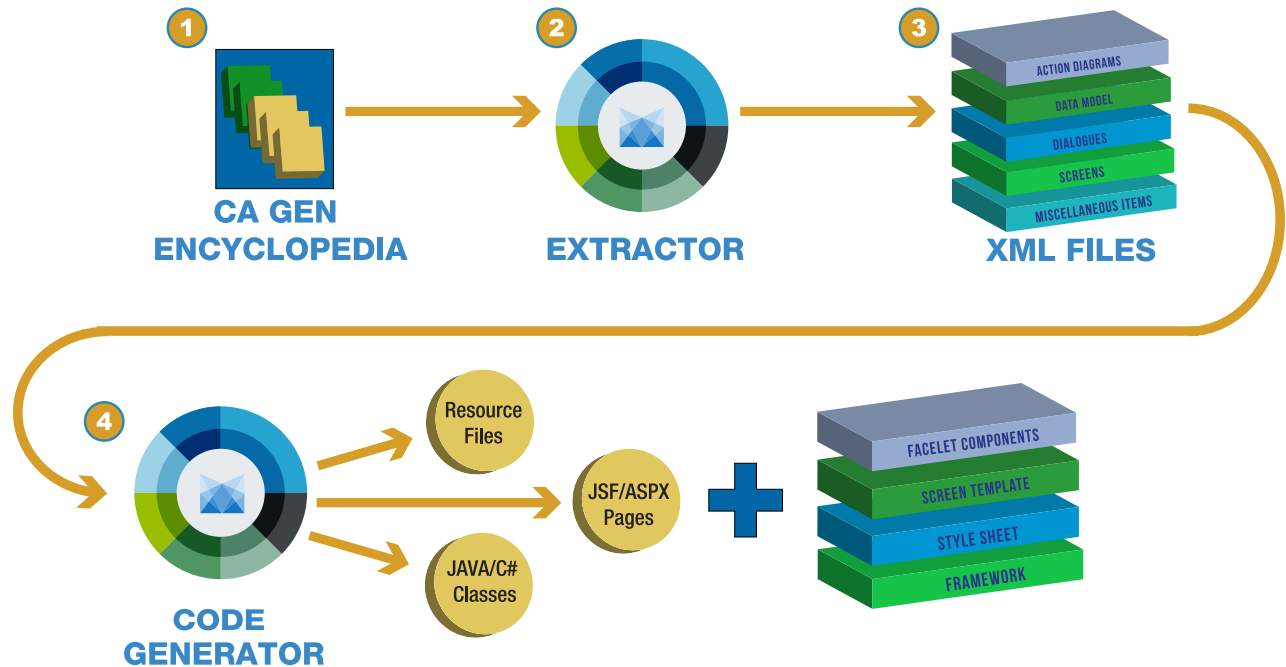


# AUTOMATED CONVERSION CA GEN

Modernize codebases and data tier without modifying behavior. Automated Conversion offers a 100% functionality match for model, view, and controller layers. This solution is great for moving CA Gen apps to Java or C#, and databases to SQL Server, DB2 or Oracle.

## HOW IT WORKS



- 1** We start by using CA GEN encyclopedia as input, not the CA GEN generated code. This ensures the converted code is compact, clean and easy to maintain. We offer flexibility around the deployment configuration- whatever it is, we can automatically convert it.
- 2** We then feed the collected input into the Extractor Engine to convert CA Gen flows into XML files to be used by the Modern Systems framework..
- 3** The inventory is fed into the final code generator to be processed into native Java or C#.
- 4** Finally, the Modern Systems Code Generator produces Java or C# source and binary code for converted CA Gen procedures and action blocks, XML files for converted CA Gen flows, source code for the Modern Systems framework, default configuration files, and documentation. Once tested to ensure like-for-like functionality, the modernized environment is ready for deployment and the legacy environment can be turned off.



CASE STUDIES



VIDEO OVERVIEW

WHITEPAPERS



MODERNSYSTEMS

MODERNIZE WITHOUT COMPROMISE.  
modernsystems.com

# AUTOMATED CONVERSION CA GEN

GETTING TO THE GOAL LINE

## Assessment

The Assessment is a complete research and analysis project that outlines all mainframe application and database refactoring candidates. Components are classified and listed in detail. Notes are attached to components requiring special attention during the refactoring process. All application components are inventoried, classified by language, and cross-referenced. Missing components are collected and added to the inventory. Duplicate components residing in multiple customer repositories are eliminated from the inventory. The assessment results in a complete understanding of the current processing environment.

This phase also includes discussing and reviewing the overall system test strategy and the division of the converted code into work packets.

Modern Systems will define a set of topics during the assessment phase that must be addressed prior to conversion. In order to speed the modernization process, these topics should be addressed by those team members who are best suited to understand the topic, the solution options, and any changes or activities that are required to address the Areas of Concentration.

Once the Areas of Concentration are identified, customer teams and Modern Systems will address the areas about which they are most knowledgeable and for which they are best suited to implement a solution. Any additional Areas of Concentration that are identified during the course of the Project will be addressed and assigned in the same manner, to the most appropriate team.

## Code Conversion

During this phase, the entire inventory of CA GEN components supplied and scoped during the assessment is automatically converted to functionally equivalent Java or C# code using the Modern Systems toolset. The toolset is configured to support the options selected during the assessment. At the end of this phase, the converted code, without compilation errors, along with the supporting framework code, is packaged and delivered to the Client for subsequent testing. The package includes:

- Java or C# source and binary code for converted CA GEN procedures and action blocks
- XML files for converted CA GEN flows, used by Modern Systems framework
- Optional: Source code and compiled DLL for generated COM proxy classes for converted CA GEN server procedures
- Optional: Source code and binaries for generated proxy classes for converted CA GEN server procedures
- Java or C# source and binary code for Modern Systems framework supporting execution of the code above, along with the documentation for it
- Default configuration files



# AUTOMATED CONVERSION CA GEN

GETTING TO THE GOAL LINE

## 3 Test, Refresh, Deploy

Modern Systems tests a subset of the converted code using a test plan with documented test scenarios (test scripts) provided by the customer. The customer will write the tests and run them on the existing system to capture and record the expected results. Modern Systems will then run the tests against the same data on the converted system, then identify, investigate, and fix discrepancies in the expected behavior of the modernized application. Pre-delivery testing will consist of test cases picked from all available functional test cases in order to be representative of different parts of the applications.

Once pre-delivery testing is complete and any discrepancies in application behavior are resolved, Modern Systems performs a code refresh to ensure that any changes that took place in the legacy application environment during the conversion process are accounted for, converted into the target language and environment, and tested.

The Modern Systems teams work closely with our customer teams to ensure a smooth and error-free transition night or weekend. A primary team may be onsite during the cut-over, and all other Modern Systems team members are on-call should assistance be required. Our teams also provide any required assistance during the 90-day warranty period following the deployment.

We also offer post-application support of the modernized application, available at a fixed rate or in hourly buckets purchased as needed by the customer.

