

The following examples shows how XINFO was used in several projects.

Example 1

We trialed XINFO for a long time -in the end the deciding factor was that with XINFO we where able to analyze the source code of 1,500 programs and find the whole call structure of the programs - including programs which used a certain variable.

Before we bought XINFO one person was able to change about 10 programs per days, with XINFO we changed 1,500 programs in 2 days.

Another reason was that with XINFO we could find UNIT statements. We had just installed a VTAS robot, and because of that we had to change a lot of JCL, changing procedures for a new SAS installation etc. XINFO simplified that completely.

Example 2

We have a large migration project.

In the first place XINFO helped us to decide which batch jobs should be converted and which modules are involved.

We also had to determine what the Language mix was for all relevant load modules used in these batch jobs for the conversion (for example whether they were Assembler, Cobol 1, Cobol 2 and if they were only Cobol 1 or Cobol 1 mixed with Cobol 2, etc.).

XINFO helped to do that very quickly. The load module analyzer especially was proved itself to be of great help.

Example 3

We recently bought another bank and we decided to incorporate the applications of that bank into our existing software inventory.

With XINFO we where able to rapidly create an impact analysis for the 15,000 Cobol programs, 25,000 QMF programs, JCL libraries, and PARMLIBs involved.

First we had to filter out unwanted programs.

How did we do that? We simply searched for batch programs, we used the flag in XINFO which says CICS=N to select them. Then we searched for any programs which ran in the last year (including weekly, monthly, quarterly and yearly runs) - we chucked out those

which are NOT in XINFO's SMF tables. However you must keep the subroutines which are called by programs that are in the list and are active; again this is a query in XINFO.

Now to complete the impact analysis, we wanted to know if those programs in the list of programs to be changed are Cobol 1 or Cobol 2 - this is to estimate the amount of work - because if we have a Cobol 1 program then we will at the same time have to convert it to COBOL 2. Again we used XINFO's load module scanner to match the programs on the list.

So instead of going manually through 15,000 programs we managed to focus on just 2,700 programs (of which 600 are Cobol 1 programs) which needed to be changed.

With XINFO we were able to scan the Cobol and QMF programs (we used the PO scanner to do that), JCL libs, PARMLIB involved and find the relevant strings, e.g. hardcoded names of existing banks and IF statements with questions about range of branches etc..

We also now knew how many changes we had to make in each program because XINFO had identified all the places in the programs which require changing. So we knew if we have one or twenty changes in a program. We even knew if those changes were difficult to make or not - again by categorizing the statements found by XINFO.

By using XINFO we were quickly able to estimate the size of the task we had set ourselves, and it continued to prove its worth throughout the whole project.