

Determining the DIBOL Conversion Effort

WHAT CBL IS AND IS NOT

CBL is a programmer's tool designed to be used by knowledgeable DIBOL programmers. A basic understanding of the C programming language is also necessary when using CBL. CBL *is not* a silver bullet that will convert your DIBOL applications to C and have you up and running on any platform supporting C in hours. CBL is a powerful conversion tool that makes converting DIBOL to C a much easier and more structured task, reducing the risk and effort inherent in any conversion project. Using CBL to convert DIBOL to C will save you a substantial amount of time and money.

USING CBL

CBL is used very much like a programming language compiler. Compilers like VAX DIBOL input DIBOL source code and use it to generate object code. CBL inputs DIBOL source code uses it to generate ANSI C code. As with a language compiler, the more skilled and experienced the programmer, the more benefits that can be derived from CBL.

Like the DIBOL compiler, CBL uses command qualifiers and compiler directives to control how the DIBOL code is converted and what sort of output code is generated. A good understanding of both DIBOL and C is helpful in getting the most out of CBL.

When using CBL, changes are made to the DIBOL source code to facilitate the conversion to C. The emphasis of CBL is on maintaining the DIBOL source code and allowing the CBL Translator to maintain the

C code. Once the generated C code is manually modified, the CBL Translator can no longer be used to maintain it. This methodology has the advantage of allowing DIBOL programmers with rudimentary C skills to quickly become proficient at producing sophisticated C code. The translation process also serves as a teaching aid in training DIBOL programmers the intricacies of C programming.

CONVERSION EFFORT

The amount of effort required to carry out a DIBOL to C conversion is dependent upon several factors. They are:

- **DIBOL programming skills.**
- **C programming skills.**
- **Level of code complexity.**
- **Level of code standardization.**
- **Size of application(s).**
- **Target environment.**
- **Target environment skills.**
- **Conversion time frame.**
- **User expectations.**

Obviously, if you are expecting to convert your DIBOL applications into full-blown windowing applications, complete with point and click Graphical User Interfaces (GUI), you have your work cut out for you, even with the assistance of CBL. Carrying out this type of conversion in a short period of time is difficult, and carrying it out while trying to learn C is almost impossible.

On the other hand, converting a DIBOL application to C to run under OpenVMS on the Alpha AXP platform is a fairly

straightforward process, provided the DIBOL source code is reasonably standardized and does not use exotic constructs or extensions. This type of conversion can be carried out quickly with minimal impact on the system and its users.

The bottom line is that a good understanding of your present DIBOL applications and the target environment, along with realistic expectations concerning time and resources, are crucial to the success of a conversion project. CBL is not a silver bullet that will automatically take care of all of your conversion issues. It is a powerful tool that will *assist* you in your conversion efforts. There is no substitute for adequate preparation and planning when conducting a conversion.

Should your conversion resources be limited, MSI offers a full suite of conversion assistance services. Contact us for further information.

OpenVMS Conversions

Converting DIBOL to C to run under OpenVMS on either the VAX or Alpha platform is the most straightforward and least difficult type of conversion. Essentially, the only application component being changed is the DIBOL source code. Command procedures, system services, the file system, and calls to non-DIBOL applications remain unchanged.

Personnel with reasonable DIBOL skills and very basic C skills can effectively employ CBL in an OpenVMS DIBOL conversion. The CBL for OpenVMS kit is designed to work with the VAX C and DEC C compilers. Using CBL is an excellent way to port DIBOL applications to the Alpha OpenVMS environment.

MSI can provide direct assistance and turnkey solutions for OpenVMS DIBOL conversions.

UNIX Conversions

Converting a DIBOL application to C to run under UNIX is more complex than converting the same application to run under OpenVMS. In addition to converting the DIBOL source code to C, you must replace the file system, command procedures, system services, and any calls to non-DIBOL applications.

A conversion to UNIX will entail comprehensive planning and application testing and should be conducted by personnel with well-developed DIBOL, UNIX, and C skills. CBL kits targeted at the SCO UNIX and HP-UX environments are available.

MSI only provides indirect support for UNIX conversions. We do not maintain Unix expertise on staff.

MS-DOS and Windows Conversions

A DIBOL conversion to C running under MS-DOS or Windows faces all of the issues a conversion to UNIX entails. In addition, memory constraints under these operating systems can hamper a conversion effort. Even using a large memory model with your C compiler, you may find that some converted DIBOL programs are simply too large to run under these operating systems. If converting a DIBOL application to run in the PC environment, we recommend that you consider WindowsNT as your target environment.

In addition to memory constraints, conversion to a windowing operating system often involves the desire to convert the interactive portions of the DIBOL application to a GUI interface. CBL will be unable to provide much assistance in this portion of a conversion, as the methods for programming a DIBOL interface and a GUI are very different. Batch DIBOL program conversion should be pretty straightforward

once the ISAM file system has been integrated, but interactive program conversion will require more manual effort.

Personnel with DIBOL experience and C experience under MS-DOS and Windows will be necessary to successfully convert a DIBOL application to run under MS-DOS or Windows. The CBL MS-DOS and Windows kits work with Microsoft C version 7.0 compiler and the Microsoft C version 8.0c compiler contained in the Microsoft Visual C++ 1.5 kit.

MSI only provides indirect support for MS-DOS and Windows conversions. We do not maintain MS-DOS and Windows expertise on staff.

WindowsNT Conversions

Like a UNIX conversion, a DIBOL to C conversion destined to run under WindowsNT will be more involved than a conversion under OpenVMS. However, converting to WindowsNT does offer the advantage of providing a target environment with similarities to OpenVMS while avoiding the memory limitations of MS-DOS and Windows.

While DIBOL batch program conversion should be pretty straightforward, converting DIBOL interactive programs to run under WindowsNT presents problems similar to those encountered when converting interactive applications to run in any GUI environment. There is no automated method for converting the interactive DIBOL code into GUI code. However, using WindowsNT Console Mode to support interactive DIBOL programs can simplify the solution to this problem.

Personnel with a good understanding of DIBOL and experience with C under WindowsNT will be necessary to successfully convert a DIBOL application to run under WindowsNT. The CBL for

WindowsNT kit works with the Microsoft Visual C++ compiler for WindowsNT.

MSI only provides indirect support for WindowsNT conversions. We do not maintain WindowsNT expertise on staff.

ISAM CONVERSION SUPPORT

If a DIBOL application is being moved from OpenVMS to another environment, then the issue of an ISAM file system must be addressed. ISAM file services are provided under OpenVMS by the RMS file system. Under other operating systems, ISAM services can be provided by a variety of products.

Since there are many methods to address ISAM file services, CBL cannot provide automated support for non-RMS ISAM file services. ISAM header files and generic function prototypes are provided with CBL to assist experienced C programmers in integrating their ISAM system with the C code generated by CBL. No further documentation or support for ISAM conversion is provided.

GUI CONVERSION SUPPORT

Even with the assistance of the CBL translator, conversion of a DIBOL application to a windowing environment is a job for an experienced GUI programmer. CBL does not provide automated support for conversion to any type of GUI environment. As with ISAM support, programmers are basically on their own when carrying out a GUI conversion. The CBL for Windows and WindowsNT kits provide GUI enabling source and header files. These files are intended as examples to be used by the programmers carrying out the conversion. No further documentation or support for GUI conversion is provided.

CONVERSION TO A DATABASE

When converting from a flat-file system to a database environment, be sure the database system has been adequately planned and designed before attempting the conversion. Trying to retrofit a flat-file application to a poorly designed database can be very costly and time consuming.

OpenVMS-based Database Conversions

If part of the goal in a OpenVMS-based DIBOL conversion is to move the applications into a relational database environment, then we recommend that this be undertaken in two steps. Step one would involve the conversion of the DIBOL code to C and the testing of the converted applications. The CBL Translator makes the DIBOL to C conversion a very automated process. Step two would involve the replacement of the RMS file system with calls to a relational database.

The CBL Translator can be used in the database conversion process to integrate database or SQL calls into the converted C code. The CBL Translator accomplishes this by permitting the user to embed SQL and relational database code in the DIBOL source code. The user flags the embedded code as passthru code, whereupon the CBL Translator passes the embedded code through to the generated C source file and includes it in the C application.

CBL does not automatically convert RMS calls to relational database or SQL calls.

Non-OpenVMS Database Conversions

If part of the goal in a non-OpenVMS DIBOL conversion is to integrate a relational database into the generated C code, then we recommend integrating the database during the conversion of DIBOL to

C. CBL's passthru capabilities can be taken advantage of to embed the calls to the database system in the DIBOL code before it is converted.

CONCLUSION

Converting an application from one programming language to another is always a challenging task. CBL helps reduce both the time spent converting an application and the risks involved in an application conversion.

Effective use of CBL requires that you understand the DIBOL applications being converted, the DIBOL programming language, and the target environment. A successful conversion effort requires careful planning and adequate resources, regardless of the conversion tools used.

Feel free to contact us for help in developing a better understanding of the conversion process and how MSI and CBL can help you reduce risk and effort in a conversion project, saving you time and money.