# O'Brien/Reynolds e3000 Migration Framework (Ver: 1.4.98)

## Overview:

The O'Brien/Reynolds e3000 Migration Framework is a set of guidelines and recommendations for e3000 shops that need to migrate existing applications to a different platform. It is primarily oriented towards e3000 shops that have internally developed application logic that needs to be migrated in some fashion to a non-e3000/MPE system.

The Framework is a series of steps that should generally be followed in the appropriate sequence. However, many of the steps may occur concurrently. As with all Frameworks, the O'Brien/Reynolds e3000 Migration Framework is just a starting point that may work for a majority of e3000 shops that are considering migration. Many shops will need to modify and adjust the Framework as necessary to suit their particular environment.

The steps used in the O'Brien/Reynolds Framework are first presented as an outline of steps and any recommendations appear in each step. A detailed discussion of each step follows the outline.

## Feedback:

The authors have developed the preliminary framework based on their many years of e3000/MPEiX experience as well as their years of experience in Unix and Windows development using TurboImage and the major relational database management systems. Feedback from various sources will be included in future revisions. Please contact the authors below with any feedback or if you would like to receive future versions.

## Contact the authors at:

Terry O'Brien
President
DISC Migration Services Inc.
(303) 909-7992
tobrien@disc.com

Bill Reynolds
Director, Integration Development
Dynamic Information Systems Corporation
(303) 444-4000
breynolds@disc.com

# O'Brien/Reynolds e3000 Migration Framework Outline

## Step 1: Determine the overall migration approach.

1) Hope HP reverses its position to discontinue the e3000 or extends the timeframe.
2) Hope that an MPE/TurboImage binary emulator or open source environment is created.
3) Stay on e3000 after HP discontinues support using 3$^{rd}$ party maintenance companies or spare parts.
4) Move existing 3$^{rd}$ party application to same application on new system.
5) Convert to 3$^{rd}$ party applications.
6) Rewrite the applications.
7) Migrate the applications using migration tools (mapping APIs, MPE command emulators, source conversion tools, etc.).

Recommendation: Use a combination of options 4 through 7.

## Step 2: Determine the overall time frame for the migration effort and key target dates.

Sample Dates:

| | |
|---|---|
| 1$^{st}$ Qtr 2002 | - Decide on general direction and key vendors. |
| 2$^{nd}$ Qtr 2002 | – Inventory existing applications. |
| 3rd Qtr 2002 | - Do initial test conversions and development. |
| 4$^{th}$ Qtr 2002 | - Establish new time line and dates for development completion. |
| Year 2003 | – Perform rewrite and conversion work. |
| 4$^{th}$ Qtr – 2003 | – Size production server, establish budget and order equipment. |
| Year 2004 | – Install production server and perform concurrency testing. |
| Year 2005 | – Switch processing to new servers(s). |
| Year 2006 | – Use as backup year for re-development if conversion fails. |

Recommendation: Layout a tentative time frame and then adjust as needed. Wait as long as possible before sizing and ordering the production servers.

**Step 3: Take an assessment of the existing e3000 environment.**

- Number of CPUs
- Number of users/computer connections
- Number of non PC Terminals (especially HP specific terminals)
- Number of databases
- Amount of disk storage
- Critical applications
- Non-critical applications
- Purchased applications
- 3rd Party vendors for all applications and tools
- Types of languages (Cobol, Transact, Basic, Powerhouse, Speedware, etc.)
- Current user interface (green screen, client/server, web based)
- Screen tools used: V/Plus, CP3000, etc.
- Current development tools (editors, source control, etc.)
- Current operational tools (spoolers, job control, etc.)
- Deficiencies in current applications that may be considered for enhancements
- Number of technical staff – operations, development.
- Expertise of technical staff – trained on non-e3000 technologies

Recommendation: Try to determine the overall assessment of the existing e3000 environment as quickly as possible to determine overall scope and to aid in timeframe adjustment and migration planning.

**Step 4: Break down existing applications and processes into the following categories.**

1. Applications that will move to the same vendor on new hardware.
2. Applications to be replaced by new 3rd party applications.
3. Applications or processes to be rewritten.
4. Applications or processes to be run using mapping/emulation tools.
5. Applications or processes that will be discontinued.

## Step 5: Work with existing 3<sup>rd</sup> party application vendors and tools vendors on porting options for existing application.

The following vendors are known to have ports to other platforms of their existing HP3000 application solutions:
Sungard Bi-tech (SRN)
Ecometry (Smith Gardner)
eXegeSys (MM, PM)
Mitchell Humphrey FMSII

The following vendors are known to be working on ports of their applications:
QSS
Carter-Pertaine

The following development tools vendors are known to have ports of their tools:
Cognos -  Powerhouse
DISC - Omnidex
Robelle – Qedit, Suprtool
Speedware - Speedware 4<sup>th</sup> GL

## Step 6: Determine initial 3<sup>rd</sup> party application vendor options.
A search for packages on Google showed the following:
Over 100 Manufacturing vendors
Over 30 CRM vendors
Over 100 Financial/Accounting vendors

## Step 7: Determine vendor or vendors for mapping/porting.
See the HP Web site: www.hp.com/products1/mpeixservers/future/index.html

## Step 8: Determine the application development environment.
1) Java Centric
2) Microsoft Centric (.NET allows multiple languages)
3) Specialized Language/4GL Centric
   a. Macromedia Cold Fusion (the leading choice in this category)
   b. Cognos Powerhouse
   c. Speedware
   d. Others
4) Open Source (PHP/Zoap, etc.)

### Step 9: Determine the Operating System for the Data Base Server
　　　1) Unix/Linux
　　　2) Windows 2000

Recommendation: Choose Unix/Linux if scalability is a current concern.

### Step 10: Determine the data base platform.

For the Windows operating system:
　　　1) DB2
　　　2) Oracle
　　　3) SqlServer
　　　4) Postgres (open source)
　　　5) MySQL (open source)

For Unix/Linux operating system:
　　　1) DB2
　　　2) Oracle
　　　3) Postgres (open source)
　　　4) MySQL (open source)
　　　5) HP Eloquence

Recommendation: Choose one of the three proprietary data bases (DB2, Oracle, SQLServer) as part of a long-term strategy.

### Step 11: Determine the preliminary hardware vendor.

For high-end Unix proprietary systems, choose Compaq, HP, IBM, Sun,
For Linux systems, choose Compaq, Dell, HP, or IBM
For Windows systems, choose Compaq, Dell, HP, or IBM

Recommendation:  Start initial vendor relationships but delay ordering production servers until well into the migration process.  Hold off locking into Compaq or HP until the merger drama is reduced.  Buy initial Windows or Linux development systems to begin a preliminary hardware relationship.

### Step 12: Set up initial development systems.

Set up low cost development systems.
Delay ordering production systems until initial ports are completed.
Delay purchasing of data base for production system.
Begin account relationship with primary hardware vendor.

# O'Brien/Reynolds e3000 Migration Framework Details

The following is more detailed information that expands on the initial outline and recommendations.

## Step 1: Determining the migration approach

The first step is to determine what migration approach or combination of migration approaches should be taken. The Framework has identified six possible approaches as follows:

1. Hope HP reverses its position to discontinue the e3000 or extends the timeframe.
2. Hope that an MPE/TurboImage binary emulator or open source environment is created.
3. Stay on e3000 after HP discontinues support using 3$^{rd}$ party maintenance companies or spare parts.
4. Move existing 3$^{rd}$ party application to same application on new system.
5. Convert to 3$^{rd}$ party applications.
6. Rewrite the applications.
7. Migrate the applications using migration tools (mapping APIs, MPE command emulators, source conversion tools, etc.).

The O'Brien/Reynolds Framework believes that option 1 and 2, the reversal of HP's decision and the availability of a binary MPE/TurboImage emulator is extremely high risk and both options are highly unlikely and should not be considered. Option 3, stay on an e3000 and use 3$^{rd}$ party maintenance is only viable in a short term of one to two years after HP's termination of support and only for shops needing additional time to move to other systems or shops that have a fixed short term life of their application.

The Framework thus assumes that the majority of e3000 shops will opt for a combination of options 4 through 7. Additionally, if shops are opting to purchase 3$^{rd}$ party applications, they may want to select a package first and then work with the 3$^{rd}$ party vendor to determine the OS and data base platform. As such the Framework is focused on the e3000 shops that will be primarily rewriting or emulating/mapping major applications with some 3$^{rd}$ party application purchase as an adjunct to the conversion process.

## Step 2: Determine the overall time frame for the migration effort and key target dates.

Determining a timeline for the migration work is an initial attempt to try to map the time constraints and resources against the required work. For most shops, the following timeline can be used and modified as needed:

| | |
|---|---|
| 1st Qtr 2002 | - Decide on general direction and key vendors. |
| 2nd Qtr 2002 | – Inventory existing applications. |
| 3rd Qtr 2002 | - Do initial test conversions and development. |
| 4th Qtr 2002 | - Establish new time line and dates for development completion. |
| Year 2003 | – Perform rewrite and conversion work. |
| 4th Qtr – 2003 | – Size production server, establish budget and order equipment. |
| Year 2004 | – Install production server and perform concurrency testing. |
| Year 2005 | – Switch processing to new servers(s). |
| Year 2006 | – Use as backup year for re-development if conversion fails. |

Some shops may be able to accelerate the timeline and some shops may need the full five years or even longer. Shops will also want to plan around their business cycles and other operating and budgetary constraints. But an initial timeline should be established with the understanding that modifications will likely be required after the conversion work has been fully started.

## Step 3: Take an assessment of your current environment.

Taking an assessment of the current environment will provide the necessary information for future planning. For some shops, this will be relatively easy and for some shops, this will take some detailed analysis of the current environment. Although some shops may require more detailed information, the minimum assessment should contain the following information:

- Number of CPUs
- Number of users/computer connections
- Number of non PC Terminals (especially HP specific terminals)
- Number of databases
- Amount of disk storage
- Critical applications
- Non-critical applications
- Purchased applications
- 3rd Party vendors for all applications and tools
- Types of languages (Cobol, Transact, Basic, Powerhouse, Speedware, etc.)
- Current user interface (green screen, client/server, web based)
- Screen tools used: V/Plus, CP3000, etc.

- Current development tools (editors, source control, etc.)
- Current operational tools (spoolers, job control, etc.)
- Deficiencies in current approach that may be considered for enhancements

Critical success points that need to be preserved in the migration effort (response times, functionality, etc.)

## Step 4: Break down existing applications into the four migration options.

6. Applications that will move to same vendor on new hardware
7. Applications to be replaced by new 3$^{rd}$ party applications
8. Applications to be rewritten
9. Applications to be run using mapping/emulation tools

This is a first attempt to try to categorize the existing applications and make an early decision on what to do with each major application. During the overall conversion process, this decision may change but an initial must be made as it impacts the overall direction.

## Step 5: Work with existing 3$^{rd}$ party vendor

Review feasibility of moving the existing application.
Determine current status of vendor's port and how they are doing it (full rewrite, mapping/emulation tools)
Determine data base and operating system restrictions that the application runs on.

## Step 6: Determine initial 3$^{rd}$ party vendor options.

## Step 7: Determine vendor or vendors for mapping/porting.

## Step 8: Determine how the rewritten applications will look and decide the future development platform.

After taking an assessment of the current environment, an initial attempt of how the migrated applications will look to the end users needs to be performed. Also a look at the future user interface and some consideration for the development platform must be made.

Some shops may want to preserve green screen heads down data entry applications while others will want to move to a client/server or web interface either immediately as part of the migration or after the initial migration.

The Framework recommends preserving some green screen based user interface as needed for high volume heads-down data entry. The Framework also recommends preserving any existing client/server applications if possible with plans to develop new

applications using a web interface.  For new development using a web centric environment, a decision on the four current web development approaches should be made:

      5) Java Centric
      6) Microsoft Centric (.NET allows multiple languages)
      7) Vendor Centric
          a. Macromedia Cold Fusion (the leading choice in this category)
          b. Cognos Powerhouse
          c. Speedware
          d. Others
      8) Open Source (PHP/Zoap, etc.)

These four approaches are generating the most significant investment and all four approaches can maintain state. A CGI approach using Perl, C, or other languages is not viable for web based applications designed to replace existing client/server or green screen applications and should not be considered.

## Step 9: Determine the Operating System for the Data Base Server

The next step in the O'Brien/Reynolds Framework is to determine what operating system will be used for the data base server.  Note that in a multi-tier environment, the Framework does not need to determine what OS will be used for Application Servers and Web Servers.  Those can be determined at a later time and do not need to be the same as the data base server.

Today's computer environment shows only two mainstream operating systems for the majority of data base servers.  Those are the Microsoft Windows operating system and the propriety Unix implementations and open source Linux.  Note that the Framework considers Unix and Linux a single operating system type and refers to it as Unix/Linux further in the Framework.

The decision of the operating system may be influenced by the decision of the development tools and environments.  However, most mainstream tools work on both Unix/Linux and Windows or can access the data base servers using ODBC or JDBC in a multi-tiered environment.

Unix/Linux is more scalable today than Windows with Unix available on large multi-user Unix systems (HP Superdome, Compaq Wildfire etc.) and Linux now available on mainframe class systems from IBM and large servers from IBM and soon by the other major Unix vendors.

Shops with existing Windows expertise and those that prefer the Microsoft-centric development tools such as Visual Basic or plan on using Microsoft's .NET environment may want to consider Windows for the data base server.  However, note that a Windows

development environment with Windows applications servers talking to Unix/Linux data base servers for scalability provides the benefits of both operating systems.

Either operating system will be a viable choice for the overwhelming majority of the e3000 shops. The major criteria for deciding on the operating system should be the current expertise of in-house staff. If the shop does not have scalability concerns then Windows will be the likely choice as most shops have at least some Windows familiarity. If existing staff already has Unix/Linux expertise or there are scalability concerns, then Unix/Linux will be the preferred choice.

Shops may also want to determine the OS planned for application and web servers. Windows and Unix/Linux are both viable options for these types of servers and both can be set up and used effectively. As such, cost, internal expertise of staff, and simple bias towards one of the other OS may be the deciding factors in picking the OS for web and application servers. Note that the data base server is the most important choice and that shops may elect to use the same OS for the web and application servers for consistency.

## Step 10: Determine the data base platform

The database choices have dwindled in recent years to three vendor-supported databases and one open source database. For shops desiring maximum flexibility and portability between Unix/Linux and Windows then DB2 and Oracle are the only two viable choices. Both databases will easily meet the needs of existing TurboImage applications.

For shows that plan on staying completely with Microsoft Windows, then Microsoft SqlServer is a viable option.

For shops wishing to go open source, then PostgreSQL is generally considered more robust than other open source databases such as MySQL. Unless an RDBMS purchase is prohibitive to a shop, a vendor-supported database would be preferred over an open source database.

So for shops working under Windows, the following databases should be considered:
- DB2
- Oracle
- SqlServer

And for shops using Unix/Linux and additionally may also be using Windows, the following databases should be considered:
- DB2
- Oracle

Either DB2 or Oracle will meet the overwhelming majority of e3000 shops needs. For shops considering IBM hardware, IBM DB2 may make a good "all IBM" choice. Oracle is a good choice due to its large installed base and wealth of tools and products surrounding it.

Some shops may consider HP Eloquence because it has a similar look and feel to TurboImage. Although this may be a viable option for some shops, most new development, research and training is in the relational database space. Because HP Eloquence is not a mainstream data base and does not have the same level of investment or market share enjoyed by the other RDBMS vendors, this is not a viable long term option for most shops that are concerned about investment protection and using future new development tools.

## Step 11: Determine the preliminary hardware vendor

The Framework recommends staying with mainstream hardware vendors. Although a hardware vendor will be chosen, the sizing and ordering of production servers should not be attempted at this time. The Framework specifically recommends that the sizing of production servers is handled after initial porting and benchmarking has been performed as there is no reasonable way to predict performance between hardware for applications running under different operating systems and using different data base systems. And after the initial porting and benchmarking of the migrated applications, the primary server hardware vendor can still be changed before ordering the production server or servers.

For those shops that are satisfied with their current HP relationship, they may want to continue with HP. Further, HP has announced trade in credits for existing e3000s. However, for shops that are no longer enamored with HP, then IBM, Dell, and Sun would rank higher on the selection list than HP or Compaq due to the current HP/Compaq merger and the problems associated with the merger. Due to the current uncertainty surrounding both HP and Compaq, other vendors should be strongly considered.

But all the vendors on this list have proven track records, have the financial strength, and the company organization to be around for the future.

For Unix proprietary systems, choose one of the following:
    IBM, Sun, HP, Compaq

For Linux systems, choose one of the following:
    IBM, HP, Compaq, Dell

For Windows systems, choose one of the following:
    IBM, HP, Compaq, Dell

There are other companies such as Honeywell/Bull that some shops may want to consider for specific reasons. But for shops without any direct tie to other vendors, choose from one of the hardware vendors previously mentioned.

## Step 12: Order and set up development systems.

Before a production grade server can be properly sized, a critical portion of the re-written or emulated/mapped applications has to be moved to a development server of the target platform. Then a multi-user benchmark can be performed and initial performance characteristics can be determined.

Many shops may attempt to size a production server at this time but this is unnecessary and generally a waste of money as servers continue to drop in price in relationship to their performance on an annual basis. Further, the purchase of the production server should wait until adequate benchmarks can be performed from the converted applications. The Framework also recommends that production servers not be sized and ordered until six months before the actual application is cut over. This means that most shops will not be ordering production servers for another one to two years.

Shops should set up a small proprietary Unix/Linux development system or even an Intel based Windows or Linux server with adequate drive space to host at least 40% of the TurboImage data. For most e3000 shops, a multi-user development system should cost below $20,000 and many shops will be able to use Intel based PCs running Windows or Linux for development at an even lower price.

The authors specifically recommend that developers have a copy of the database and application on their individual workstations. The used market, specifically eBay, is an excellent place to determine current prices of development servers as well as checking with hardware resellers. Additionally, some consideration should be given to ordering a development system from the chosen hardware vendor so that an initial relationship with the new vendor can be established.