

# AUTOMATED CONVERSION NATURAL/ADABAS

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 Natural apps to Java , C#, or COBOL and ADABAS databases to SQL Server, DB2 or Oracle.

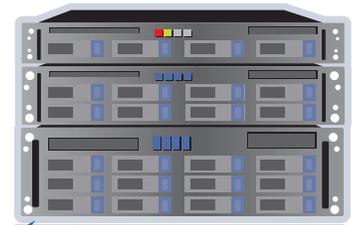
## HOW IT WORKS



SOURCE SYSTRANS



CONVERSION TOOL

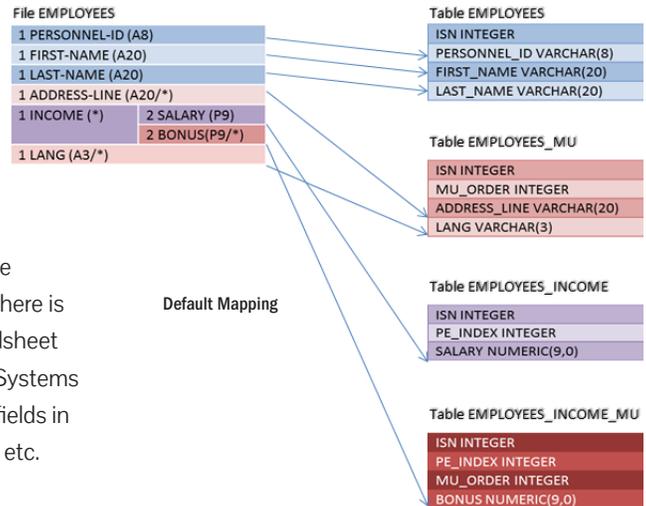


WEB APPLICATION

Java or C# Classes

Web (JSP or ASP) Pages

The Modern Systems conversion process takes Natural Source code and generates a fully functional Web Application. The process requires no “post modernization” manual fixes – the code generated is ready for deployment and maintenance.



Prior to the Natural code conversion, the ADABAS-to-Relational mapping must be determined. The Modern Systems tool has a Default Mapping that it uses when there is no other input (shown at right). This mapping can be changed by editing a spreadsheet (generated by the Modern Systems Modernization Tool) or by using the Modern Systems DDM Editor. Changes to the Default Model include denormalizing fields, placing fields in different tables, have different column names, use functions to build field values, etc.



CASE STUDIES



VIDEO OVERVIEW

WHITEPAPERS



# AUTOMATED CONVERSION NATURAL ADABAS

GETTING TO THE GOAL LINE

## Mainframe Assessment

The mainframe assessment is a complete research and analysis project that outlines all mainframe application and database conversion candidates. Components are classified and listed in detail. Notes are attached to components requiring special attention during the conversion 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 Mainframe 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 mainframe 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.

## Data Migration

The Modern Systems Tool uses the Canonical DDMs, plus the Data Model as adjusted by the customer, and generates a pair of programs for each. There is an Unload Program generated in Natural and a Load program, generated also in Natural but then converted to Java using Modern Systems' Modernization Tool. The Load and Unload programs are mirror images of each other – the Unload program reads the ADABAS file and writes the data to a variable length sequential file, and the Load program reads the sequential file and writes to the database. Only the C\* counts of MU and PE fields are written to the sequential file. The Unload programs are packaged as a single JCL stream that can be run with minor modifications such as setting JOB card and STEPLIB and altering the sizes of the output datasets.

The Load programs are packaged by our Migration Team into a stand-alone GUI utility that provides facilities for creating the target database, creating all tables, indexes, triggers and constraints, creating bulk load files from the raw data and loading this data into the database. Most operations can be done for all files or for a subset of files. The DDL for creating all Database Objects is also created and included by the Modernization Tool.

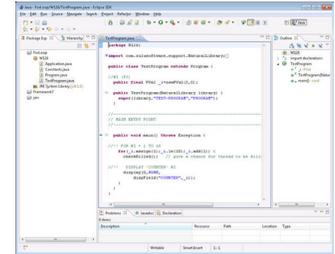
A new facility Unload Compare verifies the validity of the conversion, where it compares the mainframe and generated versions of the unloaded files.

# AUTOMATED CONVERSION NATURAL/ADABAS

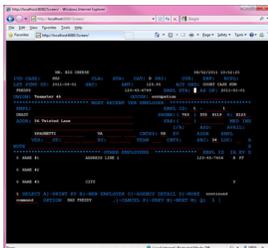
GETTING TO THE GOAL LINE

## Code Conversion

The code has the original Natural source embedded in it as comments and this makes it easy to find business logic when one is familiar with the structure of the application. The code is pure Java®, C# .NET, or COBOL code and object-oriented conversions are maintained by use of Microsoft Visual Studio (.NET) or through any of the Java IDEs now widely available such as Eclipse, IntelliJ, NetBeans etc. The full functionality of the IDE can be used to refactor, replace, rename etc.



Java in Eclipse



Map Post-Migration

The Modern Systems Tool emits modern web pages in the form of Java Server Pages (JSP) or Active Server Pages (ASP .NET) that implement the user interface. There is one page for each map and one page for each INPUT statement in the application that does NOT use a map. The JSP pages use a Custom Tag Library, and the ASP pages use a Custom Control (ASCX) to implement the functionality of Natural's map processor. During the modernization process, the maps are also emitted as XML documents, one document per map. Each XML document contains the entire map in simplified form. The XML provides a repository for tools that operate on maps. One of these tools is the Modern Systems Map Editor. This utility enables editing of maps in a comfortable drag-and-drop setting that is in some ways superior to the original Natural Map Editor. You can move items around the screen by dragging them, double click on an item to set attributes, you can set a customer renderer for a field such as drop-down list or date picker, or write your own renderer.

## Test, Refresh, Deploy

Modern Systems tests a subset of the refactored 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.