The Real Cost of Conversion to Xpa

10 Forgotten Factors to Consider

When Calculating the True Expense of

Upgrading from Magic to XPA

Abstract

Decision-makers considering whether or not to upgrade their Magic applications to Xpa often neglect to consider a number of factors that can drastically impact their project's final cost. This white paper examines 10 of the most significant factors with their associated hidden costs, providing a guideline for calculating an estimate of what a conversion from Magic to Xpa will realistically cost.

Table of Contents

The Problem with "Free"

MSE provides a couple of free tools that convert Magic 9.4 applications to Xpa (however, these tools will not help you with any Magic applications from v3.5 – v8.3). Since these tools are free and come from MSE, there's a tendency to believe that in a matter of just a few hours, an organization should have no trouble doing its own conversion. After all, Magic developers that wrote the code are already on staff & know the application better than anyone.

In our extensive experience though, we've found time & again that even the most senior staff members can be utterly unprepared (and even shocked) by the sheer volume of tasks which must be properly executed in order to carry out a successful conversion, *the first time*.

Converting a Magic application to Xpa correctly the first time can make all the difference between a project that meets expectations, deadlines, and budgets, versus one which ends up having a lot of unexpected rework, unmet deadlines, and excessive cost overruns.

In the end, the old adage about there being no such thing as a free lunch will be proven true once again.

10 Forgotten Factors (Hidden Costs)

After having successfully converted over 1,000 Magic applications to Xpa for about 75 clients, our engineering team assembled a list of the top 10 most costly forgotten factors organizations tend to neglect (or discover too late) when embarking upon their conversion project. This list is by no means exhaustive, but it represents the factors that we've observed tend to cost organizations the greatest expense in rework.

1. **V9 Compatible Objects**
   Using MSE's v9converter tool will cause some of the objects in your application to be implemented as v9 compatible. These objects disable some Xpa features and, tend to remain stuck in your application, forever.
   Additionally, v9 compatible objects will NOT work in XPA.

2. **RM (Record Main)-Converter**
   The RM Converter, another free tool provided by MSE, knows how to handle basic, simple Record-Main operations. However, when it encounters more complex Record-Main operations with block groups, error and warning messages, parking conditions, and more advanced code, it leaves your converted application with altered behavior which can crop up in unexpected places. This free tool also has many problems converting a v9 application that has handlers with conditions.

3. **Link Validate**
   Link Validate is no longer supported in Xpa. MSE's free tool automatically adds a return code with an error message for each Link Validate in your application, even if it already exists in your v9 code. Additionally, the free tool does not handle the issue of Link Success codes used more than once in the Record Main. Also, the error message is constant and the way it's implemented in the converted code could cause downstream problems if your application gets migrated to RIA in the future.

4. **Child Windows and Sub-Forms**
   Xpa provides a new Sub-Form object, but the free tool does not handle the child windows nor build Sub-Forms in the converted application. After converting your application, you will probably need to invest an inordinate amount of time finding and fixing all the child windows and/or replacing them with Sub-Forms. If this manual effort is not expended, the child windows in your application will not work.

AUTOMATED CONVERSION TECHNOLOGIES OF INFORMATION SYSTEMS

5. **Duplicate Select Program**
   Some fields or controls inherit Select Program from Model, or include a declaration of
   Select Program and a zoom handler with a call to the same Select Program.
   Xpa behaves differently from previous versions and will call this program twice.
   You will need to find every instance of Select Program in your application and delete
   one of the calls.

6. **Transactions and Locks**
   If your application is working with SQL databases, you will find out very quickly that the
   gateway in Xpa behaves differently than the gateway in v9.4. This could lead to
   numerous unforeseen errors, each of which will have to be found and addressed
   individually.

7. **Tab Order& Control Names**
   MSE's free tool implements the Tab Order with some errors, and in some cases it
   implements the wrong Tab Order, thus changing the behavior of your application. Also,
   Xpa works with handlers based on a Control's name. Most developers prefer that
   the Control's name be meaningful, so that they are easier and more intuitive to work
   with. The free tool implements auto-generated Control names that are not meaningful.
   This can be a significant disadvantage if an application has dozens or even hundreds of
   Controls.

8. **Tables**
   MSE free tool does not handle table columns, and replaces the old v9 compatible table
   with the one in Xpa. Without addressing all the potential ramifications of this issue,
   a future conversion to XPA will not work and worse, will delete some of the controls on
   that table.

9. **Preparing for RIA**
   The free tool from MSE does nothing to prepare your application for a future migration
   to RIA. In fact, it incorporates features which will not be supported in RIA and will need
   to be removed if you decide to convert your application to RIA/Mobile. This could result
   in a significant downstream expense.

10. Enhanced Features of Xpa

Xpa includes many enhanced features; however, none of them are implemented through the free conversion tool.

Your converted application will lack basic features like:

- A standard Windows Style appearance
- Built-in PDF printer
- Enhanced GUI properties
- Windows table control
- Unicode support
- Raise Event operation instead of KBput
- Gradient colors
- User State Identifier
- And many more…

A Better Solution

Go Up Technology applied the lessons learned from its years of performing successful Magic application conversions to Xpa, and encapsulated them in its own proprietary conversion solution (United States Patent # 6,698,014). The Go Up Technology solution can perform conversions starting with any version of Magic, not just v9.4, while ensuring converted applications are standardized and implement all the advanced capabilities of Xpa. This approach minimizes the costs and timeframes of the conversion project. Specifically, here is how it handles all the aforementioned issues.

1. **Windows Objects**
   Go Up Technology's proprietary solution takes an uncompromising approach to Windows Objects. During the conversion process, it allows selection of the new objects in the target version and implements the new capabilities of these embedded objects. The optimization process embedded in the solution ensures that after conversion, your application will no longer have any old version Compatible Object.

2. **RM (Record Main)-Converter**
   The proprietary solution from Go Up Technology was built to understand and handle complex and other problematic cases when converting Record-Main operations to Handlers. It also undergoes constant tweaking and improvement to ensure that the conversion of Record-Main operations to Handlers occurs as smoothly as possible, thus minimizing errors or problems.

3. **Link Validate**
   The Link Validate issue is handled as part of an overall solution that also deals with other links and their associated problems. Go Up Technology's proprietary solution will implement an error message only if it didn't exist, and will also fix the duplicate use of the same success code found in other links.

4. **Child Windows and Sub-Forms**
   Our philosophy is that when a new version of Xpa is announced, its new capabilities should be taken advantage of. We do not want our customers to depend on the "gig" solutions that generate lots of billable hours for their contractor, but leave the customer with features   that will no longer be supported in the future. This is the reason we implement Sub-Form controls instead of Child-Windows.

5. **Duplicate Select Program**

    The optimization process in Go Up Technology's proprietary solution knows how to recognize cases of duplicate Select Programs even due to inheritance or direct definition. This results in a converted application free from duplicate calls to Select Programs.

6. **Transactions and Locks**

    During the conversion process, the proprietary solution from Go Up Technology identifies declarations that can be problematic for performance or transactions. These declarations are located by the conversion engine and handled during the conversion, ensuring that after the conversion the application will work better and faster than before.

7. **Tab Order & Control Names**

    Go Up Technology's solution to the tab order issue is part of a solution for handling the connection between controls. During the conversion process we identify the unlinked controls and link them to their parents. The Tab Order is thus set according to the original order in record main and to the parent they linked to. In Xpa, the name of the Controls is a key issue since the handlers that are generated are based on those names. The application's ease of maintenance in the future depends on these names. Our solution ensures that the Control's name will be meaningful and intuitive, aiding future maintenance. This solution also includes an option to deal with the handler's names if a customer wants to deploy their application on computers that do not support the language your handlers are named in.

8. **Tables**

    During the conversion Go Up Technology's solution implements standard Windows table & column models and set all their new capabilities. This process ensures that tables include columns with titles, while removing the old labels' titles.

9. **Preparing for RIA**

    In our experience, many organizations who convert their applications to Xpa eventually convert those same applications to RIA. With that in mind, it is usually prudent to refrain from implementing features during the conversion that will cause problems in any future migration to RIA. The solution from Go up Technology therefore looks over the horizon beyond this conversion to Xpa, and lays a smooth foundation for a possible downstream conversion to RIA as well.

10. **Enhanced Features of Xpa**

    All the new and enhanced features of Xpa are implemented during the conversion, so that the converted application will not only be better than the old one but will also take advantage of new capabilities and improved GUI designs. Go Up Technology's conversion engines are flexible and can incorporate any type of solution required by the customer.

Conclusion

In their "Top Reasons to Upgrade to Xpa" brochure, MSE outlines a number of enhancements to their software that make upgrading a compelling proposition. Not mentioned are the challenges present in taking that upgrade path. Organizations considering a move to Xpa would be prudent to spend at least a little bit of time researching all available conversion options before deciding which is best. A successful conversion should be judged not only by its ability to unleash the full benefits of Xpa today, but also the potential rewards of RIA tomorrow.