

OFFICE MEMORANDUM

TO: Steve McCullough, City Manager

PC: Sandy Cash, Deputy City Manager
Jon McCarty, Assistant City Manager
David Cardenas, Assistant to the City Manager
Pat McMacken, Emergency Management Coordinator

FROM: Nancy H. Bauer, Director of Information Services

DATE: January 17, 2000

SUBJECT: Information Services Year 2000 Report

The ideal situation would have been to report no glitches from the IS Y2K project. However, IS has had two minor Y2K problems found that were each corrected in less than a day.

One error was with the Code Enforcement synchronization program. When that program was run the second time in 2000, it thought that we had not synchronized in 1900 years, as it was then showing the year to be 100. This required IS to further clean programs off the old IBM ThinkPad laptops to be able to load a later Service Pak release on Microsoft Access. This required three people to re-enter their inspections for two days. This further emphasized the need to replace the old laptops.

The second error occurred in an Investment mainframe program written in Cygnet. The program "helped" us by stripping off leading zeros (the year) and calculating days passed as negative days. The mainframe programmers found the problem and immediately corrected it.

We are not fully convinced that all errors have been found. We will be watching all systems very carefully and expecting some possible glitches for February 29, 2000.

A sizable amount of money was spent making the City hardware and software Y2K ready. (Schedule attached showing total of \$1,441,917) All hardware that was replaced was either obsolete or had become undersized. However, had it not been for Year 2000, we would have had the time to move off the mainframe to newer software rather than replacing the mainframe and converting the old software. That additional time could have saved us \$731,917 to apply toward the new software, and therefore, I see that amount as the true IS cost to the City for Y2K.

I have attached full documentation of the IS Y2K project to allow you to see the magnitude of the job performed. I, too, am very proud of the staff and the job they did.

TEAM 2000 PREPARING FOR YEAR 2000

SUMMARY

Preparation for the year 2000 presented a daunting challenge for the City of Irving's Information Services Department. Near the end of 1997 testing determined that the existing IBM mainframe computer hardware and Operating System (OS) were non-compliant for the year 2000 and beyond. The mainframe applications were checked and all used two digits for the year and assumed the 19 for the first two digits.

A replacement mainframe was specified and purchased. It was placed in service in June 1998. The hardware was now ready and the focus shifted to converting software applications to handle four digit year dates. The desire was to expand all date fields, but detailed analysis determined that not enough time remained for this expansion.

The decision was made to window all applications except Courts. Since Courts modules contained year of birth (YOB) data, and since there are many people alive with YOBs before 1929 the windowing solution could not be used for the court applications.

WINDOWING SOLUTION EXPLAINED:

Windowing City of Irving applications was the method selected for all applications except the Courts. This is the solution that most business and government agencies selected, time and money driving the decision. The software giant Microsoft also chose to window its programs and applications.

Windowing simply depends on a pivot year in which the applications are told to assume the first two digits are 20 versus 19. The City used 1929 as our pivot year as did Microsoft Excel. This means the entry of a two-digit date from 30 through 99 is assumed to have the century digits of 19 associated. The other side of this method tells the computer to assume that 2-digit year entry of years 00 through 29 are to be assumed as having the prefix of 20.

Actually, the problem has been pushed into the future 29 years (70 years for Irving systems-- see below). The assumption being that all systems will have been replaced with fully functional four-digit year capability. If systems are not replaced by the year 2029 then the Y2K bug bites again. What other pivot dates were selected? Microsoft selected at least three different years. One year selected is rumored to be 2007. The windowing system that the City selected slides the window pivot year ahead by one year as each New Year arrives. This is not true of all systems by other agencies which windowed. The windowing performed by the City of Irving will not fail until the year 2070 at which time the dates can no longer be calculated with certainty.

METHODOLOGY

The first order of business required that all the COBOL source code be converted from DOS COBOL 1974 (a non-2000 compliant language version) to COBOL 1985. A tool

was selected for automating this conversion. The tool was CCCA, which could find and fix the majority, about 80%, of the changes to upgrade the old COBOL to the New COBOL. The rest of the changes had to be found and modified manually.

The City's application programs consist of 2749 individual modules. These are COBOL Source, Maps and Library modules. In addition, some applications were written in RPG and other languages that brought the total number of modules to just over 2900 to be modified.

These modules contain a total of 1,369,000 lines of code (LOC). Each of these LOCs had to be CCCAed, then processed through the IBS year 2000 tool. The code was re-checked, additional manual modifications made, and then tested in three phases before it was rolled out for operational use.

Additionally the City of Irving had numerous other systems including applications at the Police Department, Fire Department, Library, Finance, Garage, ICVB, ICTN and others which required hardware and or software upgrades to become year 2000 compliant.

The City of Irving also has more than one thousand personal computer workstations that required testing each for year 2000 readiness. Each of these workstations required testing and remediation ranging from software BIOS upgrades to total replacement of the hardware. All systems required the upgrade of the standard Microsoft products to bring them ready for year 2000.

FIRST OF THE YEAR 2000 PROBLEMS OCCUR IN 1999

The first working day of January 1999 the Information Services Department experienced and reported the first of the year 2000 problems with the applications on the mainframe. Several of the old applications had a feature that created new tables for the new year by adding 1 to the current two digit year 99 which then equaled 00 (since it was a two digit field). The programs then tested to see if 00 was greater than 99, since it was not, 1 was again added to 00 producing 01, then re-tested against 99, then 1 was added again producing 02, and on and on until it reached 00 again. This caused the program to be "hung" in an endless loop and fail to operate. IS had to produce immediate "hard coded fixes" to correct this problem on the old non-compliant code and hardware. This effort had to be done immediately and pulled resources away from the primary year 2000 tasks.

This type of small problems continued throughout 1999. Normally associated with a report license that was looking forward into 2000. Alarm permits is one example. The permit would show as expired since it was good until 00. These were addressed and workarounds were developed in the old code as they occurred. The finding and fixing of these "bugs" helped us understand just how poorly the older code had been written and clearly told Information Services that it must be re-written just to correct logic errors and poor coding practices. These bugs also provided the insight into areas in the old code that had never actually worked correctly. These problems in program logic and control were included in the code being ported to the new mainframe.

A REAL DEADLINE

The year 2000 transition represented the first “unmovable” deadline ever faced in recorded human history. The year 2000 would arrive on schedule. The City of Irving faced the need to have some applications ready before the New Year. The tax system for instance had to issue tax bills that were due the following year. Tax had to be solid and on line live before the tax statements could be mailed.

TESTING

The key to the successful remediation of software applications is testing. Only by testing can you know that all the errors have been found and that results are the same as those produced by the older non-compliant code.

All City of Irving departments with applications on the mainframe produced a baseline test. This involved performing all functions and running all reports and program outputs and checking them for correctness. When the baseline was complete the data for that application was copied to tape and thus “frozen” in time. Each successive test after every step in the remediation process was then run against the “frozen in time snapshot”, this allowed for an exact comparison of the results produced.

The virtual machine created for testing was moved backward and forward in “date” to first check the results in 1999, then again as December rolled into January 2000. Still further testing was done well into 2000 dates. All departments participated fully in testing their applications to assure that each change still produced full and correct results. This was a time consuming and labor intensive effort, but it found and identified hundreds of problems. Information Services corrected the problems found and the failed test was repeated until it was successful.

The final “testing” consisted of parallel operations using both the old code and the new code. This entailed dual entry of all daily operations in each virtual machine and checking daily results. This parallel testing found additional problems that were corrected and re-tested until the old and the new systems results were identical. In every case all applications were completed and in production operation on the new code before the New Year arrived.

The key to any large-scale migration or software systems change is testing. Testing requires 80 percent of the actual time required for the remediation. The hundreds of hours spent testing provided all departments and IS with the confidence that we were indeed year 2000 ready. This was born out when the real year 2000 arrived.

What applications were remediated and brought into year 2000 readiness? Listed on the next page are the mainframe applications that required conversion and remediation. No particular order of size or complexity should be inferred from the listed order.

Building Inspections
Water Utility Billing
Tax Billing
Courts
Accounts Receivable
Paving
Investments
Multi-Housing
Health
Community Action
Inspection Complaints
Alarm Permits
Bank Reconciliation
General Ledger

Garage was approximately 60% complete when it was abandoned to go with a Client/Server project.

Fixed Assets was approximately 20% complete when it was abandoned in favor of going to Banner

CCCA Conversion was completed for all mainframe systems

Mainframe replacement: IBM 4381 to IBM S390 Multiprise

VSE operating system replacement: version 1.3 to 2.3

Creation of 2 new VSE partitions

Smaller applications on PCs were located all over the City and required conversion. Listed below are some of these applications

Paradox Conversions

Community Development	Zoning Database
Community Development	Subdivision Cross Reference
Fire	Inspections
Fire Academy	Training Records
Fire Academy	Class Records
Police Academy	Firearms Records
Police Academy	Personnel Records
ICTN	Video Inventory
Public Health	15 different databases
Fire	Interface for Fire Alarms

New Applications written in Access:

Fire Emergency Contact Inquiry
 Dialer File Inquiry
 Locations Inquiry
 Streets Inquiry
 CAD Call Inquiry

Police Academy ITX (Training) – still in process of writing

Converted Garage work order system

Developed new client/server Garage application to replace the old mainframe application

System Upgrades

Upgraded the Operating System (AIX) and Unidata for the CAD system to make it Y2K compliant. Used the current version of the EAI software. Bought a new server for the application.

Paradox Databases

Fire Personnel database, time keeping, injuries, etc.
Building Inspections - Certificates of Occupancy, Inspections
Engineering - NOI Data
Finance - Check Reconciliation
 Legal Record Keeping
 Injuries
Parks - Personnel
Fire - Training Records
Library - Personnel Records
HR - Kardex- Personnel Records
Streets work order system and Streets Index

Hewlett Packard Servers

Purchased the new N4000 server including installation, and did upgrade of disk drives
Applied patches and software upgrades to the N4000
Upgraded/removed legacy software on the existing HP K200 and E35 which required approximately 200 patches and 25 pieces of software.
Upgraded the printer network cards from NetJet/NetQue to HP JetDirect ~ 60 printers

PERSONAL COMPUTERS AND THE CITY NETWORK

In addition to software applications Information services had to test and certify all network hardware and software. The network printers incorporate a card or device known as a print server. This allows the printer to be used by everyone on the network. It was found that 47 print servers purchased before 1998 were not year 2000 ready. These were replaced. IS also replaced 160 PCs mostly older NEC machines that were not and could not be made ready for the year 2000. This was accomplished largely using the in-place computer replacement policy.

Operating system software in the City's routers required an upgrade to be year 2000 ready.

MID-RANGE COMPUTERS

Three older Micro VAX computers, which were non-compliant, were replaced. One of these machines was the old GIS system computer and the remaining two were the Police CAD and Records Management machines. The operating systems of two DEC Alpha machines were upgraded to a year 2000 ready version.

Banner and BTI Year 2000 related milestones

Brought production databases to Y2K compliant (Oracle 8.0.5 for Banner, Oracle 7.3.3.6 for BTI)

Migrated Banner Databases to new HP N4000

Migrated BTI Databases to NT 4.0 Server

Upgraded Banner products to release 4.0 and applied all Y2K patches

Upgrade BTI software to release 5.32 Nov. 1, 1999

Upgraded Developer 2000 to Y2K compliant version 1.6.1 and recompiled all Banner forms.

Converted Bid Contact application for Warehouse

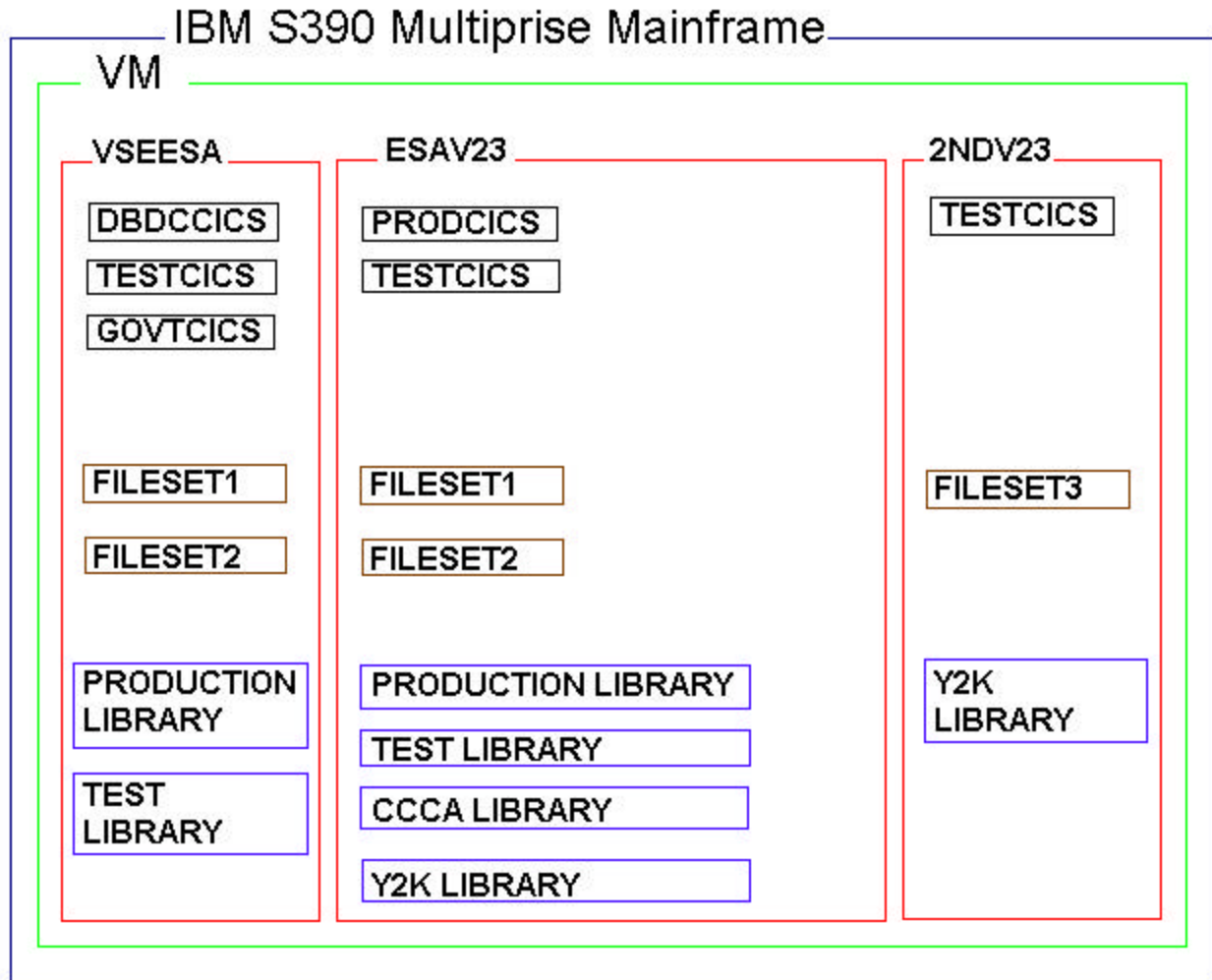
Developed interface for Fixed Assets Module in Banner

All Y2K testing for releases.

The operating systems were replaced on the mid range systems which support the Library. The vendor for the library system installed year 2000 patches and bug fixes to the Library application.

The detailed process on exactly how the mainframe applications were remediated and tested has been listed below. It is included for those wishing to more thoroughly understand the long technical process.

THE MAINFRAME Y2K PROCESS



Generally, the new mainframe's foundation is the VM operating system. Within VM, multiple VSE systems can be run. In our case we have three: VSEESA, ESAV23, and 2NDV23. VSEESA is an identical replica of the old 4381 mainframe system. ESAV23 is our new production VSE. 2NDV23 is a test VSE that was used as an independent area to modify system dates and test programs for Y2K. The process of getting the software ready for Y2K was broken up into multiple steps; producing baseline data, converting/upgrading the COBOL, performing the Y2K remediation, performing extensive Y2K testing, and performing the parallel run.

BASELINE TESTING

The first thing that we did after the mainframe system was set up was to move all the data from FILESET1 on VSEESA to FILESET1 on ESAV23. We also moved all the programs from the PRODUCTION LIBRARY on VSEESA to the PRODUCTION LIBRARY on ESAV23. Using ESAV23, we then assisted departments to run the first tests on the mainframe for BASELINE results using ESAV23 PRODCICS, the PRODUCTION LIBRARY, and FILESET1.

CCCA CONVERSION

In the second major step, we ran all the programs through a COBOL upgrade conversion using an IBM tool known as CCCA. The reason that this had to be done was because the old version of COBOL-74 that had run on the IBM 4381 mainframe was not Y2K compliant and had to be replaced with COBOL-85. Since the two COBOL standards have some significant differences, the programs had to be modified to the new language. The conversion tool took the programs from the ESAV23 PRODUCTION LIBRARY, converted or flagged what it recognized as a problem with the old COBOL, and output them to the CCCA LIBRARY. The programmers then evaluated the results and rebuilt any programs that CCCA flagged, but could not fix without human intervention. The data from the old mainframe was reloaded into FILESET2 for use in testing. When this process was complete, departments performed their testing with the converted COBOL programs using ESAV23 TESTCICS, the CCCA LIBRARY, and FILESET2.

Y2K REMEDIATION

In the third major step, we began the actual Y2K remediation of the programs. The COBOL and RPG programs were downloaded through the network from the CCCA LIBRARY to a specially configured PC, where the IBS Y2K remediation tool was installed. This tool searched for date related variables and issues and then attempted to correct what it could and flag the problems that needed human intervention. The programs were usually run through the tool twice. The first time was to get a general idea of how many Y2K issues there were with a program or set of programs, which allowed us to reconfigure the IBS tool to help it make better 'decisions' in fixing or flagging the code. We then ran the programs through the reconfigured tool. The programs were then uploaded through the network from the IBS PC to the ESAV23 Y2K LIBRARY on the mainframe. The programmers then reworked the programs to second check the coding changes that the tool had made to make sure that they were correct. They also inserted or reworked the code that had been flagged by the IBS tool as needing a 'human' decision as to how to proceed with the fixes.

The IBS tool was not able to be of assistance with the programs that had been written in the CYGNET language. All of these programs had to be searched, line-by-line, by a programmer. They were all printed on green bar paper and then manually searched and documented. These programs were taken from the ESAV23 PRODUCTION LIBRARY, edited to perform the documented changes, and output to the ESAV23 Y2K LIBRARY.

When all the programs for a system had been remediated, the object code was copied from the ESAV23 Y2K LIBRARY to the 2NDV23 Y2K LIBRARY. The data from the old mainframe was reloaded into FILESET3 for use in testing. Any data files, in FILESET3, where one or more fields were expanded within the programs, had to have the old data expanded. This was achieved using COBOL programs that were written by the programmers. Using the 2NDV23 TESTCICS, the 2NDV23 Y2K LIBRARY, and FILESET3, the programmers performed some general unit testing of the system.

Y2K TESTING

When the programmers felt like a system was generally functional, the system was ready for Y2K user testing. The users and the programmers worked together to test, check, and fix any problems that were encountered during this testing phase, until the users were happy with the results and felt like the system was working properly. These tests were performed using 2NDV23 TESTCICS, the 2NDV23 Y2K LIBRARY, and FILESET3. This testing phase encompassed 5 full tests whereas the testing up to this point had only been one test per test session. The five different tests were performed with the 2NDV23 VSE set to various specific dates that would provide the date challenges to the system that would be encountered during the year 2000.

PARALLEL RUN

When the first system was preparing for the first parallel run, the ESAV23 PRODUCTION LIBRARY was emptied, so that only Y2K remediated programs would be put there. The reason for this was to try to cull out all the programs that had been in the production libraries that had been deemed unused, during this whole process. This would allow the ESAV23 PRODUCTION LIBRARY to be a great deal more organized than the old production libraries had been for the past several years.

When the 5 tests for test phase three were complete for each system a schedule was set up to prepare for the parallel run. Over a weekend, the system would be set up for the parallel. The programmer would move the programs from ESAV23 Y2K LIBRARY (at this point they should be identical to the programs in 2NDV23 Y2K LIBRARY) to ESAV23 PRODUCTION LIBRARY. The data would be copied from VSEESA FILESET1 to ESAV23 FILESET1. The data would be expanded, using the programs that were written earlier, by the programmer. The CICS screens would be tested. All Friday evening processing would be run on old and new systems as a first test to verify that the data had not been corrupted in some way.

In our parallel run, the systems on VSEESA and ESAV23 were run simultaneously as production systems for a specified amount of time or until the department and the programmers felt like the new system on ESAV23 was running properly. The new system is run using ESAV23 PRODCICS, ESAV23 FILESET1, and ESAV23 PRODUCTION LIBRARY. Every daily process was duplicated on both systems and then reports, data browses, bills, and any other output were checked to verify that both systems were producing the same output. When the department and the programmers felt like the new system was working as it should on ESAV23, the old system was abandoned on VSEESA. When it could be done, precautions were taken to try to keep departments from accidentally getting into the old system and working on the old system.