

REVELATION

S O F T W A R E

Converting Applications from
Advanced Revelation to OpenInsight:



Perspectives from Developers

201 Broadway • Cambridge, MA 02139 • Phone: (617) 577-0300 Fax: (617) 494-0008

www.revelation.com

A Division of Revelation Technologies, Inc.



INTRODUCTION

Since our inception in 1987, Revelation Software has been dedicated to delivering a fully integrated application development environment that is optimized for local area network and workgroup computing environments. Advanced Revelation is an award-winning environment that provides a complete set of non-procedural application development tools, a robust programming environment, and a flexible, variable length, multi-valued database that enable security and performance on large multi-user networks.

In response to the explosion of demand for graphical user interfaces using Microsoft Windows, in 1993 Revelation delivered OpenInsight, a complete Windows compliant development environment. The objectives with the delivery of OpenInsight were:

1. Provide an uncompromised development environment that completely supports Windows user interface standards, event-orientation and object properties.
2. Deliver an environment that was language and database compatible with Advanced Revelation. This capability was required to enable our customers to leverage much of their prior investment in Advanced Revelation applications and bring them forward into Windows.
3. Provide a breadth of tools comparable to Advanced Revelation. OpenInsight provides Windows versions of a forms painter, popup builder, R/List, messages, help, and QuickEvents (equivalent to the "Code and Command" concept)
4. Set the stage for fulfillment of our mission of delivering a new generation of application development tools that enables iterative application development through repository technology; providing version control, check in/out, object browsing, and automated application distribution.

Revelation also recognized that technologies have advanced and matured rapidly to where PC's are becoming less computing devices and more communication devices. The next generation of Information Systems will leverage these technologies to deliver well integrated workgroup and workflow capabilities. Last year, OpenInsight for Workgroups was delivered to application developers who require the following;

1. Ability to exploit client/server architectures, enabling developers to connect natively to a wide variety of database structures that will be dispersed; SQL Server, ODBC, flat file, and unstructured database formats such as Advanced Revelation files and Lotus Notes.
2. Object oriented methodologies providing the means for reuse and sharing of application components and architecture.
3. Workgroup, workflow, and messaging technologies that enabling business applications to be extended internally and to customers, suppliers, and business partners outside the enterprise.
4. A platform for supporting the Internet and world wide web. As applications become more externally oriented, a new class of non-mandatory or elective users will need to be serviced. Organizations will need to provide richly visual, intuitive systems to those outside the organization to attract them to doing business with their organizations.
5. Repository technologies which provide the means for integrating and managing application components.

Many end users are now requiring that new versions of their applications be graphical and communications based - technologies such as Windows, Lotus Notes, and Internet are being demanded. Many companies have been reducing the many barriers to Windows incrementally; investing in hardware, software, training, and application migration. The purpose of this document is to outline the steps that an Advanced Revelation developer would follow to convert an Advanced Revelation application to OpenInsight for Workgroups cost effectively.

These techniques are outlined in three separate articles written by professional application developers, each of whom have successfully converted a number of Advanced Revelation applications to OpenInsight. The articles begin with "Converting Advanced Revelation Applications to OpenInsight" which describes in detail the appropriate steps to follow in migrating your application to Windows. "Sharing an application between Advanced

Converting Applications From Advanced Revelation to OpenInsight

Revelation and OpenInsight" discusses how you can migrate your application to Windows while still using your Advanced Revelation application. The last article "Preserving you skills: Tools and Techniques in Advanced Revelation and OpenInsight" provides a number of useful comparison charts showing the methods with which to implement similar abilities across the two products.



CONVERTING AREV APPLICATIONS TO OPENINSIGHT

The "good news"

There are many aspects of your Advanced Revelation application that will translate into OpenInsight. Here's what you can convert: data files, dictionaries, indexes, and the basic logic of your programs - and the best news: you don't have to re-enter your data. You will need to evaluate the user interface components of your application and determine how to make these forms, messages and popups Windows compliant.

What doesn't get converted?

The components that communicate to the user must now use the Microsoft Windows user interface standard and must be recreated or evaluated. Remember that Windows is an entirely new environment. It would be a big job (if it's even possible) to convert DOS-based data entry screens and programs to an object-oriented programming style.

It is not surprising that data entry screens are not converted, because the behavior of screens under Advanced Revelation and OpenInsight is completely different. It is not that OpenInsight is different, it's Windows itself. In Advanced Revelation there is a controlled program flow throughout an entry screen. The programmer knows what's happening, and where it's happening. In Windows it's not that simple. OpenInsight, like any other Windows application, allows the user to enter and exit fields on a screen at will. A logical progression of prompts is removed and, therefore, so is straightforward program flow.

Some of the ideas are the same, however. For example pre-prompt and post-prompt hooks are almost the same as the *gotfocus* and *lostfocus* events that you can capture in any object in the window.

On to the conversion process...

Before we discuss how to convert the user interface components of your application, lets first examine what you need to consider before converting:

1. Decide whether your OpenInsight application will share tables with Advanced Revelation. If your application will not share data, copy your data tables into a different sub-directory for access by OpenInsight only. If you don't do this, you may end up sharing tables whether you want to or not. If you want to share the data, please see the following article entitled "Sharing an application between Advanced Revelation and OpenInsight"
2. OpenInsight does not support periods in table names. You can convert file names while still in Advanced Revelation by running Update Application (in the SYSOBJ table), choose Tables, and then choose "Convert Dots to Underscores".

Phase 1. Create a new application

Login to OpenInsight with the user name of SYSPROG and choose New Application from File menu. Fill in the Application ID name (no spaces), Title (a description), Author Name, Password, and Company if you like. Since we are not basing this application on an existing OpenInsight application, leave <Default Inheritance> alone. Click OK and the application will be created.

Phase 2. Access AREV Tables

If your application is not open, choose Open Application from the File menu. Choose Database Manager from the Tools menu, then click File, Add. Make sure "Drives" and "Directories" are pointing to the correct location of your

Converting Applications From Advanced Revelation to OpenInsight

AREV tables, then choose the data tables you want added from the Tables box by clicking the Table name and clicking Add. If you want everything click the Add All button.

The tables you selected should now be listed in the Add Tables box. Click OK and then choose Save from the Database menu. You have now "ATTACH"ed your data by setting an image within OpenInsight pointing to your Advanced Revelation tables in the location you specified.

Phase 3. Create menus

Now create an MDI Frame in the Form Designer that will act as a menu. On this form you can place buttons, bitmaps, icons, and a menu that your users can click on to navigate through your application. This is just one method of getting your forms, windows, reports, etc. to run.

To build an MDI Frame, from the Application Manager, choose Form Designer from the Tools menu and then choose New MDI Frame from the File menu. Since this is your first Form, you don't have a style sheet defined, so click on OK to continue. You should see a default menu and an "MDI Client Window" which serves as the workspace in which your data entry forms, popups, and messages will appear. In most Windows applications you should allow the majority of the MDI Frame to be filled with the "MDI Client Windows" except for an area at the top of the frame that will provide a space for a toolbar. Toolbars can be created by dropping a series of equally sized buttons onto the frame.

To customize the default menu, choose Menu, then Design. At the " Menu Popup/Item" prompt, type in the name of the first selection for your menu. This is the name that appears on your menu as a choice for the user. Press [Enter] and you'll be moved to the next level under your choice to define the submenu. To move back to the top level, press the left arrow key and type the next main selection of your menu.

If there is a blank entry at the end when you're finished, delete it and choose Exit/Update from the File menu. Now double-click the window background and fill in the system name for your menu and a Title to display to your users. Choose Save from the File menu, then exit back to this menu once we have forms or windows to attach to the menu selections.

Phase 4. Set up the entry point into your application

The entry point is the name of the form you wish to display when a user first logs into your application. To specify the appropriate form, go back in the Application Manager choose Application Properties from the File menu and click Setup Entry Point. A list of windows will be displayed and choose the name of the Menu window you created. Click OK twice, and your entry point is set.

Phase 5. Create forms

If you're building a form from a table you've already added (as we did earlier), skip to the next paragraph. Otherwise, in the Application Manager choose Database Manager from the Tools menu, then choose Add from the File menu. Add the table you want to build your forms with to the Tables list box and click OK.

In the Form Designer choose QuickForm from the File menu. Double click the table you want and choose the fields you want. You can choose Add All, but if there are too many fields to fit in your displayed window space, it's tough to get to them after the form is built. So just add a few of your fields for now, then go back and add the rest later. You'll have to put on your artistic hat and make your form look polished by rearranging and resizing your controls. Save your form.

Phase 6. Let's put it all together

You're still in the Forms Designer, so choose Open from the File menu and open up the menu window you created. Choose Menu, then Design from the Forms default menu and click your first menu selection. Choose QuickEvents

Converting Applications From Advanced Revelation to OpenInsight

from the Events menu and in the Summary of Event Handlers dialog box, select your menu item again and click the QuickEvent button. Choose Execute a Window from the QuickEvent Options and in the Send Message To section click the drop-down arrow of the entry field, then select the form you would like to display. Click OK, then close the Summary of Event Handlers dialog box and repeat the process for all your menu items.

Remember, as with all window controls, you can put "&" in front of any character in your menu item to add Alt-key access (Marked with an underscore) to the selection. You can also assign accelerator keys to your menu items in the Menu Designer.

Phase 7. Change your R/BASIC programs

Most of your code remains in OpenInsight, but Windows is a different world, so there are some changes. Specifically, implicit formatting is not supported, so you would have to write:

```
x = FMT('name', 'L#40')
```

instead of

```
x = 'NAME' 'L#40'
```

Generally some user interface code must be rewritten because routines like MSG are different and others like INPUT.CHAR and CATALYST (replaced by QuickEvent) are not applicable in windows. Commands like PERFORM, EXECUTE, and PCPERFORM are now handled by the UTILITY function and the WRITE statement must now contain a full THEN/ELSE clause. Table 1 is a list of unavailable R/BASIC commands and the corresponding Basic+ commands.

General Comparisons

Data tables

No change between Advanced Revelation and OpenInsight.

Dictionary tables

AREV 3.1 and above, no change. Prior versions below AREV 3.1, the dictionary will not carry flags that indicate exclusive use by OpenInsight or Advanced Revelation. If you plan to share data, upgrade to AREV 3.1.

Indexes

Advanced Revelation 3.0 and above, no change. Below 3.0, remove all indexes and reinstall them under OpenInsight.

MFSs

Advanced Revelation 2.0 and above, no change. Below 2.0, refer to Technical Bulletin #43, "Modifying MFSs and BFSs for the 2.0 BFS specification."

Environment records

Definitions for the environment are now attached to the database, not the user. Most of the other utility work such as printer definitions are configured via Microsoft Windows.

Menus

As you've seen, menus are attached to forms under OpenInsight.

Data entry windows

New terminology: data entry windows with tables attached are called forms. What we know as "collector" windows are also called forms. The only difference is whether a table is assigned to a form.

Converting Applications From Advanced Revelation to OpenInsight

Popups

Most of the Advanced Revelation pop up features have been reproduced and extended in OpenInsight 2.6 and OpenInsight for Workgroups.

Help

Application Helpnotes is a system-wide facility quite similar to the AREV help system.

Reports

Report Designer is the main reporting tool in OpenInsight and provides drag & drop, banded, two-pass reporting. Since this is a new tool, all reports using Report Designer will need to be built from scratch. Revelation has also provided "Report Builder" which replaces EasyWriter to create R/LIST reports. OpenList replaces R/LIST. QBF replaces QBE.

Filters

You need to implement filters through your own Basic+ stored procedures

Messages

The parameters you pass MSG are quite different, and MSG now calls the Windows MsgBox API. The capabilities of MSG are still the same, and there is a Message Builder to assist you in creating messages boxes.

Assembly language or C routines

These must be DLLs called from your Basic+ programs. The OpenInsight Technical Reference Manual has more information on DLLs.

Security

Security is implemented on a component basis, which means you can control who has access to each and every window, form, report, table, and so on.

Source and object code location

No matter where you are editing your source, upon compilation the source code is written to the SYSPROCS table and the object code is written to SYSOBJ. Therefore there is no need to catalog anything; OpenInsight already knows where it is.

General equivalents between AREV and OI

Table 1 documents R/Basic commands that are not applicable in OpenInsight as well as new commands introduced. Table 2 is a comparison chart between Advanced Revelation Terminology and OpenInsight.

Converting Applications From Advanced Revelation to OpenInsight

Table 1

Removed R/Basic Commands	New Basic+ code
@ GET_STATUS	GET_STATUS
ABORT GETPOINTER	GETPOINTER
BREAK KEY GETVALUE	GETVALUE
CHAIN LOCKVARIABLE	LOCKVARIABLE
CLEAR SET_FSERROR	SET_FSERROR
CLEAR COMMON SET_STATUS	SET_STATUS
CLEARDATA UNLOCKVARIABLE	UNLOCKVARIABLE
DATA	
ECHO	
EXECUTE	
EXPENDABLE	
FIELDCOUNT*	
GETBREAK (GETECHO, GETPRINTER, GETPROMPT)	
GET.CURSOR	
INMAT()	
INP	
INPUT	
OCONV(MOX)	
OUT	
PCPERFORM	
PERFORM	
PRINT**	
PROMPT	
PUT.CURSOR	
STOP	
SUSPEND	

*No longer an opcode; replaced by the function FIELD_COUNT, which must be declared and compiled

**Unless used in conjunction with PRINTER ON

Table 2

Advanced Revelation	OpenInsight
Data tables	Data tables
Dictionary tables	Dictionary tables
AMV (associated multivalued) group	Edit table
Boolean field	Check box, Radio button
Standard prompt	Editline
Collector window	Window (non-database-aware)
Label	Static text
Multivalued field	List box
Indexes	Indexes
MFSs	MFSs (Basic+ stored procedures)
Assembly or C routines	Rewrite to DLL's outside of OpenInsight
Applications, locations, users	Applications, databases, locations, users
Environment records	Database environment, Windows
Menus	Menus (defined in the context of a form)
Data entry windows	Forms/windows (collectors)
Messages	OpenInsight MSG
Popups	OpenInsight popups
Reports	OpenList, Basic+ stored procedures
Filters	Basic+ stored procedures
R/Basic programs	Basic+ stored procedures
Help	OpenInsight Application HelpNotes
User Levels (Security)	Entity-level security, Basic+ programs



SHARING AN APPLICATION BETWEEN ADVANCED REVELATION AND OPENINSIGHT

by Michael Ruane

Michael Ruane is a principal at **WinWin Solutions**, a consulting firm specializing in Advanced Revelation and OpenInsight. Mr. Ruane can be reached by telephone at (201) 722-9814 and via email at info@winwinsol.com.

What makes OpenInsight so special is the fact that it can use the same data, dictionaries, and indexes as Advanced Revelation, even at the same time! In fact, most of the R/Basic code written in the Advanced Revelation can be used in OpenInsight with little or no modification. In this article I'll show you what you need to do and watch out for to have your Advanced Revelation and OpenInsight systems share the same data.

Before I discuss the particulars of sharing an application between Advanced Revelation and OpenInsight, I should discuss a few reasons you would share an application instead of converting it completely.

Change Incrementally

Making the switch from a character-based application to a Windows-based application is a monumental paradigm shift. Users who aren't mouse-proficient or who are just adverse to change will probably have a difficult time with the new system. But if you share an application between Advanced Revelation and OpenInsight, the change can come gradually - one or two data entry screens at first, then maybe a report. This slow migration allows users to become acclimated gradually, making it easier for them to accept the new system.

Use existing knowledge

From a developer's point of view, programming concepts in OpenInsight are similar to those in Advanced Revelation. Many Advanced Revelation features - pre-prompt and post-prompt processing, pre-save and post-save processing, options, and the message subroutine (MSG) - are all available in OpenInsight, but are slightly different because of the Windows paradigm shift. Sharing parts of your application between the two products will allow you to learn the lay of the land of the Windows world without adversely affecting your existing Advanced Revelation application.

Prerequisites

Before you share an application between Advanced Revelation and OpenInsight, note the prerequisites listed below.

- Microsoft Windows 3.1 or Windows for Workgroups 3.11.
- Either OpenInsight Version 2.5 or OpenInsight for Workgroups Version 3.0. Both are available right now. You can share applications with either version of the product, I recommend Workgroups because of ODBC, SQL Server and Notes capabilities.
- Advanced Revelation version 3.1 or above. This is very important, because only with this version do dictionary items contain a value in field 34, which is equated to DICT_PRODUCT. The values are:

0 or null	Both Advanced Revelation and OpenInsight
1	Advanced Revelation only
2	OpenInsight only

Without this information (and some other changes made to the dictionary structure for 3.1), the data, dictionaries, and indexing cannot be shared. Even though earlier versions of the dictionary might look similar, DICT.MFS will know how to handle the data in field 34. *It is important to remember that the requirement for Advanced Revelation 3.1 is only if you want to share data - if you are looking to do a wholesale migration to OpenInsight and don't care about using the data in Advanced Revelation anymore this requirement is waived.*

- Both applications must be running on a network. The AREV.EXE and OINSIGHT.EXE programs do not talk to each other directly, and so neither acknowledges the other's local lock table. Therefore, both products must use network locking. If you are not on a network or do not have the correct network drivers loaded, you will not be able to lock correctly between the two products.

Use existing hardware

Our clients have PC equipment ranging from the original IBM PC up to the latest Pentium machines. Because Windows doesn't run very well on the 4.77 MHz 8088 chip, clients with those machines won't be able to run either Windows or OpenInsight. It has been our experience that the minimum machine configuration to run OpenInsight and Advanced Revelation together is a 386-33 with 8MB of RAM. Rather than force users to upgrade all their equipment, we can allow them to use the computers they have, and log into Advanced Revelation or OpenInsight as it suits their hardware.

Exploit Windows without a full conversion

If your application needs to pass data to Windows-based applications such as Microsoft Excel or Microsoft Graph, the task is much easier with OpenInsight. Getting data between Advanced Revelation and Windows-based applications is laborious at best, but with OpenInsight it is almost painless. Dynamic data exchange (DDE) and Multiple Document Interface (MDI) are available now, and Object Linking and Embedding (OLE) is coming later this year, opening new avenues of data interoperability. If you share your application between Advanced Revelation and OpenInsight, most users can use the Advanced Revelation system they are used to, while a few users use an OpenInsight application to pass data between the two platforms.

Exploit OpenInsight without a full conversion

A number of tools available in OpenInsight can enhance your existing applications. The new Report Designer in OpenInsight is an incredibly powerful tool, allowing you full OLE capabilities to insert graphics and pictures in your report, use multiple fonts and font sizes, and read Advanced Revelation 3.1 data. Imagine an employee report listing employee information and showing the employee's picture! This is available right now when you share an application between Advanced Revelation and OpenInsight.

Mouse-optional applications

You might want to have two separate front ends to your system, based on whether users prefer using the mouse. Managers, executives, and directors might want point and click their way through the system, starting at the top and drilling down to the data they need. However, data-entry personnel might still want a character-based application for familiarity and speed. Sharing an application between Advanced Revelation and OpenInsight will satisfy both groups of users.

Experiment

Finally, If you want to see how your application would look in Windows, sharing an application with OpenInsight gives you a preview. You can try different types of menus - drop down menus, icons, buttons, and so on - without affecting your existing application. You can re-evaluate your screens and reports, making any changes needed, and take advantage of whatever GUI tools you want.

How to share an application

The steps listed below outline the general procedure for sharing an application between Advanced Revelation and OpenInsight. You'll find that the procedure is easy and relatively painless. In the sections following I'll discuss individual steps in greater detail.

1. Confirm that the dictionaries in Advanced Revelation 3.1 are clean. By clean, I mean the following:
 - No dots allowed (underscores only) in table and column names.
 - Data typing is correct. OpenInsight requires that all dictionary columns have data typing. If you migrated your application from Rev G to Advanced Revelation 3.1, you might still have an R/DESIGN dictionary description in the data field (12).
 - Field contents are up-to-date. Some fields in the dictionary have been used differently in previous versions of Advanced Revelation or Rev G. For example, the single or multi-valued field, SM, at one

Converting Applications From Advanced Revelation to OpenInsight

point contained M1.1 or M1.2, instead of only S or M. Give the dictionary records the once over and make sure all is in order.

2. Make a new application in OpenInsight that uses the same application name as your Advanced Revelation application (account). This is not absolutely necessary, but makes things easier to remember.
3. Add Advanced Revelation tables to the OpenInsight database definition. This is as simple as telling OpenInsight where the Advanced Revelation tables resides; the system does the rest.
4. Create your windows in OpenInsight. You will not be able to port your windows from Advanced Revelation to OpenInsight. However, recreating them is not difficult at all thanks to the tools provided in OpenInsight.
5. Convert your reports to OpenInsight, or create new ones. R/BASIC reports can probably be converted with only minor changes. R/LIST and Easywriter reports need to be recreated or converted because of changes between R/LIST and Report Builder. Don't be confused between Report Builder (a Windows based version of R/LIST) and Report Designer (a new drag & drop, banded reporter writer) which is also provided in OpenInsight.
6. Create new menus in OpenInsight. You have many new options for the look and feel of menus in OpenInsight, including icons, buttons, and drop-down menus.

This list does not cover all the pieces in an application, but does allow you to get to your data, enter and modify it, and access it through reports.

Adding Advanced Revelation tables to OpenInsight

First, create a new application in OpenInsight with the same name as your Advanced Revelation application. Once the application has been opened, follow these steps to access to your Advanced Revelation tables:

1. From the Tools Menu, click Database Manager to display the Database Manager menu.
2. From the Database Manager menu, choose File, then Add. OpenInsight displays the Add Tables window.
3. If it is not already set, set the table type to "OpenInsight tables". Include Advanced Revelation tables.
4. Using the Directory and Drive windows, set the correct path to your application data tables. Once you have set this path, the Tables window will fill up.
5. Individually tag the tables you want to add, or choose Add All to add all of them.

That's all there is to it! Repeat these steps as many times as necessary to add all your data tables. Don't worry yet about the tables containing programs and window.

When you're done with these steps, you can access your data, dictionaries, and indexes in OpenInsight. If you're running on a network, all locking will be coordinated between the two applications correctly.

There are some additional points to keep in mind after you've converted:

- If you create a new table in Advanced Revelation that you want to access in OpenInsight, you must add it manually to the table definitions as described earlier. Otherwise OpenInsight doesn't know that the table exists.
- If you add an indexes to a table in the Advanced Revelation, you must manually add the new index table (one file) to OpenInsight as described earlier.

Creating Windows in OpenInsight

The following describes one way to create windows quickly in OpenInsight. (Actually, they're referred to as forms in OpenInsight).

1. Click Form Designer in OpenInsight's main window. Or choose Form Designer from the Tools menu.
2. Choose QuickForm from the File menu. (QuickForm is a lot like Quick Paint in Advanced Revelation).
3. In the Table box, double-click the name of the table containing the data you want to import. If you don't have too many columns (let's say eight or less), click Add All to add all of the database columns. Otherwise, choose only about eight columns. You can add the rest manually later. I recommend that you limit the number to

Converting Applications From Advanced Revelation to OpenInsight

eight because when you choose more, the prompts and labels display oddly, and it's usually more work to straighten that problem out than it is to add columns manually.

4. Click OK to save the columns you've selected. OpenInsight creates the form automatically, using the column names as text labels.
5. If you want, you can resize and smooth out the window controls. For example, you might want to revise the text labels or move edit line controls.
6. Repeat the process for each of the windows you need to duplicate from your Advanced Revelation application.

Those are the basic steps. There are a few additional considerations if your window uses symbolic columns or contains multiple pages, or if you want to add columns to the dictionary. In addition, because OpenInsight's window processing is very different from that in Advanced Revelation, you must redesign the processing currently handled by commuter modules in Advanced Revelation.

Symbolics

Because of the way OpenInsight performs window processing, @record is loaded when the record is read off disk, and isn't maintained dynamically by the system during the window process. Therefore, symbolics that use @record references will get out of date quickly. There are two workarounds:

- Use the {} operators. These invoke the CALCULATE system stored procedure (SSP) which will update the data correctly.
- Call the CALCULATE SSP directly. This technique is documented on page 61 of the *OpenInsight Technical Reference Manual*.

Multi-page windows

Multi-page windows are now available in OpenInsight. You may use the built in ability provided in the Form Designer, use a tab control outlined in the OpenInsight for Workgroups sample application, or you can devise a convention to call another window for adding more data to the same record. One possible way is to create a button labeled 'Page 2' or 'More', then write event handling code to call the next window. When users click the button, the handler would have to save the record in the current form, unlock the key, and call a new window and pass it the key.

Adding columns to the dictionary

If you use OpenInsight's Window Designer to add a column to a dictionary while Advanced Revelation users are sharing the application, the Advanced Revelation users must re-attach that table. If you add columns by choosing Edit Column Attributes from the Database menu in the Form Designer, the OpenInsight Table Builder recreates the table and assigns it a new DOS file name.

Recreating commuter modules in OpenInsight

Commuter modules in Advanced Revelation will not convert directly to event handlers in OpenInsight, and this is probably for the best. The major reasons they won't work are that *window_common%* is no longer available and that *@record* is no longer maintained dynamically by the system. Any references to those variables have to be changed.

In addition, the idea of using one large program to support a window is still feasible with OpenInsight. Because OpenInsight is event-driven, actions like pre-prompt and post-prompt processing must now be treated as *gotFocus* and *lostFocus* events. Each event can have event handlers associated with it, written in Basic+. An event can also have a QuickEvent associated with it, which is a 'fill in the blank' event handler that can execute one of a number of events such as calling help, displaying a popup, closing a window, or more.

On the surface, it might seem that window processing has gotten more difficult in the event-driven environment. However, it has been my experience that most behind-the-scenes processing has actually gotten easier. You can set values or actions in a form using one command: SET_PROPERTY. This is true for all controls in a form, whether

Converting Applications From Advanced Revelation to OpenInsight

they are text labels, edit lines, edit boxes, edit tables, or whatever else you might have. You no longer have to try to find just the right combination of *display.action*, *reset*, and *amv.action* to make a window behave properly.

Some processing that you are used to doing with commuter modules is now handled automatically by OpenInsight. At the same time, other processing that used to be automatic in Advanced Revelation isn't any longer in OpenInsight. In order to make your windows look good and behave properly, you must really think about how they will be used in the GUI environment. Use buttons instead of softkeys. Use checkboxes or radio buttons for boolean columns. There are a lot of new possibilities - explore them!

Converting R/BASIC reports

A number of R/BASIC commands won't work in OpenInsight's Basic+, as described in the earlier article. The following is a list of differences between R/BASIC and Basic+, which also suggests workarounds for your R/BASIC reports.

Reports must be called from a parent window using an event. An advantage is that this allows you to use dialog boxes, calls to the *MSG()* subroutines, and other user interface elements.

- The *STOP* command is no longer valid; all programs must end with *RETURN 0*.
- All *WRITE* commands require a conditional branch (*THEN/ELSE* clause). This is optional in Advanced Revelation.
- No dots are allowed in table names or column names.
- The *PERFORM* and *EXECUTE* commands are no longer supported. Use the *RLIST()* subroutine instead. For example:

```
CALL RLIST('SELECT EMP BY EMP_NO',5,',')
```

The final three parameters will always be the same. Many Advanced Revelation TCL commands are available in OpenInsight as subroutines, such as *MAKE_TABLE_SUB()* and *DELETE_TABLE_SUB()* - use them when possible. Remember that the *RLIST()* subroutine is not a direct replacement for Advanced Revelation's *PERFORM* and *EXECUTE* - it supports only the OpenList *SELECT* and *LIST* statements. You cannot use the *RLIST()* subroutine to execute windows, popups, or any other Advanced Revelation TCL commands.

- The *PRINT* command works only if a *PRINTER ON* command has been executed beforehand. Even then, because of the way Windows works, the report will not print unless you execute a *PRINTER OFF* command after the report is done.
- You can no longer call collector windows. Instead, use dialog boxes, which you can display using the *DIALOG_BOX()* command.
- The *MSG()* does not work the same way in Advanced Revelation and OpenInsight. For example, there is no "UB" message type. Refer to the OpenInsight documentation for the changes in syntax.
- Instead of displaying a status line gas gauge to indicate activity, change the shape of the icon to an hourglass using the *UTILITY()* subroutine...
- Implied format statements are not supported. For example, this statement does not work in Basic+:

```
Print "John" "L#40"
```

Instead, use the *FMT()* or *OCONV()* functions to format data:

```
Print FMT("John","L#40")
```

- To preview the results of a *SELECT* command, use the *OR_VIEW()* subroutine to list the rows you selected.

Some additional points to consider when moving R/BASIC programs from Advanced Revelation into OpenInsight are these:

Converting Applications From Advanced Revelation to OpenInsight

- If your programs are in the SOURCE table, and it has PROGRAM.MFS installed on it, you must either move the MFS into OpenInsight first, or you must remove the MFS before you add the source table to your OpenInsight application.
- If you are going to use the same programs in both OpenInsight and Advanced Revelation, you might need to determine if you are in Advanced Revelation or OpenInsight from within a program. One way is to check SYSTABLES for the existence of the SYSREPOS table; if it's there, you're in OpenInsight.
- Object code for your programs must reside in the SYSOBJ table in OpenInsight. OpenInsight automatically stores object code in this table when you compile your programs in the System Editor.

Window processing equivalents

This table shows the equivalents between Advanced Revelation's and OpenInsight's processing hooks. For the most part, it isn't too far a stretch to see the relationship between the two groups.

Advanced Revelation	OpenInsight
Window	Event
Pre-prompt	GotFocus
Post-prompt	LostFocus
Options	Options
Pre-, Post-delete	Delete
Pre-, Post-read	Read
Pre-, Post-write	Save
Post-application	Close
Pre-application, Pre-init	Create
Help	Notes, Help
Pre-, Post-refresh	Clear

Remember that when you write event-handling code for pre-processing or post-processing (for example, pre-read or post-read), you must forward the event using the FORWARD_EVENT or SEND_EVENT commands.

R/LIST reports

OpenInsight provides Report Designer, which is a visual, drag & drop, banded report writer. If you use Report Designer (and I recommend you do) you will need to recreate all of your reports. Revelation has provided a tool "Report Builder" that will allow you to use the familiar R/LIST language and reports. If you know R/LIST, you'll find Report Builder very similar, lacking only the COUNT and SORT commands, and differing in other small details.

You can run Report Builder reports the System Monitor, using the RLIST command. The syntax is:

```
RUN RLIST "listStatement", {OPTION}
```

Set the options to zero or null for output to the printer, or 1 for output to the screen. The default destination for the report is the printer, so be sure to specify 1 if you want a screen report, or you'll end up wasting a lot of paper. There are five other options, documented in the OpenInsight KnowledgeBase document.

Other differences include:

- Report Builder does not support in-line formulas.
- Report Builder does not support the use of parentheses to change the precedence of AND and OR.
- Reports sent to the screen do not show headings, footings, or column headings, although these do appear when you send a report to the printer.
- When you run Report Builder, there is no way to terminate the report prematurely.

Converting Applications From Advanced Revelation to OpenInsight

- The SELECT statement is trickier and more involved to use, and appears to work well only when called from programs. Put your filtering criteria into your RLIST statements and you'll be better off.

To view reports on the screen, OR_VIEW is probably the tool of choice. It is a report viewer that allows you to view the pages of the report, and send the output to the printer. Its syntax is:

```
OR_VIEW "reportName" c"RlistCommand"
```

You can use either a report name or a Report Builder command, not both. The reportName parameter is the name of a report that was build using the Report Builder.

Alternatively, use the OR_PRTF command to send your report to a DOS file, which makes it the equivalent to the Advanced Revelation PDISK command. Its syntax is:

```
OR_PRTF "reportName" | "RlistCommand", "dosFileName"
```

As with the OR_VIEW command, you can use either a report name or an RLIST command, but not both.

Conclusion

Sharing an application between Advanced Revelation and OpenInsight is actually quite easy. The people at Revelation have made this task as painless as possible by providing the tools and support to ensure success. By following the guidelines described previously, you'll be able to support users' applications in both Windows and DOS.

Once you've converted your Advanced Revelation applications to Windows using OpenInsight, you'll have made a giant stride toward entering the mainstream. With DDE, MDI available right now, and OLE on the way, data can be taken from and added to our linear hash tables seamlessly, providing us with a path to the real world. Our applications can now talk to, send data to, and receive data from other non-Advanced Revelation or OpenInsight applications easily.



PRESERVING YOUR SKILLS: TOOLS AND TECHNIQUES IN ADVANCED REVELATION AND OPENINSIGHT

by David Harmacek

David Harmacek has been providing complete, custom-designed systems in products from Revelation Software since 1985. He recently added development in Lotus Notes to his repertoire. He can be reached at (978) 256-9922 or daveh@shore.net. Dave offers enhancements, upgrades and conversions of existing Advanced Revelation based systems including advice on conversion to OpenInsight. **Harmacek Database Systems**, 116 Turnpike Rd #7, Chelmsford, MA, 01824.

As an Advanced Revelation developer of long standing (since Revelation F) tasked with developing applications using Microsoft Windows and OpenInsight, I began learning OpenInsight by reviewing the sample applications and the documentation. I saw many similarities between the products, but many contrasts as well. I attended an OpenInsight class and can see the road to knowledge is long, but I feel that diligence and a lot of experimenting will bring success!

I then began to prepare notes comparing and contrasting the two products. This led to a presentation at the Revelation Developer Conference. The following tables are from my presentation. I hope they assist you to build many successful OpenInsight applications. These tables should be use to when you have the question - I used to do it like this in Advanced Revelation, what is the similar method in OpenInsight? These tables are broken into the following; General Comparison; terminology & tools, window processing equivalents, prompt information, equivalent controls, code and command equivalents, Window Common equivalents.

Converting Applications From Advanced Revelation to OpenInsight

General Comparisons between Advanced Revelation and OpenInsight

Feature	Advanced Revelation 3.1	OpenInsight
Terminology	Mostly Revelation Technologies-specific, with evolution from prior versions of Advanced Revelation, Revelation, Prime Information, and Pick.	Based on developed or developing industry standards.
User interface	Has evolved to Common User Access (CUA); Designed for Graphical User Interface (GUI); Microsoft Windows standard	IBM character-based standard
Programming language	Robust; conventional third-generation language with fourth-generation extensions	Report Designer Report Builder (derived from Advanced Revelation); and SQL
Debugger	Rudimentary	Advanced with multiple windows
Repository	None	Included
Windows/Forms	Can be bound to a table; multi-table windows unwieldy; virtual space supported	Can be linked to underlying table; multi-table forms are easy, multipage forms
Popups	Single-selection, multi-selection, read-only, grouping, ordering, and sorting types	Derived from Advanced Revelation; all features plus more!
Message Boxes	Modal, Yes/No, Yes/No/Cancel, Input	Same
Help	Component of Paint	HelpNotes Builder (Windows Help can be launched instead)
Event handling	Hooks to commuter module	Built in, bound to controls
Quick events	Code and command	Built-in standard objects
Indexing	Interoperable with OpenInsight	Interoperable with Advanced Revelation 3.1
Stored procedures	Main program, subroutines, and functions	Subroutines and functions only
Background execution	None	Can be multitasked via Windows
Data storage	Variable-length, multivalued columns in rows	Same, ODBC, Lotus Notes, SQL Server
Data Typing	Optional	Required
Collaborative development	None	Check in and out
Impact Analysis	None	Available via the Repository
Source Code	SOURCE, OBJECT, or any other table	SYSPROCS, SYSOBJ

Converting Applications From Advanced Revelation to OpenInsight

Advanced Revelation window processing and OpenInsight event processing

Advanced Revelation 3.1	OpenInsight
Allows developers to specify codes and commands in the Process window to add special processing. Called subroutines can access and modify <i>window_common%</i> variables. Developers must learn which process is invoked at what place in normal window processing, and which <i>window_common%</i> variable to manipulate in order to achieve a particular effect.	Windows or controls recognize a predefined set of events (window events and control events) for which developers can write code or apply a QuickEvent to create an event handler. Developers must learn which events occur (in what order) and which components recognize which events. Components can be manipulated by setting values for their properties.
Pre-application Post-application Pre-initialize Post-initialize	<i>create</i> window event
Pre-read Replace-read Post-read	Pre-read process Forward_Event (window) Post-read process
Perpetual process	Write code or apply QuickEvents that send messages to a control or a repository entity. The recipient responds to a message by executing the code for that message.
Pre-save Replace-save Post-save	Pre-write process Forward_event (window) Post-write process
Pre-refresh Replace-refresh Post-refresh	Pre-clear process Forward_event (window) Post-clear process
Pre-delete Replace-delete Post-delete	Pre-delete process Forward_Event (window) Post-delete process

Converting Applications From Advanced Revelation to OpenInsight

Features of Advanced Revelation prompts and OpenInsight controls

Advanced Revelation 3.1	OpenInsight
Unbound prompts that directly reference <i>@record</i> variable	Similar, do not reference <i>@record</i> variable values are place in <i>text</i> property of the control
Joined prompts	Controls are individually bound to a column in a table
Directly update <i>@record</i> variable	Update <i>text</i> property of the control; <i>@record</i> resolved upon <i>calculate</i> or <i>write</i> event
Symbolic prompts	Yes
Text justification	If relevant to control
Upper case conversion	If relevant to control
Edit masks	Change <i>text</i> property
Tab stops	Yes
Required	Yes
Protected	Set <i>enabled</i> property
Filled from default	Yes
Limit length or depth	Edit table
Default value	Yes
Smart labels	Set <i>text</i> property of a static text object from the window <i>create</i> event
Validation patterns	Yes
Uniqueness in multivalued prompt	Yes
Row exists in table	Yes
Output conversion	Yes
Options	Yes, as [ALT-DownArrow]
Index lookup	Yes
Pre-prompt code command	<i>gotFocus</i> event
Post-prompt code command	<i>lostFocus</i> event
Post-prompt code command	<i>changed</i> event
Help	<i>help</i> event
<i>@record</i> < <i>n</i> >	<i>text</i> property
Prompt registers	User Defined Properties (<i>@propertyName</i>)
Invalid	No

Advanced Revelation windows and OpenInsight forms

Advanced Revelation 3.1	OpenInsight
Application windows (bound to a table)	Form
Collector window	Form
Border type	Windows standard
Menu	Windows standard
Softkeys	Substitute push button control or menu item
Exploding window	None
Default mask	Form preferences for font or color
Registers	Create user defined property in the form
Related windows	Use <i>event</i> code
Disable keys	Set <i>enabled</i> property
Concept help	<i>notes</i> event in Form property
Security	Set via entity properties
Row locking	Same

Converting Applications From Advanced Revelation to OpenInsight

Equivalent types of controls in Advanced Revelation windows and OpenInsight forms

Advanced Revelation 3.1	OpenInsight
Labels	Static text
Softkey	Command button, menu item
Single-selection popup	Option button (radio button), list box
Box drawing	Group box
Multivalued prompt with multi-selection pump	Check box or multiple selection list box
Associated multivalued prompts	Edit table
Single-valued prompt	Edit line
Multi-lined single-valued (text) prompt	Edit box
Single-valued prompt with single-selection popup	List box, drop-down list box
Single-valued prompt with single-selection popup	Combo box
Boolean prompt	Check box

Non-procedural processes in Advanced Revelation and OpenInsight

Advanced Revelation code command	OpenInsight equivalent
B Index lookup	Index lookup QuickEvent
V Display values from an indexed column	None
P Display a popup	Execute Popup QuickEvent
M Display a menu	Menu always displayed
W Display a window	Execute Window QuickEvent
H Display help store in SYSHELP or the HELP table	Display Note QuickEvent
HD Display help from the Description column in a dictionary row	None
HF, HI, HM, HMD, HT display other types of help	None
E, X Run TCL commands	Use Utility function
S Call a cataloged R/BASIC subroutine	Execute Procedure QuickEvent
{, C, @, F, K, I	None

Other comparable processes in Advanced Revelation and OpenInsight

Advanced Revelation	OpenInsight
WC_RESET%=RESET.WINDOWS	Close Window QuickEvent
READ @record FROM WC_SRC_FILE%, @ID	Read Row QuickEvent
WRITE @record TO WC_SRC_FILE%, @ID	Write Row QuickEvent
WC_RESET%=RESET.RECORDS	Clear Form QuickEvent
DELETE WC_SRC_FILE%, @ID	Delete Row QuickEvent
Window Query (“\” in window)	Query by Form
Call MSG subroutine	Display Message QuickEvent [<i>see immediately below - Ed.</i>]