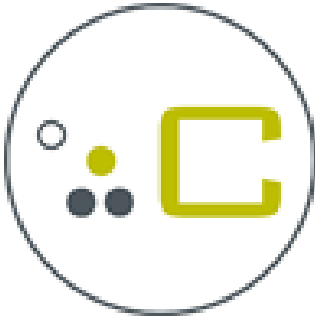# Anubex Assembler to Java/C# migration

About Assembler to Java/C# migration

The CodeTurn Assembler to Java/C# migration provides an automatic conversion of legacy Assembler to functionally equivalent C# or Java programs that run on Open Systems. The generated C#/Java code is of an unprecedented quality in the field of Assembler migration, and enables the migration of applications that were previously considered to be unfit for migration.

Introduction

For the longest time, Legacy Applications with a substantial portion of business logic written in Assembler (IBM z/OS, z/VSE or BS2000/OSD) were sentenced to eternal life on the legacy platform. The main points preventing their migration to Open Systems:

❖ The manual redevelopment of large amounts of Assembler code in C# or Java is not realistic (financially nor technically) due to the lack of proper functional documentation of the existing source code, and lack of skills to manually reverse engineer them;

❖ The quality of automated migration tools was too low, leaving too much manual analysis and development so that using said tools did not significantly reduce the cost and risk of the migration project.

With the development of the CodeTurn Assembler Migration tools finally a viable option is provided for this type of applications.

Supported Platforms

Currently the following technologies are supported:

Assembler Platform	Target Languages
z/OS	C#, Java
z/VSE	C#, Java
BS2000/OSD	C#, Java

How does it work?

Usually, CodeTurn operates on the application source code, as that is where the real business logic can be found. For the Assembler Migration this is slightly different as this starts exclusively from the *expanded Assembler compilation listings* – as can be generated with the assembler on the mainframe. This chosen approach avoids the need to emulate both the Assembler Directive and Macro (directive) behavior at runtime on the target platform, and allows an elegant conversion of the Assembler source code.

Any negative impact the expansion of the User Macro's might have on the readability of the migrated source code, is mitigated by the migration tool e.g. by the preservation of the Assembler DSECT names.

Key Features

Complete Solution

The migration solution covers each of the following elements of a mainframe Assembler applications:

- ❖ Assembly instructions
- ❖ Assembler instructions (directives)
- ❖ User defined macro definitions (inline)
- ❖ User defined macro invocations (inline & copybook)
- ❖ Invocations of system defined macros
- ❖ SVC calls

Generated Artefacts

Every mainframe assembly program is converted to a corresponding C# program or Java program.

Additionally, the conversion tool generates all the files that are required to maintain and build the converted code with standard tools in the chosen environment.

Output Parametrization

The generated C#/Java code can be parametrized on multiple levels to optimally blend in with customer standards in these technologies:

- ❖ Package names used in the source code;
- ❖ Identifier Casing in the source code;
- ❖ …

Comment Preservation

The original application comments (both the Full Line and Inline comments) are preserved in the target source code to enable future application maintenance.

Integration with other Technologies

Typically, the Assembler logic is only one part of the complete customer legacy code. For the most commonly used mainframe technologies (COBOL, Natural, IDMS, …) Anubex has developed migration solutions to C#/Java as well, and the migrated Assembler code seamlessly integrates with them.

Manual Migration?

Although the preferred approach is to work with the fully automatically converted code, in some particular cases a manual rewrite of a particular piece of Assembler code can be preferred. Such selective, limited scope rewrites can be performed and easily integrated in the automatically converted code as well.

References

Anubex has a 100% success rate, having completed numerous migration and modernization projects around the globe.

Each of these projects has been delivered on time, within budget and to the complete satisfaction of the customer.

Automated Testing

No migration project can succeed without the confidence that the migrated system is ready for production and this confidence can only be built by testing the migrated system extensively and repeatedly.

For this reason, Anubex has developed two products specifically designed to reduce the time and effort spent in testing:

- ❖ TestMatch is used to record-and-replay the behavior of on-line and transaction based applications. It can compare screens and messages produced by the original and the converted applications, revealing any differences in functional *and* non-functional behavior.

- ❖ DataMatch is a similar tool focused on data-centric and batch applications. It basically validates that the converted application (on the target OS) has the exact same effect on the relational database than the original application had on mainframe.

Both tools help in greatly reducing manual effort, by eliminating the need to write out test scenarios, and by enabling the autonomous replay of the recorded scenarios. They effectively limit the business-user involvement to just the acceptance testing.
More details about TestMatch and DataMatch can be found in the respective Product Sheets.

migrations@anubex.com
www.anubex.com

Version 1.2