Applications Services FAQs – Legacy Modernization

1. ***Can I modernize my mainframe application environment without moving to a server platform?***

   There are several very good alternatives to moving to a new platform with regards to software and platform solutions. We recommend an analysis of your environment using a holistic approach. A vendor providing both managed hosting services and applications modernization experience can provide the most beneficial options; since every client has a unique environment, the one-way approach may only be good for the vendor who only offers one type of solution. Utilizing a managed hosted model for your mainframe applications can significantly reduce costs and eliminate capital expenses. Flexibility also allows the client to take advantage of the best options within each solution.

   An approach to modernizing in place on the mainframe to "extend" the use of your mainframe and applications is by adding technology wrappers around the applications which would not affect core business logic. This approach is an increasingly popular and low risk solution which produces high business value and payback, as business rules and core system processes remain intact and untouched. New technologies can then be introduced, including Web Portals and Services where existing applications become extended and exposed to new external technologies and architectures.

   Regarding mainframe software solutions, Blue Hill can convert your legacy code to newer technologies and your legacy databases to newer RDBMS databases. As an example, JAVA can run on mainframes, and a Linux operating system opens the door to web applications. These techniques also open the door to a .NET environment allowing you to extend your environment into new areas, such as internet and web services usage, and SQL databases.

   We recommend an assessment of your environment in developing objectives as the first step when considering modernization of any type. The options and approaches can be overwhelming; and an investment in the expertise required is a wise investment.

2. ***What are the latest trends for modernization and how do I know what is right for my company?***

   After all of these years and the advent of new and improved software products, there is still no right or wrong approach to legacy modernization. Companies need to account for their own strategic goals as well as their technology goals when deciding to modernize their applications and environments. The business needs must be weighed along with the costs and risks associated with the various approaches. As technology advances and demand for mobile and social applications grow, companies need to look at how they can best open their environments and provide these capabilities while still reducing costs and integrating new technologies with their legacy systems.

   As a result of these powerful demands on companies and their capabilities, factors such as the cloud, mobility, social networks, and BIG DATA have forced companies to look at more hybrid approaches to modernizing their legacy applications. Popular approaches now include an initial process called Application Rationalization as a first step in assessing the company's best alternatives for modernizing and reaching their goals.

3. *What are the options for modernization or transformation of my mainframe applications?*

Legacy Modernization and Legacy Transformation are used interchangeably however the underlying definitions are essentially the same. Popular definitions tell us that legacy modernization refers to the **conversion or rewriting** of legacy systems to a modern programming language whereas legacy transformation aims to retain and **extend the value** of the legacy investment through **migration to new platforms** and **porting** of software code. Whether it's modernization or transformation, software and platform environments are impacted.

Legacy modernization options fall within one of four categories and many times encompass a combination of these options in varying degrees. The Blue Hill **assessment process** helps you determine the best solution for your company's strategic goals:

Converting	Using automated parsers and code converters to convert 3GL and 4GL languages and databases for a quicker and more effective process
Re-writing	Rebuilding legacy applications in a new technology or platform, with enhanced functionality usually by adopting Service Oriented Architecture (SOA)
Re-hosting	Running the legacy applications, with minimal changes required, on a different platform. There are two techniques which could be utilized, Porting or Emulation
Replace	Replacement of legacy applications, in whole or part, with off-the-shelf software such as an ERP package

4. *What are the differences in terms of costs and effort between Emulation, Porting, and Re-writing applications?*

Labor costs vs. software costs are the distinguishing factors in each method. Additional metrics to consider are the risk and technological advancement levels.

- Emulation methods require the purchase of umbrella software that sits on top of your application software and emulates a mainframe environment on a server platform. Emulation solutions require little to no changes to the code. In comparison to porting and re-writing, emulation software costs are high and labor costs and efforts are diminished. Opting for this method is purely for financial reasons which are to get off the mainframe as quickly as possible and reduce or eliminate your hardware and equipment costs. There are no technological improvements to your application software and your environment is not extensible to .NET or JAVA frameworks. The distinguishing factor is that the overall risk level with emulations is low. This approach is commonly referred to as "lift and shift".

- Porting methods allow you to bring down (or port) your mainframe software code to run on a server platform in their "native" modes (See the FAQ on running in native mode). Some languages such as COBOL become portable by using software such as MicroFocus COBOL and GT Software's NetCOBOL. Minimal proprietary changes to your code are required and the re-compiled executables run on server platforms under their specific run-time engines. Other software languages may not be portable because there is no software product available to run those languages. DYL-280 and QuikJob are examples of 3rd party software which cannot be ported and would need to be rewritten or converted to a language which could be maintained such as COBOL. JCL cannot be ported and must be rewritten for the target platform or converted to scripts using code converters. The software required for porting solutions is not as costly as the "emulators" but the labor costs and efforts are higher. The

risks, considered medium level, are also higher than with an emulator and there is some gain in using newer technologies. With porting solutions, your environment can be extensible and opened to .NET and JAVA frameworks.

- Re-writing methods require the least in software costs but the most in labor costs and effort as all code is either converted or rewritten in newer technologies. A significant reverse engineering effort is required to extract business rules. CICS applications can be rewritten as web applications. Legacy code can be rewritten in JAVA or .NET and file structures such as VSAM or older mainframe DBs can be converted to databases such as Oracle and SQL. The risks are the highest with rewriting as opposed to the other methods and the technological gains are vastly superior. Complete re-writing approaches are more susceptible to failure.

	Reengineer	Software Cost	Code Converter	Effor	Time	Risk	Technology Gain
Emulation	Low	High	No Code Conversion	Low	Fast	Low	No Technology Gain
Porting	Medium	Medium	Code Generator	Medium	Medium	Medium	50% Technology Gain
Rewrite	High	Low*	OOP	High	Lengthy	High	Latest Technology 3G/4G
Replace	High	High	No Code Conversion	High	Medium	Medium	ERP Technology

5. *What are the objectives and options available with language conversion tools?*

There is no magic bullet to automatically convert legacy applications in total, to applications that use the latest technology and platform. Partly because there are usually a variety of 3rd party products employed along with the more standard languages in most mainframe environments, and partly because the object-oriented designs of newly developed systems cannot be replicated by converters. However, migrating away from COBOL may be more important and take precedence over the need to have a true OO design. For the purposes of this FAQ, the fundamental hypothesis in discussing converters is to convert legacy mainframe code to C# or Java when moving to either a .NET framework or JAVA platform. There are converters available such as COBOL to JAVA, which can convert millions of lines of code in relatively short periods of time; a process which will save an enormous amount of time.

In a modernization effort, other requirements may arise besides the ultimate conversion to JAVA or C#. For instance, when considering a “porting” option, the solution for some 3rd party software products may be to first convert the code to COBOL then port the COBOL to a server platform. In other scenarios, multiple source languages may exist, and multiple target languages may be desired or necessary in which case, converters may or may not exist for your desired transformation. With many converters on the market and various “from” and “to” languages to consider, an assessment is recommended to perform the required research to determine the best solution to suit your objectives. Being vendor and software agnostic allows for us to be completely objective in formulating optimum solutions.

6. *Are code converters viable options to consider when the decision is to re-write legacy applications?*

The decision is dependent upon your goals with rewriting. If one of your main goals includes having applications designed and developed properly using newer technologies, then code converters need to be carefully weighed. To ensure the best design and development techniques are incorporated, a complete rewrite may be required, and a significant reverse engineering effort is necessary. This approach applies even more when introducing new functionality and interfaces through more advanced design and SOA techniques. Another example again is converting COBOL to JAVA. With code converters the old designs remain as the underlying fundamentals although significantly newer technology can be introduced.

Should you not be too concerned with the "design" and true SOA, then code converters are a good way to jump start and progress effectively on a rewrite effort. As a rule, it is our opinion that code converters can be thought of as being on average 75% effective, where 25% cleanup is required. In many instances, code converters are not a perfect science. A conversion usually entails a combination of rewriting, porting, and emulation; therefore, code converters are a viable option in many cases.

7. *What are my options for modernizing my CICS systems?*

There are several vendors that have umbrella software to run CICS applications on a server platform, such as MicroFocus and GT Software. The CICS programs are handled the same way as the batch programs. There will be a need for some tweaking and a need to re-compile. Aside from running CICS programs under umbrella software, another course is to reverse engineer the programs and rewrite the CICS applications as web applications or other interactive systems. Other options include implementing ERP packages.

8. *What code converters are available for 3rd party mainframe software products?*

There are many 3rd party software products running on mainframes and there are many vendors and numerous conversion tools available for languages. It is important to determine the products available for the source and target languages particular to your environment and objectives, and confirm the products exist. Many of these products were developed by conversion specialists for specific conversion projects at that time, and then made available to the open market. Some desired code converters may not be available, or an obscure code converter may exist, however the consensus is that in these situations, your software most likely will need to be rewritten. Two approaches can be taken: either rewrite to a convertible or portable code such as COBOL and then pursue your plans for COBOL code, or rewrite to a new native technology consistent with your target platform. Always remember the 75% / 25% rule with all converters.

As part of our assessment and because we are vendor and software agnostic, we verify vendors' claims and determine the most suitable converters available for your objectives with the most effective course of action. In some cases, several products may be required which may be provided by different vendors. Blue Hill develops the most effective solution considering all necessary software and approaches as opposed to a biased one-way approach.

9. *Are there effective code converters for Assembler?*

There are a handful of companies to consider providing Assembler code converters or advertise the ability to convert Assembler code with "proven" software. Generally, the objective would be to convert Assembler to either a "C" type language or JAVA for migrations to newer technology platforms, or to convert Assemble to COBOL for a "porting" solution where the COBOL platform will continue to be supported. Typically, the

conversion software would translate the Assembler code to a proprietary language which would then be translated to the appropriate target language.

While agreeing that assembler code converters can be an effective tool in the overall modernization process, you must completely understand and ensure that the software and capability claims meet your expectations. At times, it may be best for the Assembly code to be reverse engineered and rewritten to ensure that all the detail-oriented functionality and intended operations are captured replicated. For larger projects with numerous Assembler programs, code converters provide a means to effectively shorten the conversion process for this language. Deciding on an approach that best fits your goals should include considerations for the number of Assembly programs, the costs of the conversion software, and the effectiveness of the product. This could be a daunting process when it may be only one piece of your overall modernization effort.

Blue Hill recommends performing an assessment of your existing Assembler code and your overall modernization objectives to determine which process best suits your objectives and if the available converters will satisfy your requirements. Being vendor and software agnostic, Blue Hill will provide the most effective components and solutions to your overall modernization efforts.

10. *Can source code such as COBOL be ported to a client server platform and run in a native mode?*

Very few mainframe software languages can be **ported** to an open systems platform; and those languages that can be ported are required to be modified slightly and "recompiled" to run under proprietary run-time engines. COBOL is one of the more common examples of portable code. Although COBOL can be ported in this manner, COBOL cannot run in its mainframe native mode unless it is running under a mainframe emulation product. This is commonly referred to as the "lift and shift" approach. JCL like COBOL cannot be ported and run in its native mode unless an emulation product is employed. While little or no changes are required to COBOL and JCL when using emulation software, there are no technological gains and there remains a dependence on these software products.

11. *What are the options with mainframe code such as DLY-280 and QuikJob?*

For basically all 3rd party software products, there are only two options. Either re-write the code in a desired target language or convert the code using code converter software if it exists. Most 3rd party software languages are not portable unless the software company has developed a server version of that software language. There are converters for DYL-280 and QuikJob, however unlike the main languages such as COBOL, your target language options are quite limited. For the most part, the code conversion from DYL-280 and QuikJob would be limited to COBOL as the target language.

A conversion to COBOL is a viable and reasonable option for a "porting" or mainframe "emulated" solution and it will meet your objectives quite well. The newly converted COBOL programs can then be maintained nicely within your new open systems environment using products such as Microfocus COBOL or Net COBOL. Should your target environment be a .NET or JAVA platform, using a code conversion process for DYL-280 and QuikJob would be quite convoluted as there is no direct conversion route to "C" or JAVA **. In this case you would need to weigh the costs and risks for using multiple converters and conversion steps, against the costs for completely rewriting the software to the target language.

Just as with the Assembler process, this is a daunting effort, and Blue Hill recommends performing an assessment of your existing environment and your overall modernization objectives to determine the solution which best suits your modernization objectives.

As mentioned throughout, some of the statements we make are based on our experience and research. At any point in time, a new converter might become available on the market. Therefore, we recommend an assessment as a prerequisite for all modernization projects to determine the best solution for your objectives.

12. *What do we do with scheduling software such as CA-7 when applications are ported to a server platform?*

Job Scheduling has a long history and has been one of the major components of IT infrastructure since the early mainframe systems. The mainframe era Job Control Language (JCL) on IBM mainframes to handle dependencies, is typified by sophisticated scheduling solutions as part of the systems management and automation toolset on the mainframe. The open systems era which initially was limited to standard scheduling tools such as Cron now requires the use of more modern schedulers for a variety of architectures and operating systems. The need for mainframe types of standard job schedulers has grown with the increased adoption of distributed computing environments. More recently this category of software has increasingly been labeled as 'Workload Automation' for the next generation of job scheduling applications. An appropriate consideration is: how would we provide that same mainframe IT infrastructure and capability on an open systems platform?

If you are considering moving workloads to the Windows platform, there is a broad choice of job scheduling tools that can handle the task of workload management and automation. These tools provide the same features and functions as their mainframe counterparts. However, to plan this move, you should consider carefully whether you are planning to preserve your current IT batch operations and development practices unchanged, or whether your move requires a change to embrace the new opportunities of event-based automation. If the former, then consider adopting migration tools that preserve the operational characteristics of the mainframe (such as JCL emulation). If the latter, now may be a good time to consider adopting the advanced capabilities of a job scheduler to help you handle complex workload automation.

Job schedulers and workload automation is a topic in and of itself, and a hugely significant piece of the modernization transformation. The success or failure of your entire modernization effort can be affected by the solution implemented for your workload automation requirements. Blue Hill strongly recommends performing an assessment of your environment and objectives to determine the best solution for your workload automation requirements. This subject alone can be extensive and risky without devoting the required time and effort to select and implement the most suitable solution for your company.

13. *What are my options regarding VSAM files and how are VSAM files handled on a server platform?*

There are basically three options available when addressing your VSAM files and VSAM data which we will address; however, there may be a need for extraordinary interim steps based on your objectives and environmental situations.

- Download the VSAM data to a server-based file (usually .DAT) and access the open systems data file from your ported COBOL programs using the existing COBOL code. Products such as MicroFocus COBOL or NetCOBOL are required as run-time engines for your ported COBOL programs
- Utilize software products that provide a means for server-based programs to access VSAM files on the mainframe.
- Convert your VSAM files to an RDBMS such as SQL or Oracle. These conversions can be done “manually” or using software products.

When opting for a porting solution mainly regarding legacy COBOL applications, VSAM data can be downloaded to server based (.DAT) files and can be accessed as direct access files by the COBOL programs. The ported COBOL programs are required to run under specific run-time engine software such as Microfocus' Enterprise Server and GT Software's Net COBOL. Small modifications to the programs are required to enable direct access functionality and all programs need to be recompiled, however significant changes to the VSAM file access routines are not required. The trick (or complexity) is how to handle the data formatting which is an entire subject. Simplistically stated, data can remain in EBCIDIC format or be converted to ASCII and in the same manner, the COBOL programs need to be compiled as EBCIDIC or ASCII based modules. Be aware that your decision on how to handle the formatting issues of each individual program can have a major effect on the integrity of your applications and your overall porting effort. An assessment will allow you to have a complete understanding of the intricacies involved in a porting solution.

A second option is to leave your VSAM files on the mainframe, port or convert legacy programs, and utilize software to access the VSAM files from the open systems platform. Typically, this is an effective and efficient solution when your modernization objectives are to move to an ERP system or when a comprehensive rewrite or redevelopment effort is selected.

A third option is to convert your VSAM data to an RDBMS (database) structure. SQL and Oracle are the more common and popular examples of target database systems. DB2 is also a popular choice, however we recommend giving serious thought to this subject by carefully weighing the pros and cons of DB2 vs a SQL or Oracle approach to justify your objectives. Converting to an RDBMS is an arduous task in itself; converting to a fully optimized RDBMS can amount to an enormous effort and can possibly lead to a failure of your modernization initiative.

The appropriate solution is dependent upon your ultimate goals for your future state. However, there may be various paths and interim steps required to eventually get to your final future state and the process may require a road map designed specifically for your environment and objectives.

Blue Hill Data Services: Cost-Effective, Secure, On-Shore Data Center Hosting Solutions

Blue Hill Data Services consistently achieves 100% client satisfaction by providing Private Cloud and fully managed 24/7 data center hosting solutions, and a full array of complementary IT support services.

We deliver all services from On-Shore, USA, and since 1994 we have supported government and commercial clients, helping them reduce their operating expense, eliminate capital expense, mitigate the risk of a retiring workforce, meet all regulatory compliance and audit requirements, and achieve 100% client success.

Blue Hill specializes in all mainframes and provides multi-platform services and infrastructure solutions. We have successfully carved out a niche in supporting Mainframe mission-critical applications Acting as our Clients' partner, we can support their environments either indefinitely or until they choose to get off the Mainframe, providing flexibility in reducing costs as utilization decreases.

Our differentiation is providing customized solutions, flexibility both in contract and solutions, cost effectiveness, and personalized attention. Our client's data center environment can be hosted within Blue Hills' private cloud facilities, or Blue Hill support services can be provided remotely to the client's site.

IT Managed Delivery Services and Solutions:

- Mainframe-as-a-Service (MFaaS)
- AS/400 - iSeries-as-a-Service
- Server/Cloud Infrastructure Systems Support
- Applications Development and Maintenance Support Services
- Disaster Recovery and Business Continuity
- Colocation Services
- Remote Support Services
- Consulting Assistance

Our deep technical skills and longstanding experience enable us to support our customers' legacy environments as well as implement new technology solutions.

We do not force clients to upgrade or change the way they are used to doing business, which makes the transitions seamless, minimizing risk and completing migrations in shorter timeframes.

We are proud of our client satisfaction 100% - we have never lost a client due to poor service or high costs.

Contact us today to begin the process:

Nicholas Pidick
Executive Director, Sales and Business Development
Blue Hill Data Services
845.875.7082 | NPidick@BlueHillData.com