+ Fresche Legacy

IBM i Insights

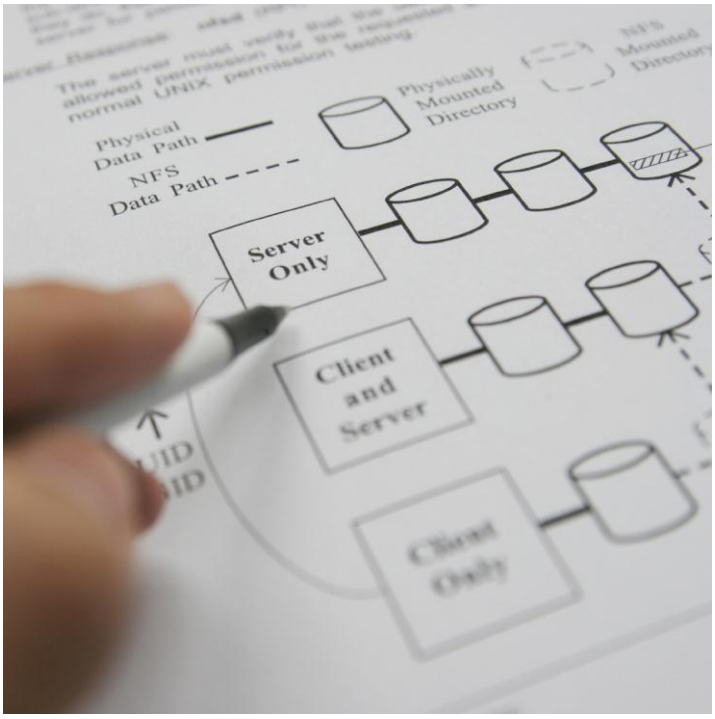
+

Understanding the Secrets of Your IBM i Code

Application Documentation – The First Step Toward Maintaining, Managing and Modernizing

By Stuart Milligan,
Databorough – A Division of Fresche Legacy

You cannot manage what you do not measure



Knowing What You Have

“If I was responsible for a million of anything, I would want to know exactly what I had at any one time.”

Our Insight

At Databorough, a Division of Fresche Legacy, our expertise has been developed and honed through the analysis and documentation of billions of lines of IBM i code.

There continues to be irony in the fact that IT organizations – run by the very people responsible for building excellent business software to accurately manage assets and inventory – so regularly disregard the value of using this same inventory management concept when managing their own software assets.

However, with modernization projects looming on the horizon, and the risk of brain drain due to retirement and attrition, an increasing number of IBM i development sites are now looking to forensic-based data to understand their code base and better manage their available development resources. Typically, organizations are turning to code analysis to answer the following two questions:

- How much code do we have to maintain/modernize?
- How can I better understand and control the quality of my code base?

The Issue

IBM i managers still are largely unsure what they've got. But not knowing your system leads to planning and estimation problems for maintenance, development and modernization. Understanding the scope of general or specific maintenance tasks and modernization projects is crucial to success.

Poor Visibility at 10,000 Feet

Unfortunately, there are still many IT organizations with ongoing dependency on IBM i who don't know precisely how much of their code base is relevant, how truly complex any part of it is, what business rules it implements, what the relational model is, or any other metric consistent with a structured management process.

It is not uncommon for IT directors in multi-billion dollar organizations to have no real idea of how much code they are really responsible for, or how low-tech their development management processes are.

In a recent power systems user group encounter, a long-time, seasoned IT manager confessed to having responsibility for "thousands" of programs, but had no precise data on the size of that code base, or how much of it was redundant. He also had no explicit design information about the system beyond what was written down 20 years ago, at the time the system was first built.

It is equally revealing (and satisfying), however, to see the expression of wonder and amazement on the faces of people who see something they have worked on for 30 years quantified, measured and visualized for the first time. Without exception, this 'bird's eye view' provides a new and more effective way of working and managing their IT systems and development processes.

Typically, one of the key deliverables of a code visualization effort is a detailed list of software problems. This list can include software and object mismatches, missing source, unused programs, files, access paths, code and a number of legacy constructs such as GOTOs, internally described files and multi-format files.

The initial wonder of seeing a system visualized quickly gives way to surprise and unease as the scope of these problems is exposed. This dismay, however, quickly translates into a proactive action plan to fix or remove problems from mission-critical parts of the existing application portfolio.

Understanding the scope of general or specific maintenance tasks and modernization projects, and accurate planning and costing can make or break an IT manager's career. Not to mention that the platform often ends up taking the blame for seat-of-the-pants planning. "Oh it's that legacy platform again! That's why we never get any new enhancements done on time!" Anyone familiar with developing in RPG or SYNON on IBM i will argue back that it is probably the most productive kit around for business application development.

Code Analysis

X-Analysis, the leading tool for application documentation and impact analysis on the IBM i, helps managers quantify and manage the size and quality of their code base, assess the risk of maintaining and modernizing legacy code bases, and even improve development quality and standards.

Is Source Code Management or RTC the Key?

Some IBM i shops use basic cross-referencing and some form of Source Change Management (SCM) to help manage development. And there is a widespread misconception that SCM can help control inventory of software assets. This is untrue. SCM allows or provides for the control of specific assets for a specific purpose, at a given time. It provides no meaningful management data on the entire system either before or after a change or series of changes.

At the opposite end of the spectrum there is IBM's Rational Team Concert (RTC), which offers an excellent portfolio of features and modules that help in managing large, complex development projects with many developers and stakeholders. RTC is a terrific solution when an IT shop is engaged in developing a new code base, with multiple teams across multiple locations.

However, IBM i development and modernization efforts don't fit that model. In these scenarios, the focus is on small teams responsible for maintaining or modernizing very large existing code bases. The use of RTC in this situation may actually discourage IBM i developers. In addition, the support for deep code analysis of RPG/SYNON IBM i applications within RTC is very limited.

The Best Option: Code Analysis

IT managers are increasingly turning to code analysis to improve visibility. Tools such as X-Analysis are instrumental in quantifying and managing the size and quality of your code base, assessing the risk of maintaining and modernizing legacy code bases, and improving development quality. By abstracting the code base one level above its syntax and using various interactive diagrams and pseudo code in place of RPG, non-IBM i developers and stakeholders are exposed to the proven designs and value of IBM i legacy applications. Most analysis tools integrate with SCM tools so that key decisions resulting from the analysis are implemented in a structured and methodical manner.

Figure 1: Summary Metrics

Summary Metrics

Metrics Analysis for XAN4CDXA (Double Click to View History)

Complexity Level	Units	Source Lines	Cyc. Complex.	Halstead	Maint.Index	Files	Device Files	Called Programs	Calling Programs
Grand Total	120	11,808	1,750	90,333	7065	222	51	165	191
Interactive Source Members	50	9,842	1,594	81,830	5694	163	51	136	113
High Total	5	3,543	435	21,583	850	34	5	31	14
Average Total	13	4,036	374	20,036	1962	46	13	46	20
Low Total	32	2,263	785	40,211	2882	83	33	59	79
Batch Source Members	70	1,966	156	8,503	1371	59	0	29	78
High Total	0	0	0	0	0	0	0	0	0
Average Total	1	421	33	3,408	99	4	0	1	3
Low Total	69	1,545	123	5,095	1272	55	0	28	75

Figure 2: Screen Metrics

Summary Metrics | Screen Metrics

Screen Metrics for XAN4CDXA

Complexity Level	Units	Files	Database Fields	Work Fields	Outgoing Calls	Incoming Calls	Function Keys	Conditioning Fields
Grand Total	88	131	664	324	64	30	36	734
High	6	6	55	13	6	0	6	48
CON001	2	2	12	2	0	0	0	16
CUSFMAINT	1	1	18	3	3	0	3	15
CUSFMOLD	1	1	18	3	3	0	3	15
DSPDISTS	1	1	2	2	0	0	0	1
DSPPTYPES	1	1	5	3	0	0	0	1
Average	71	114	590	309	58	30	30	685
Low	11	11	19	2	0	0	0	1

Problem Analysis

X-Analysis, a state-of-the-art analysis tool, provides a software problems list. This list can include software and object mismatches, missing sources, unused programs, files, access paths, code, and a number of legacy constructs such as GOTOs, internally described files and multi-format files.

Figure 3: Problem Analysis

Problem Analysis | Summary Metrics | Screen Metrics

Problem Analysis data for XAN4CDXA - Total Problems: 529

Alert/Category/Object	Total
Source/Object Alerts	12
Source member changed after devicefile created	21
No file found for existing source member	6
No source member for file	1
No program object found for source member	17
No source member for program	17
Source member changed after file created	67
No device file found for existing source member	8
No source member for device file	4
Referenced data area does not exist	4
Referenced device file does not exist	1
Referenced program object does not exist	18
No source member for copy book	0
Database Alerts	3
File has Constraints	0
Files with zero members	1
Internally described file	3
Program Code Alerts	7
Greatest depth of nested ELSEs exceeds 1	18
Number of GOTOs exceeds 0	11
Greatest depth of nested IF/DOs exceeds 5	12
Greatest IF/DO block nbr of lines exceeds 48	28
Greatest depth of nested loops exceeds 1	38
Greatest subroutine nbr of lines exceeds 80	23
Program has (non-excluded) hardcoded libraries	5
Migration Alerts	2
File has Select/Omit rules	10
File has Triggers	0
Others	4
Update date on source and object do not match	79
Unused Subroutines	5
Unused Procedures	108
Unused Logical files	24

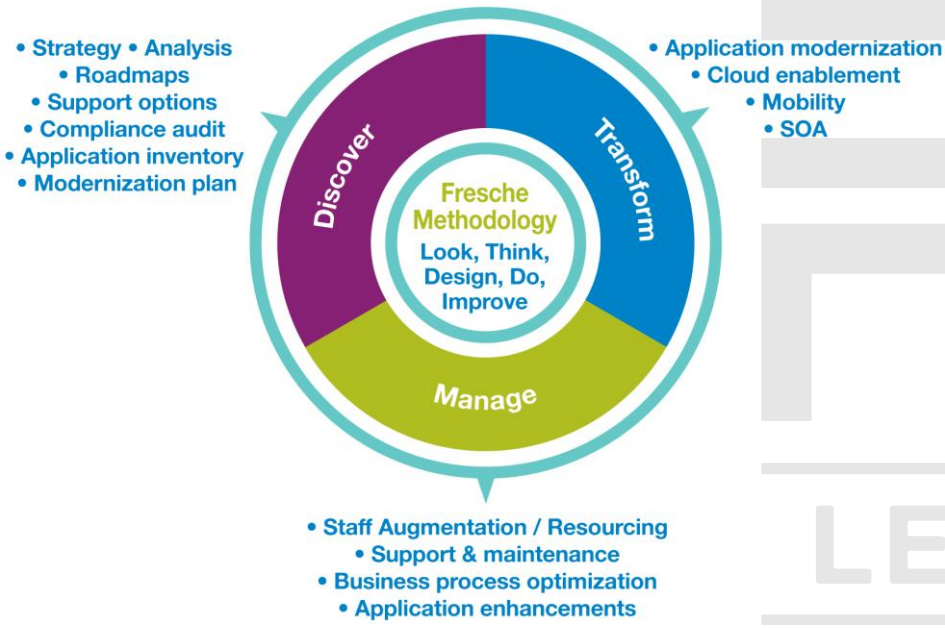
In Summary

The decrease of available RPG resources, rising maintenance costs, ever growing code base, and widening diversity in technology and skills is creating a tipping point for a more scientific and measurable approach to IBM i development and modernization. Whether it is the organization's plan to modernize or simply maintain the IBM i systems running the business today, improved understanding of the code base through structured code analysis will result in improved business understanding and a clearer roadmap for the future. For more information about solutions for documenting, design recovery, reengineering and rebuilding RPG, CA2E and COBOL legacy applications on the IBM i, go to www.databorough.com, or e-mail us: info@freschelegacy.com .

About Fresche Legacy

As a leading expert in legacy management and modernization, Fresche Legacy helps enterprise organizations transform their business to improve financial performance, increase market competitiveness, remove risk and add business value. Our team of experts has successfully completed hundreds of transformation projects within the most complex enterprise environments, helping organizations future-proof their business by modernizing their business processes, technologies, infrastructure, and methodologies. Committed to 100 percent customer satisfaction, Fresche Legacy's services and solutions span the complete legacy modernization spectrum from concept to maintenance, and include Discovery Services, Modernization Solutions, and Application Management Services & Transformation. For more information about our company, visit us on the Web at www.freschelegacy.com

Are you an enterprise organization seeking solutions for your legacy environment? Drop us a line at info@freschelegacy.com or call us at 1-800-361-6782



FRESCHE LEGACY