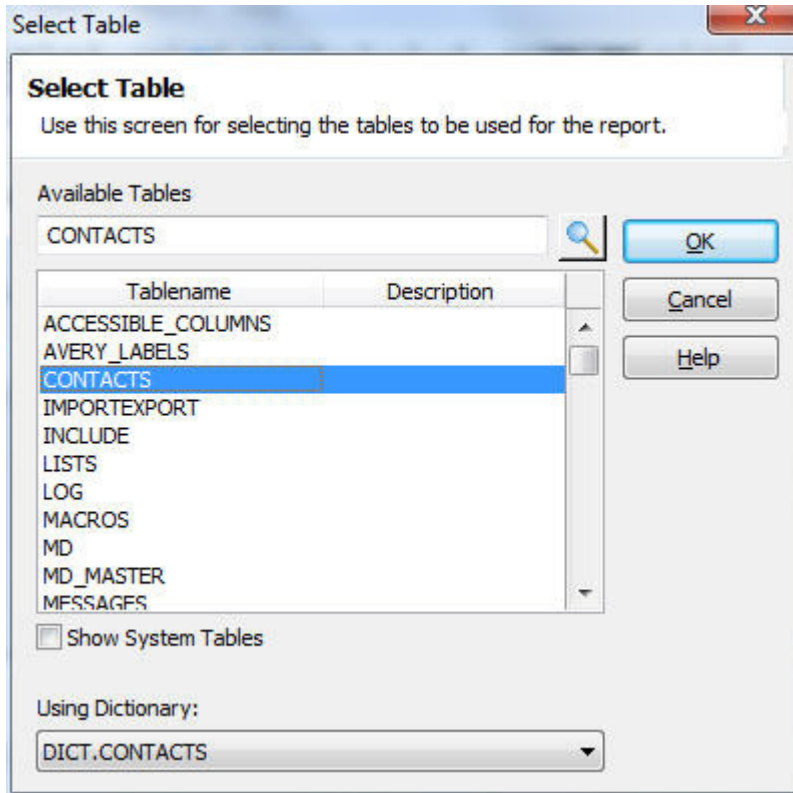
II Creating a Report

In this procedure you'll create a columnar report based on the CONTACTS table. Exit all tools and return to the Application Manager.

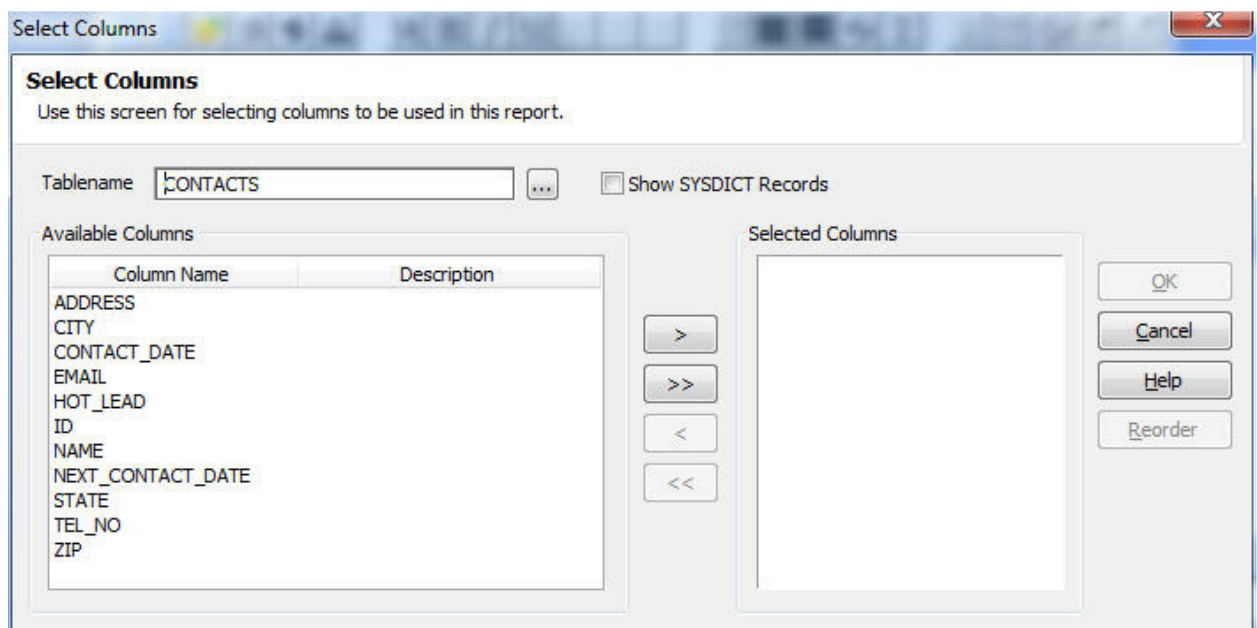
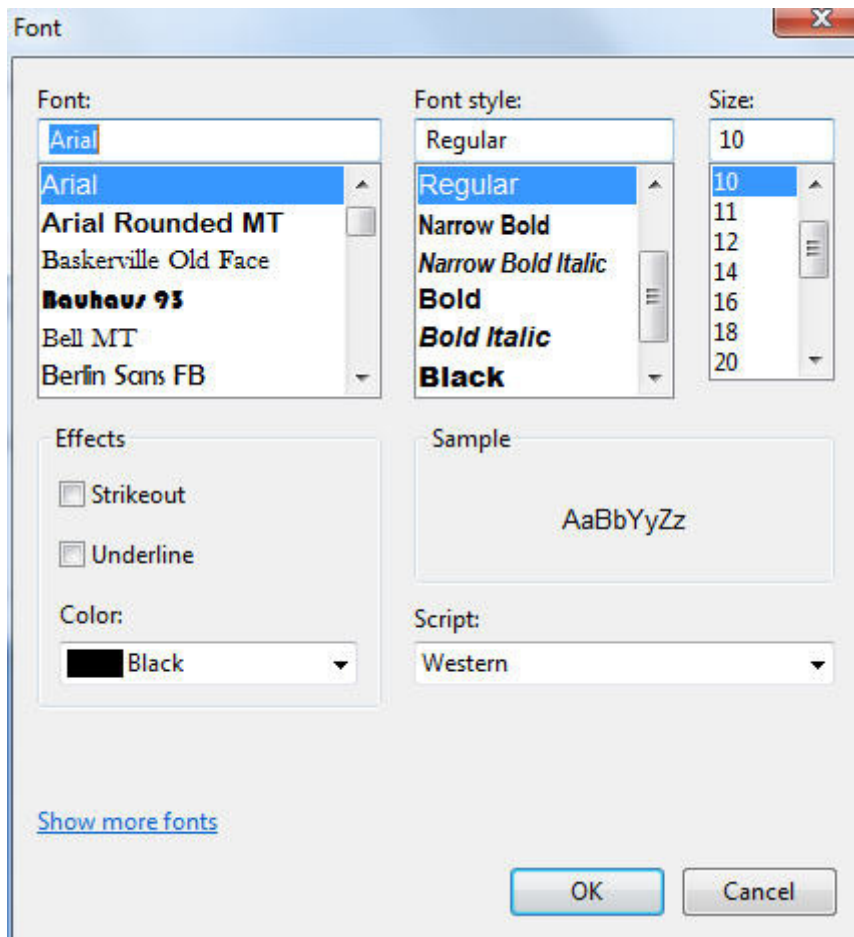
1. Click on *Report Builder* from the *Application Tools* menu or press the Report Builder button .



2. Choose *New Report* from the *File* menu. The *Select Table* dialog box is displayed. Select the CONTACTS tables



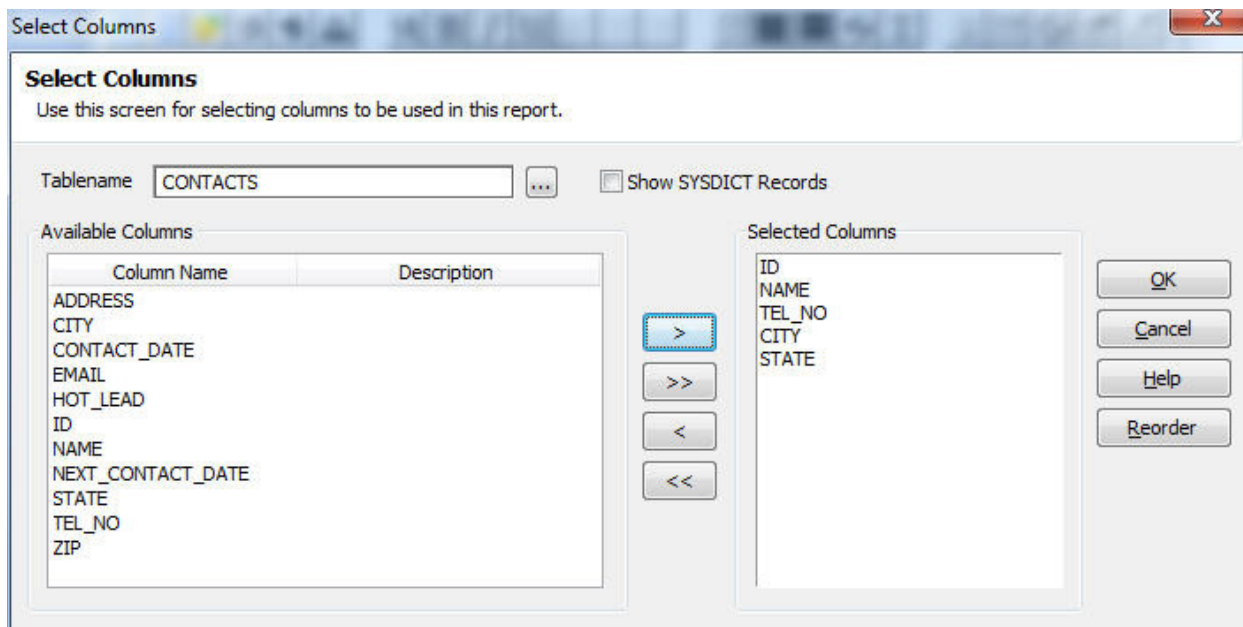
- Click the OK button. The *Font* dialog box is displayed followed by the *Select Columns* dialog box.



4. From the *Available Columns* list box, select the following fields by double clicking on the Column Name for each specified field and clicking the > button.

ID
NAME
TEL_NO
CITY
STATE

5. The selected field names display in the *Selected Columns* list.



Note: If the Selected Columns are not in the order in which you want them to appear on the report, click the *Reorder* button to display the Reorder Dialog box.

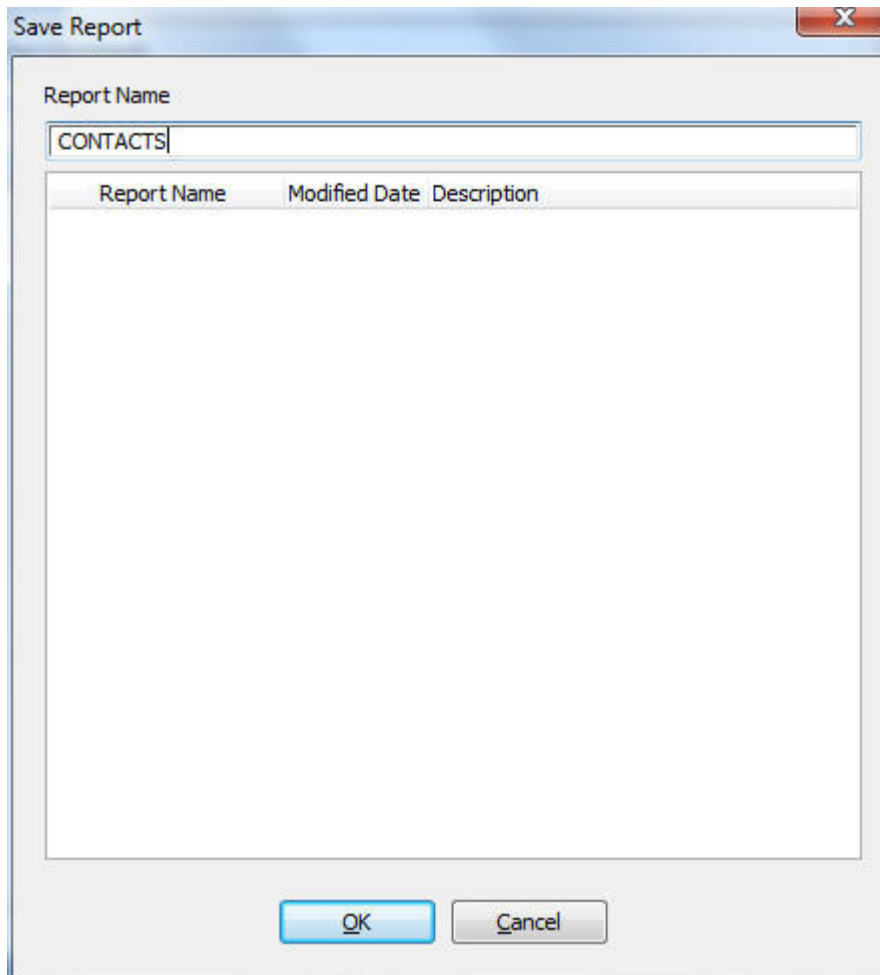
6. The screen will now resemble the following:

The screenshot shows the 'Report Builder (Untitled)' window. The menu bar includes File, Edit, Insert, Select, Options, and Help. The toolbar contains various icons for report design, including text, tables, and formatting. The main area is divided into sections for report configuration:

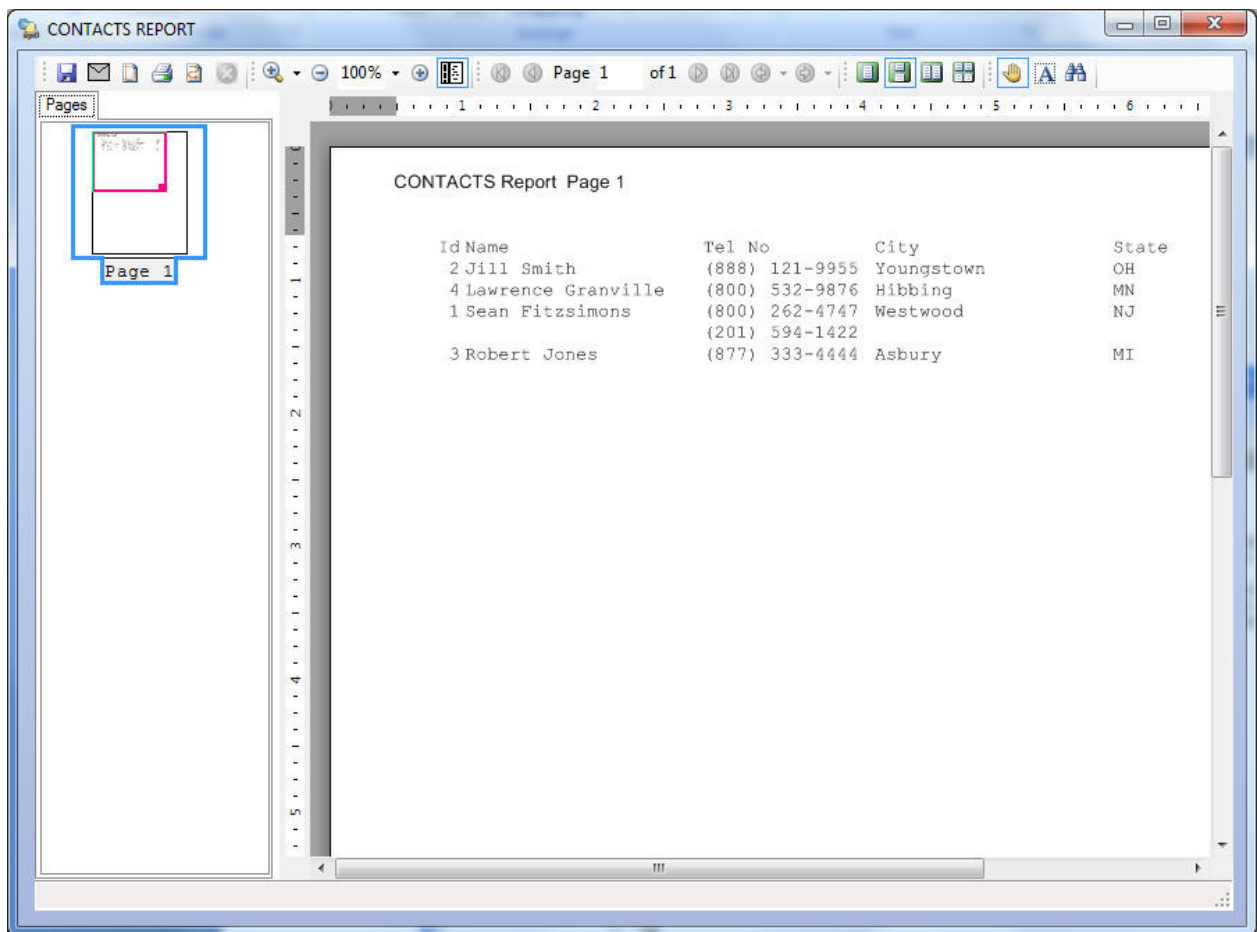
- Description:** A text input field.
- Table:** The table name is **CONTACTS**. The **Report Width** is set to **79**.
- Header:** A large text input field for the header content.
- Columns:** A table with the following structure:

Id	Name	Tel No	City	State
9,999	XXXXXXXXXXXXXXXXXXXX	[PHONE_FORMAT	XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX
- Pattern:** A text input field.
- On break:** A text input field.
- Footer:** A large text input field for the footer content.

7. To save the report, choose *Save* from the *File* menu. Type CONTACTS as the report name and click the OK button.



8. Choose *Print Preview* from the *File* menu. The following report will be displayed.

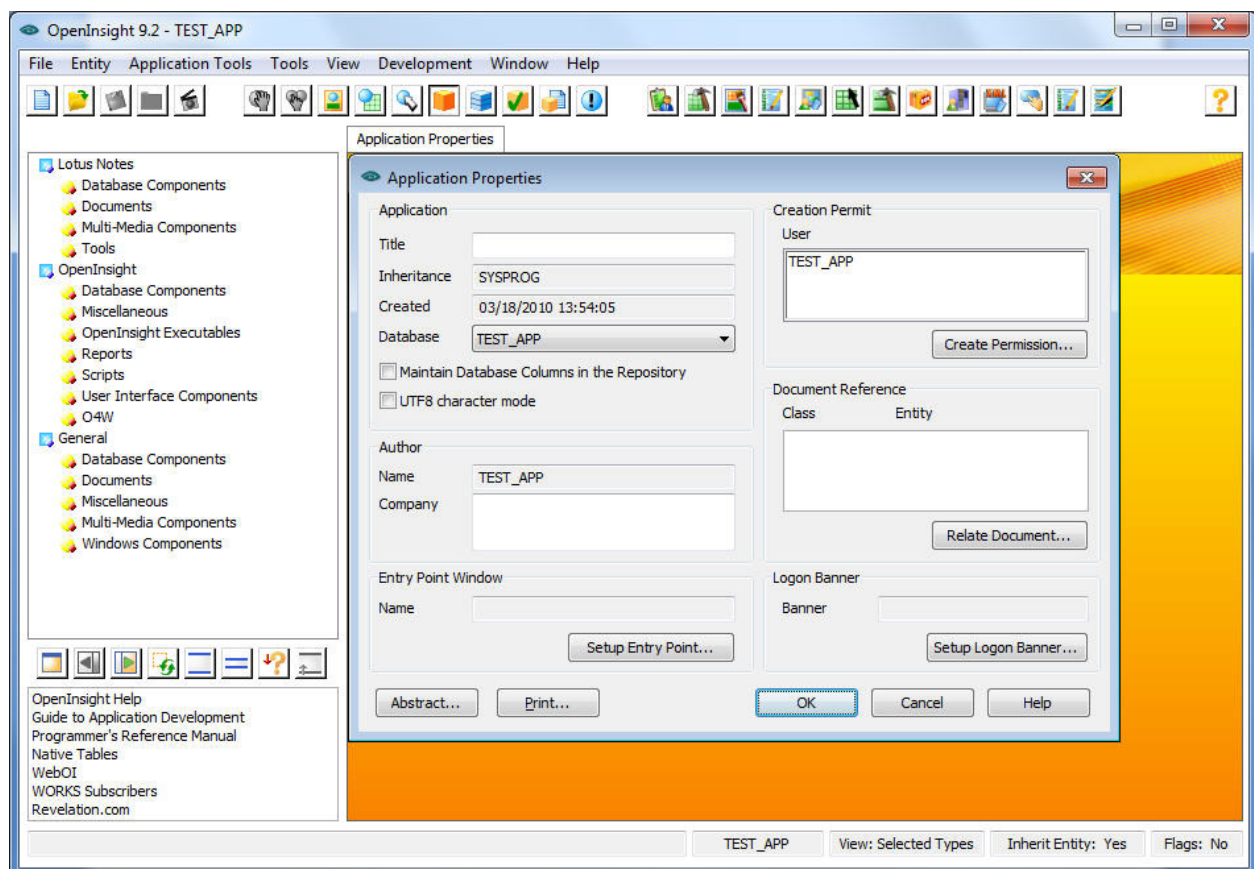


9. Click the Close button on the Preview window.

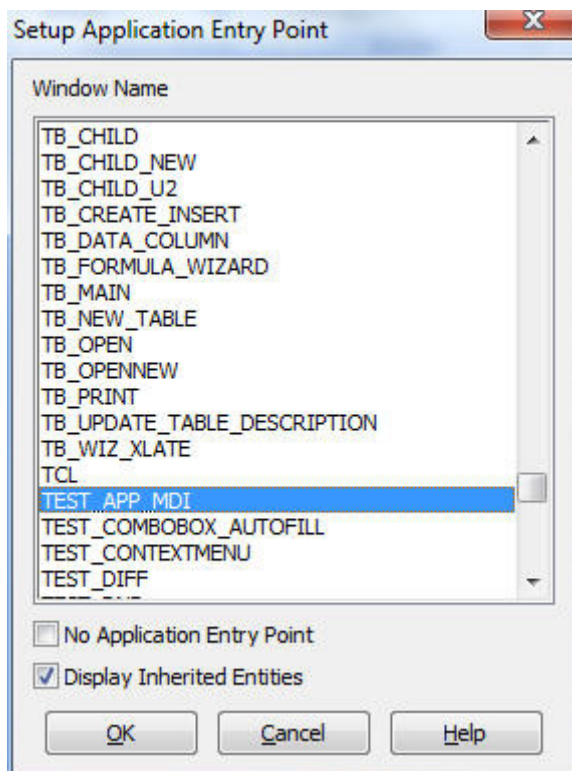
Setting an Application Entry Point

The application entry point is the first window that is displayed when the application is executed at runtime. For our purposes this will be the MDIFrame TEST_APP_MDI. The following steps will define the application entry point.

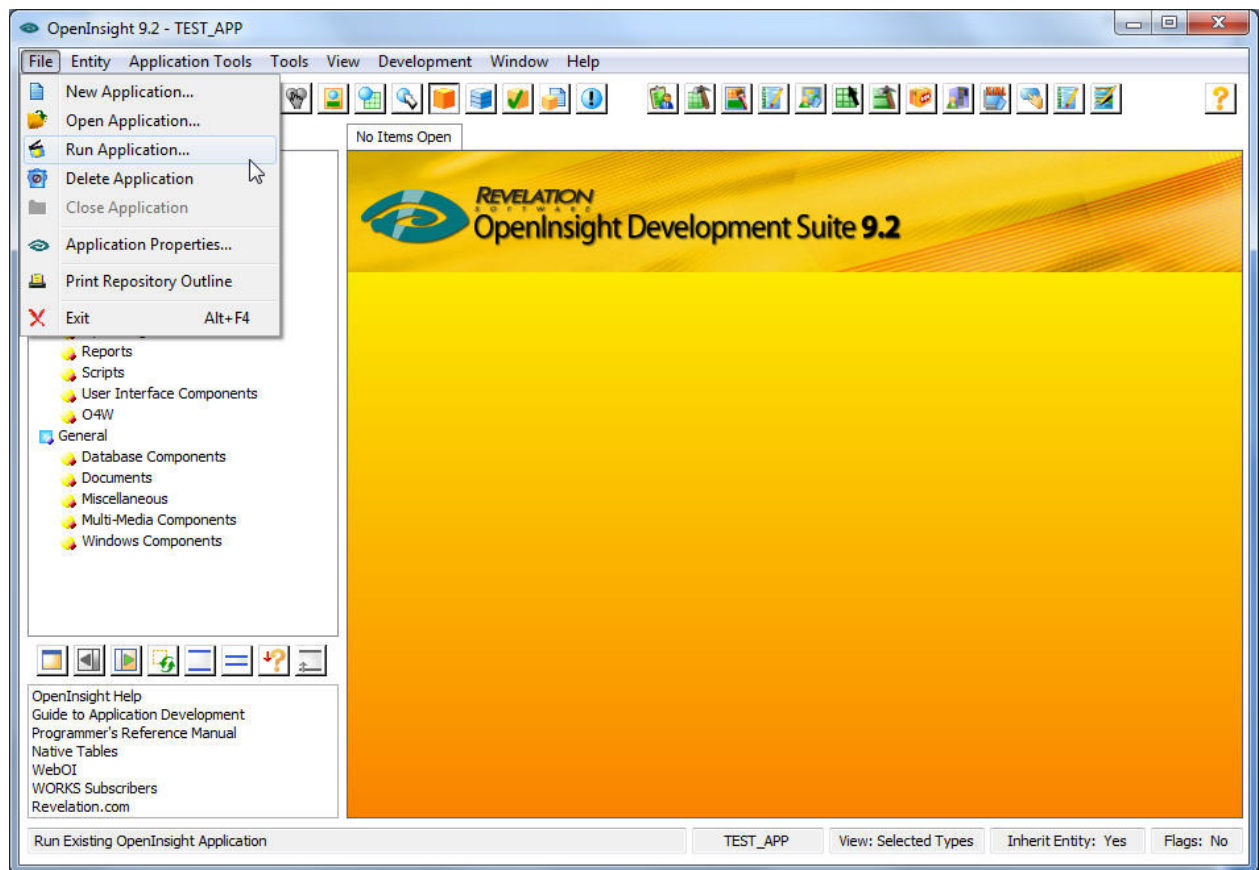
1. Choose *Application Properties* from the Application Manager *File* menu to display the *Application Properties* dialog box shown below.



- Click the **SETUP ENTRY POINT** button (in the bottom left hand corner of the dialog box). The *Setup Application Main Entry Point* dialog box is displayed, as shown below.



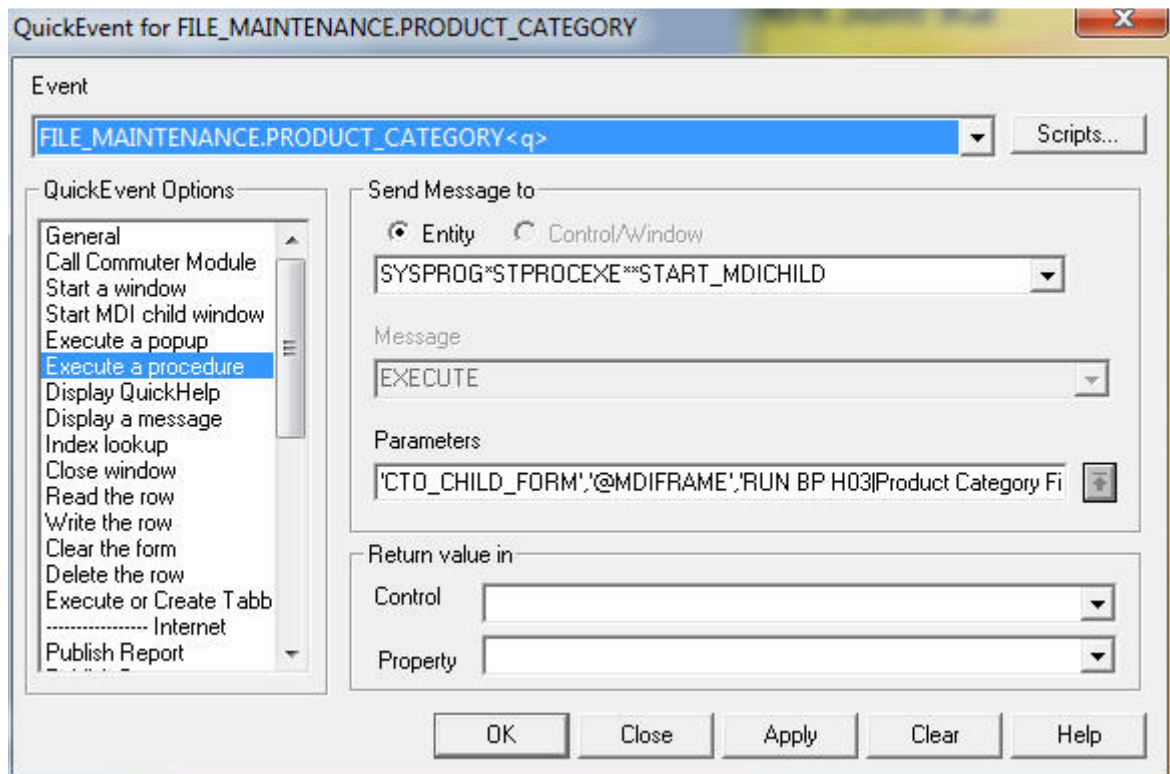
- Clear the *No Application Entry Point* check box. The window names in the *Window Name* list box become enabled.
- Select the form **TEST_APP_MDI** in the *Window Name* list box.
- Click the **OK** button to update the application Entry Point option and return to the *Application Properties* dialog box. Click the **OK** button to close this dialog.
- You will need to close and reopen OpenInsight to the **TEST_APP** application. Now choose *Run Application* from the **File, Application** menu. Your application will now launch.



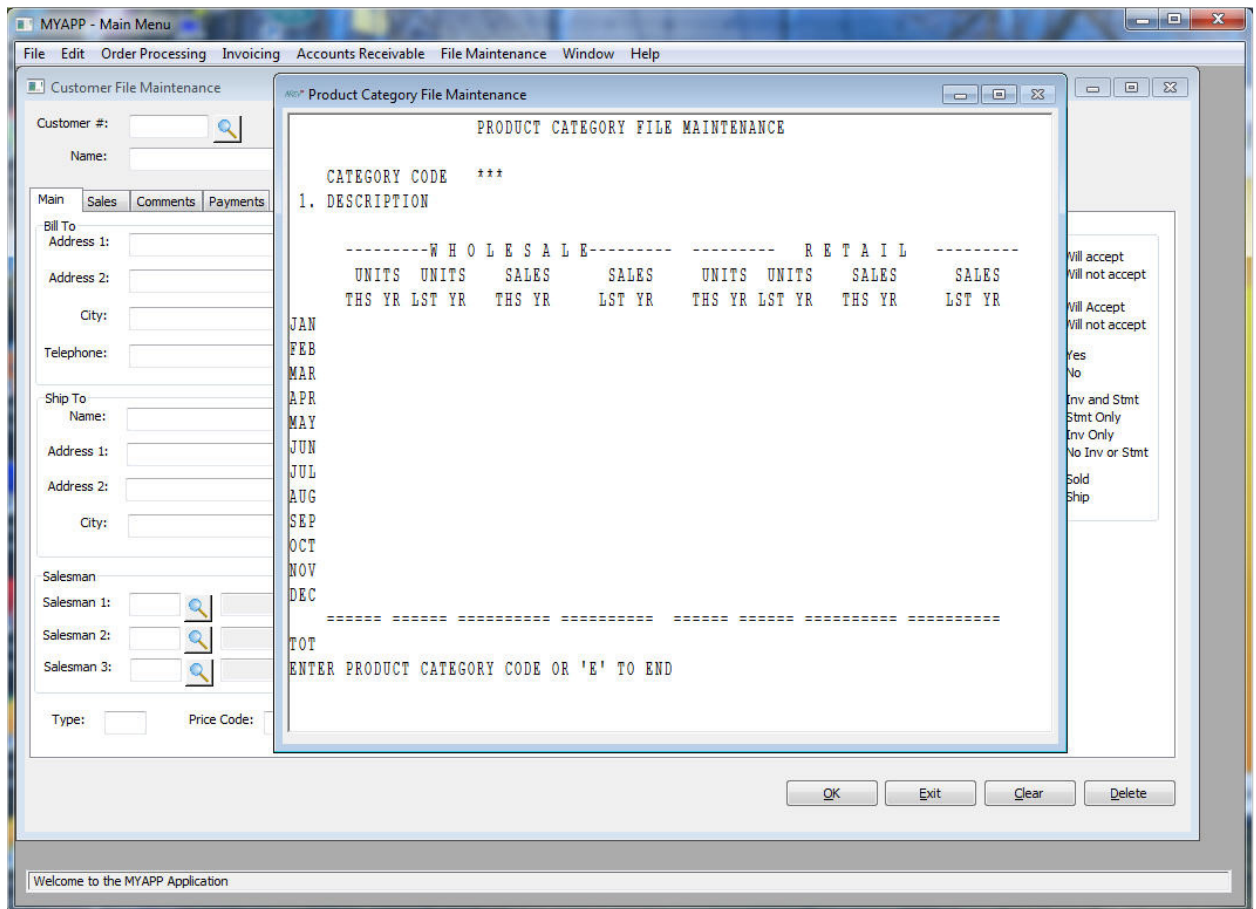
7. Select the TEST_APP application and enter TEST_APP in the User Name edit line, then click the OK button to execute the TEST APPLICATION.
8. The TEST_APP_MDI window is displayed in runtime mode.

I. Launching a character form from a GUI menu

1. From the Form Designer open your MDI Client Window, go to Menu, Quick Events and open the Quick Event for the program you want to call. Choose Start MDI child window. In the Parameters section, replace ChildName with 'CTO_CHILD_FORM' and CreateParam with the command to run your program followed by the pipe symbol and then the name to appear as the form title.



2. You can now launch multiple instances of your character and graphical forms within your MDI Frame.



3. You can add a GUI popup to your character-based program. In the source code of your pram add code similar to the following:

OpenInsight Character Interface: MYAPP

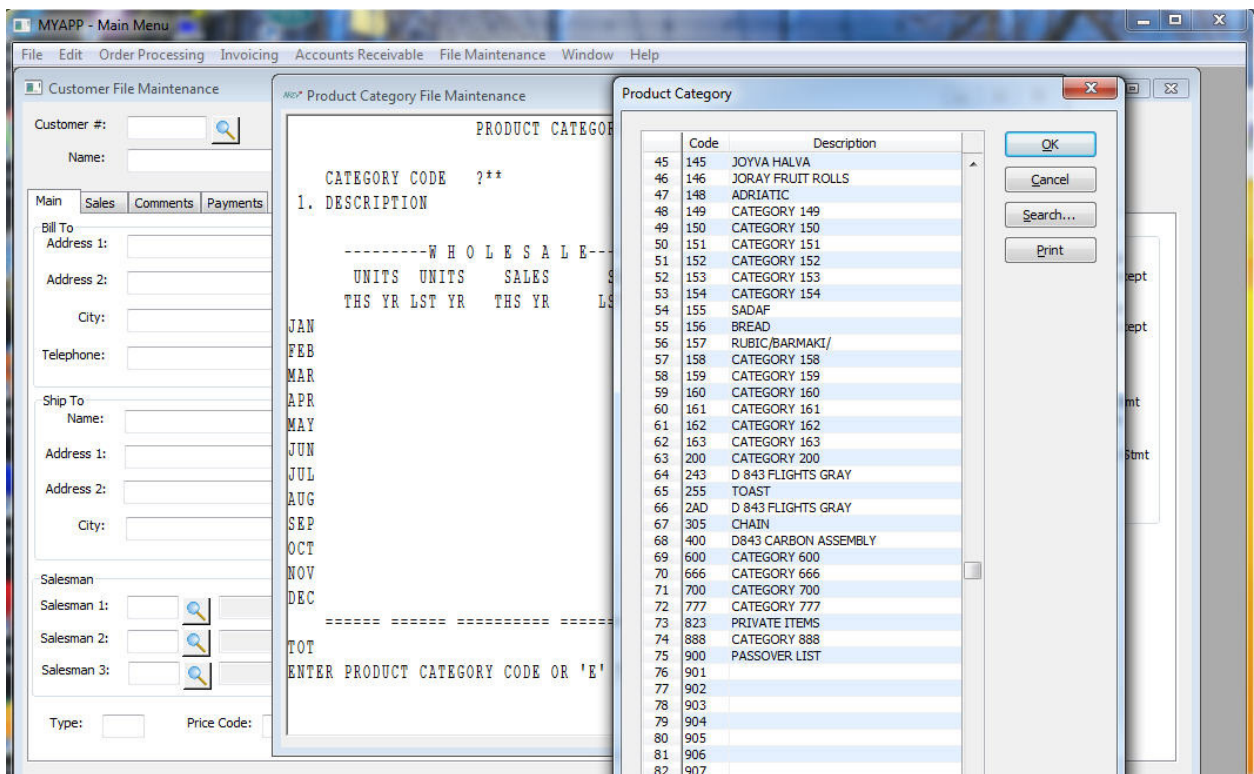
File Edit Colors

```

.P
00021     NEXT I
00022     PRINT @ (0,21): "TOT"
00023     PRINT @ (4,20): "=====
===== "
00024 *
00025 005 PRINT ENTLIN: "ENTER PRODUCT CATEGORY CODE OR 'E' TO END"
00026     PRINT @ (20,2): "***      ": @ (20):
00027     INPUT CO
00028     IF CO='?' THEN
00029         DECLARE FUNCTION CTO_POPUP
00030         CO=CTO_POPUP("", "CATEGORY")
00031     END
00032     IF CO='E' THEN GO 999
00033     IF LEN(CO)>3 THEN GO 005
00034     PRINT @ (20,2): CO'L#3'
00035 *
00036 015 ID=CO
00037     READU GREC FROM CMF, ID ELSE GO 020
00038     GO 300
00039 * ADD ROUTINE
00040 020 PRINT ENTLIN: "DO YOU WANT TO ADD THIS CATEGORY (Y/N)? - * ": @ (42):
00041     INPUT OP
00042     IF OP='N' THEN

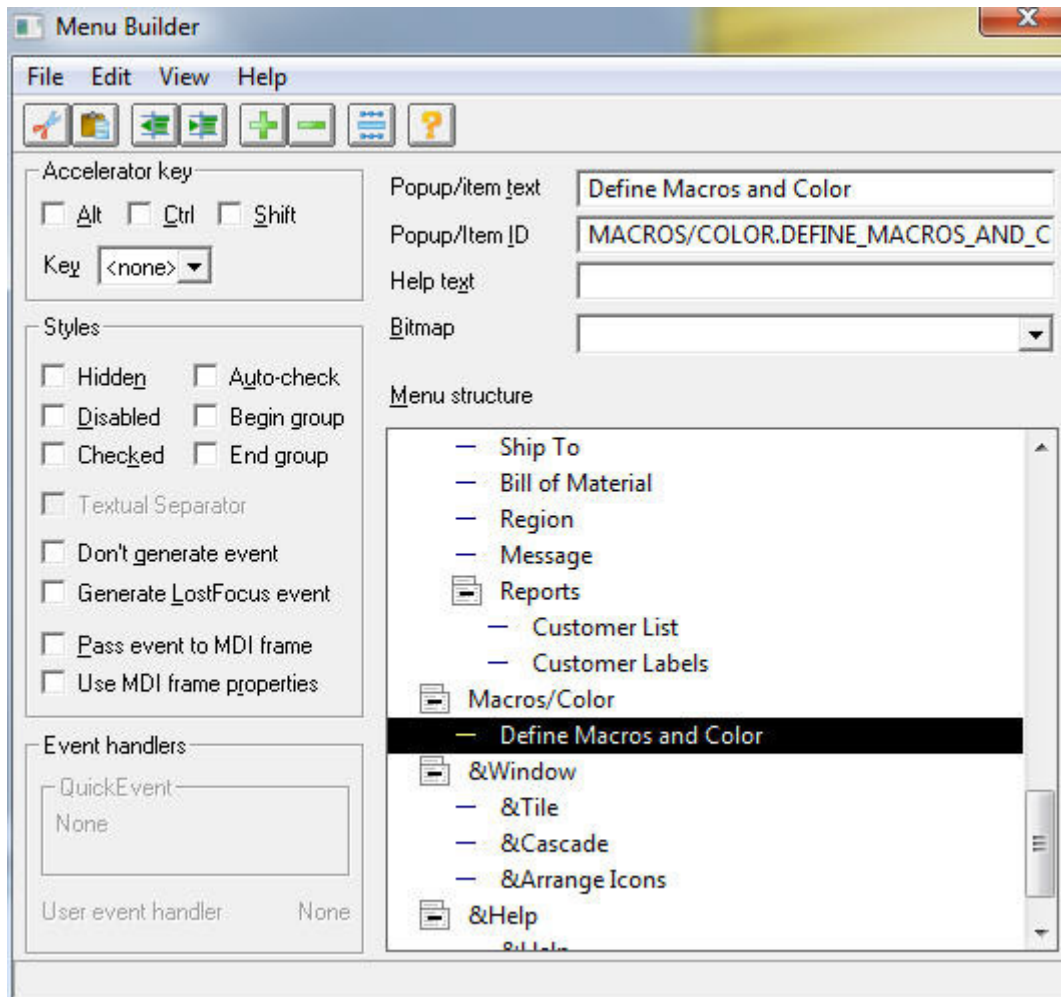
```

4. Your character-based form can now call graphical popups.

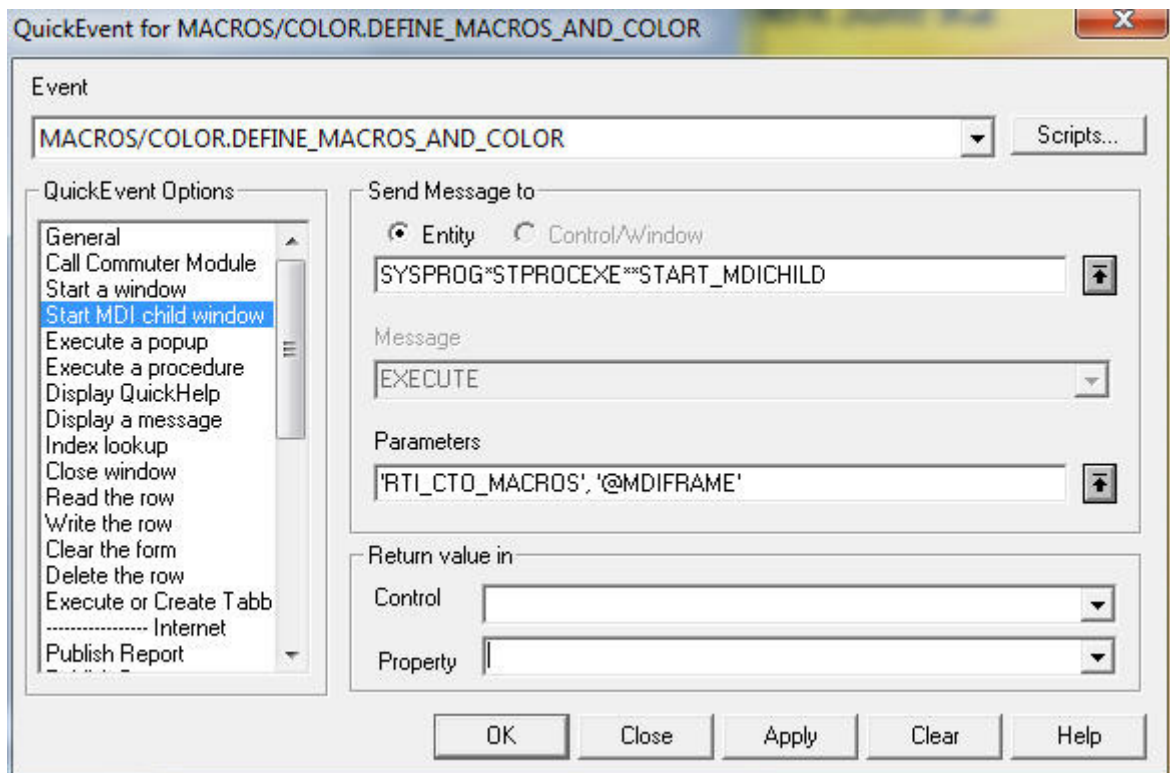


II. Adding an option to define macros and colors to your menu

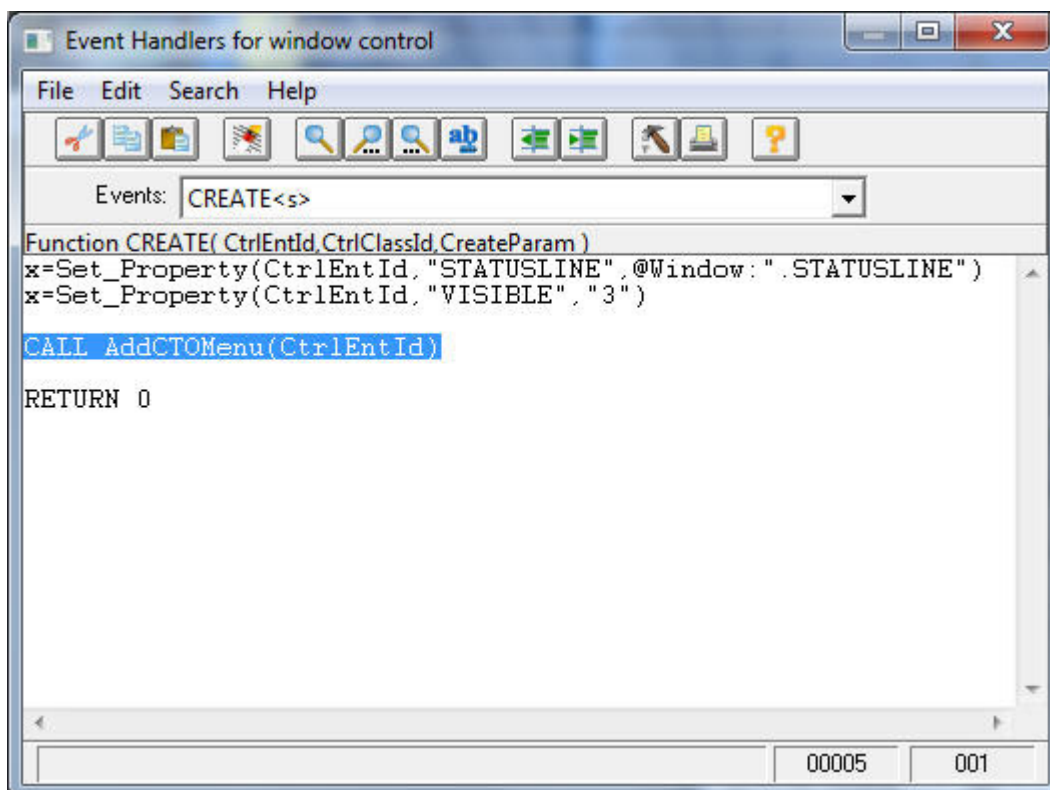
5. With CTO you can add a menu entry to Define Macros and maintain the foreground and background colors of the CTO_CHILD_FORM. From the Form Designer open your CONTACTS menu and add the following to the entry to the Menu Builder.



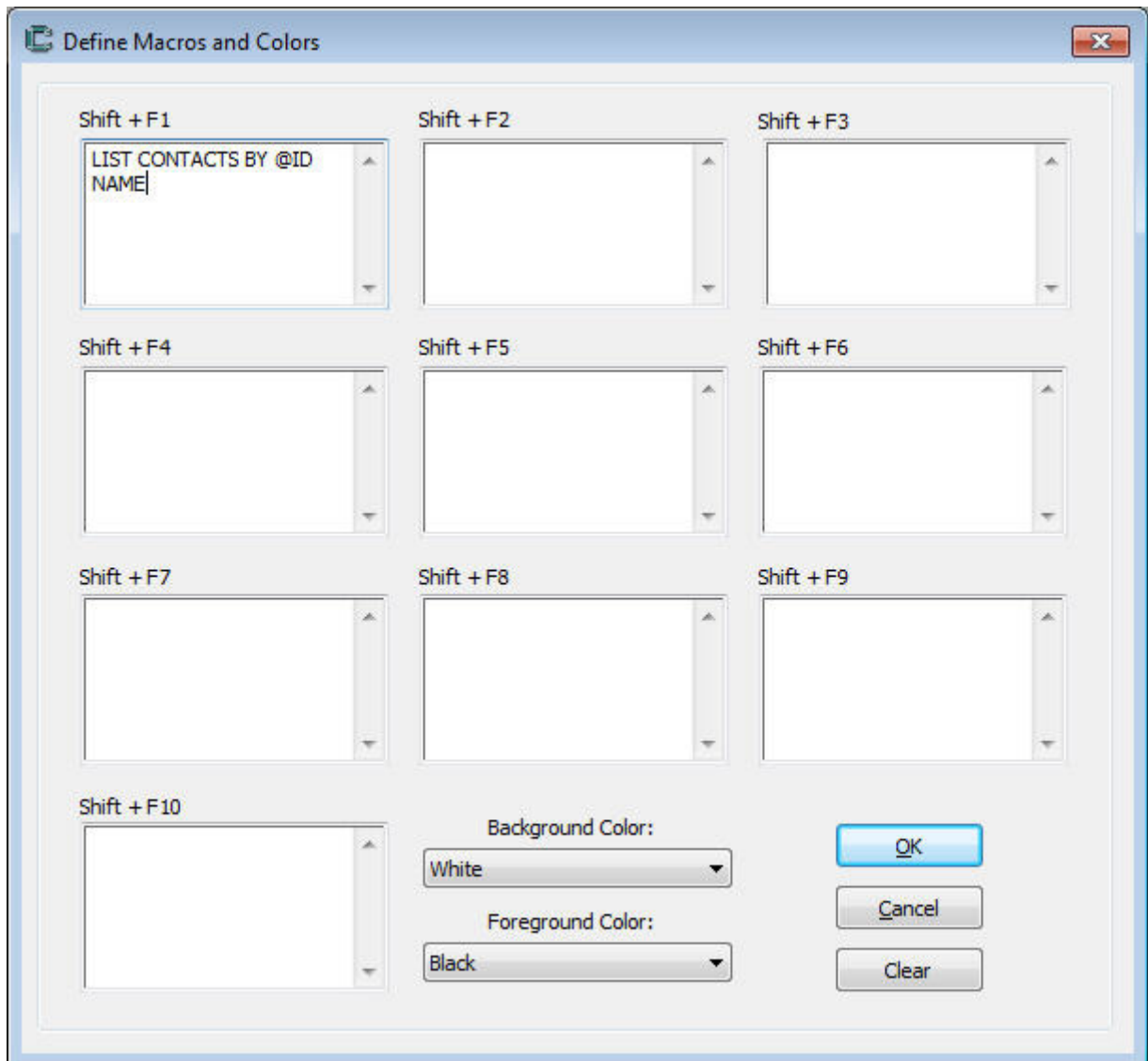
6. Add the following Quick Event for Macros/Color.



7. Add the following line of code on the CREATE event of the form.



8. The Define Macros and Color option on your menu will load the following form.



The image shows a dialog box titled "Define Macros and Colors". It contains ten text input fields arranged in a grid, each associated with a specific keyboard shortcut. The first field, for "Shift + F1", contains the text "LIST CONTACTS BY @ID NAME". The other fields are empty. Below the grid, there are two color selection options: "Background Color:" with a dropdown menu set to "White", and "Foreground Color:" with a dropdown menu set to "Black". To the right of these are three buttons: "OK", "Cancel", and "Clear".

Shortcut	Macro Text
Shift + F1	LIST CONTACTS BY @ID NAME
Shift + F2	
Shift + F3	
Shift + F4	
Shift + F5	
Shift + F6	
Shift + F7	
Shift + F8	
Shift + F9	
Shift + F10	

Background Color: White

Foreground Color: Black

OK Cancel Clear

Congratulations! You've successfully completed your first CTO application using OpenInsight. And what you've learned here is just the beginning. You may continue by taking a look at the sample applications that are included with OpenInsight, or begin your own development project. Whatever your choice, we wish you continued success using OpenInsight.

REVELATION

S O F T W A R E

Revelation Software, Inc
99 Kinderkamack Road, First Floor
Westwood, NJ 07675
U.S.A
Toll Free: 800-262-4747
Phone: 201-594-1422
Fax: 201-722-9815
www.revelation.com

Revelation Software Ltd.
45 St Mary's Road
Ealing,
London, W5 5RG
U.K.
Phone: +44 0 208 912 1000
Fax: +44 0 208 912 1001
www.revsoft.co.uk

Software Australia Pty Ltd.
PO Box 300
Brookvale, NSW 2100
Australia
Phone: +61 2 8003 4199
Fax: +61 2 9332 6099
www.revelationsoftware.com.au

Revelation Software is a division of Revelation Technologies, Inc.

Part No 103-963