IBM

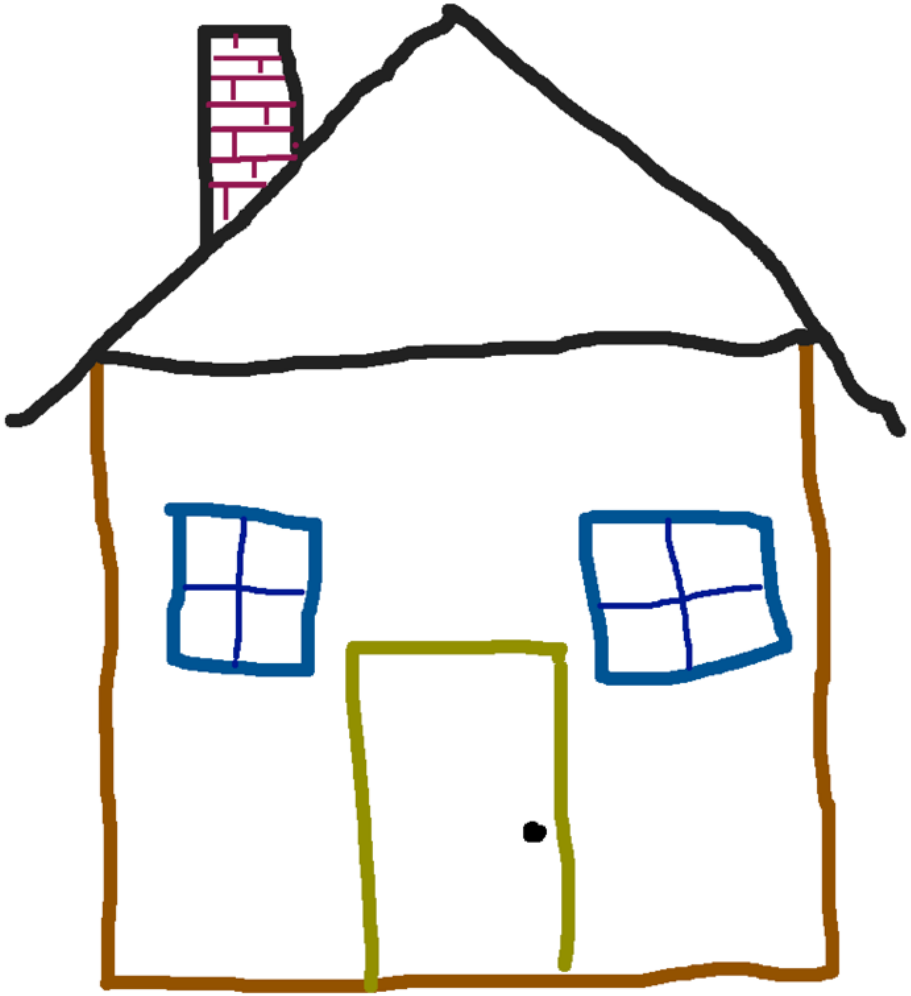
Cognitive Systems

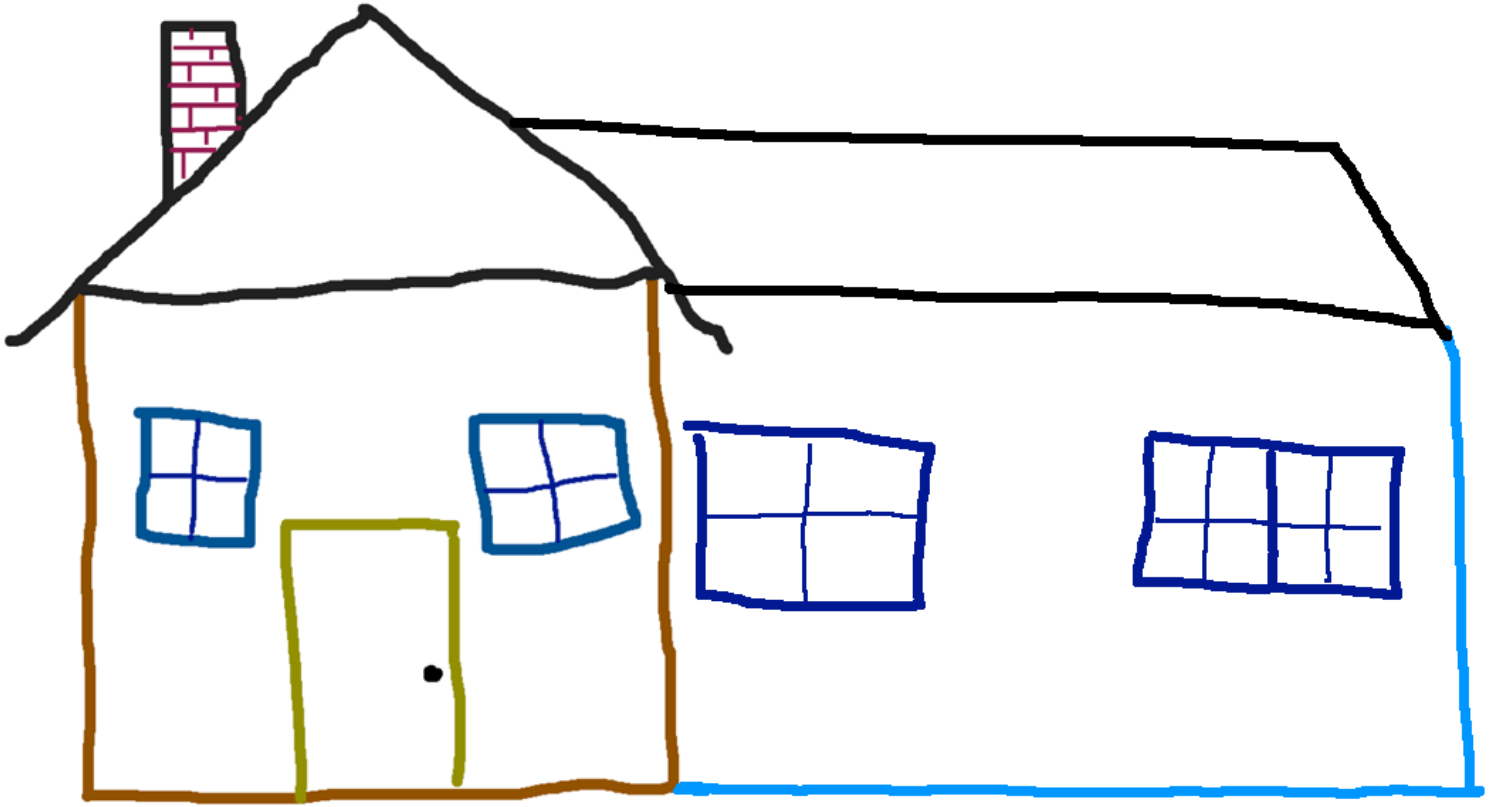
What is Modernization on IBM i ?

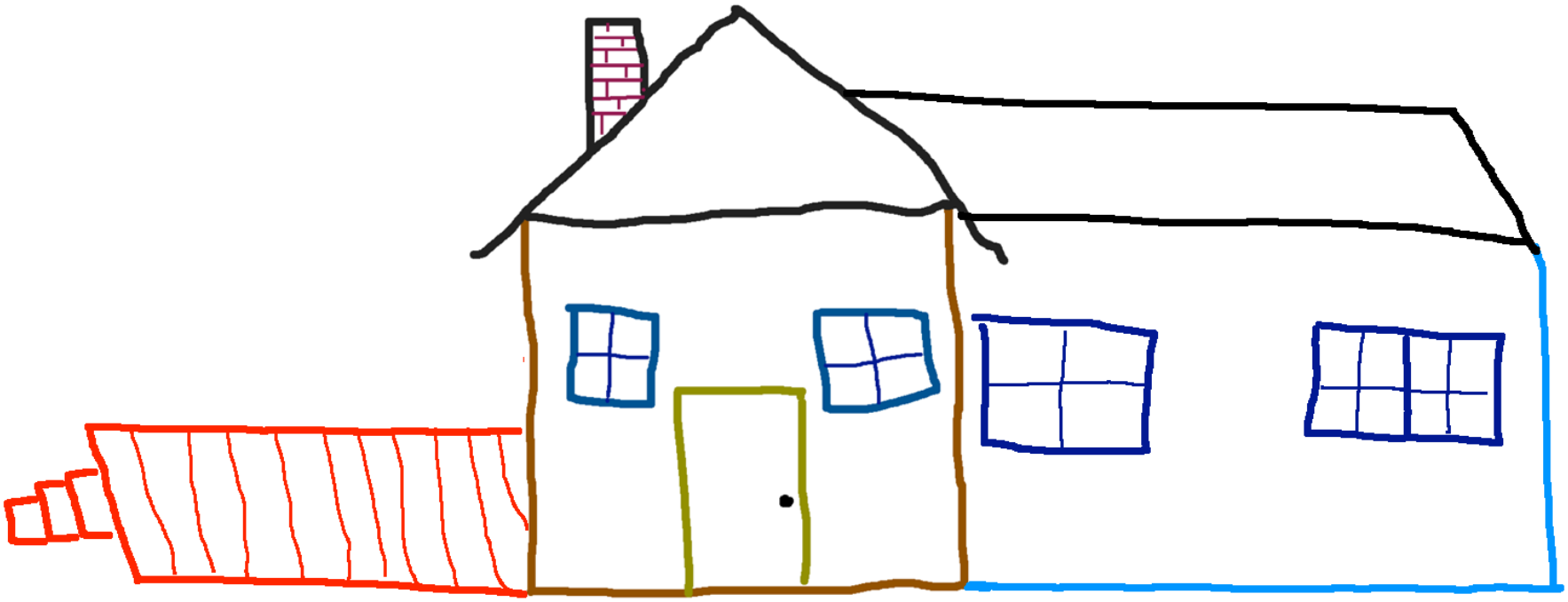
Tim Rowe- timmr@us.ibm.com
Business Architect Application Development

Modernization on IBM i

- Who are your developers ?
- What are the tools you use ?
- How are you doing development today ?
- Are you using the ‘best’ building blocks ?
- Building your self ? Leveraging Partners ?
- How is the IBM i team helping you move into the future ?



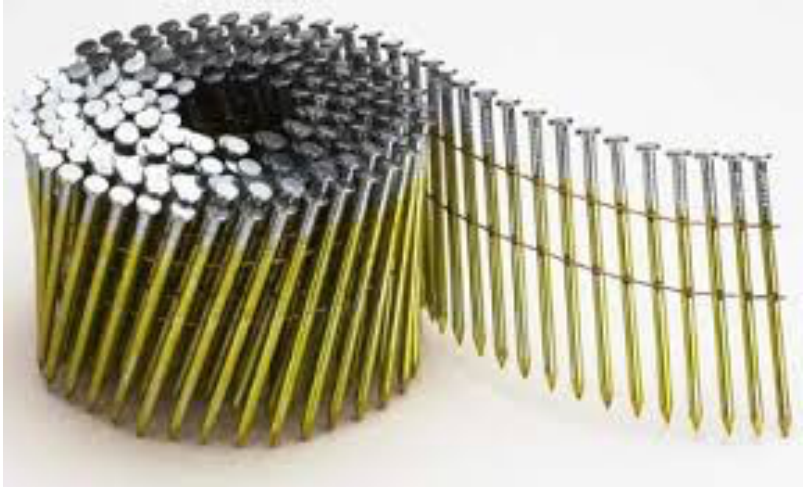


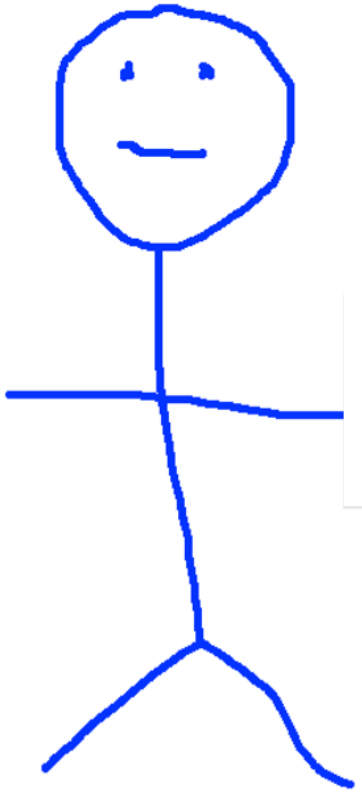


How do you avoid these types of mess ups ??

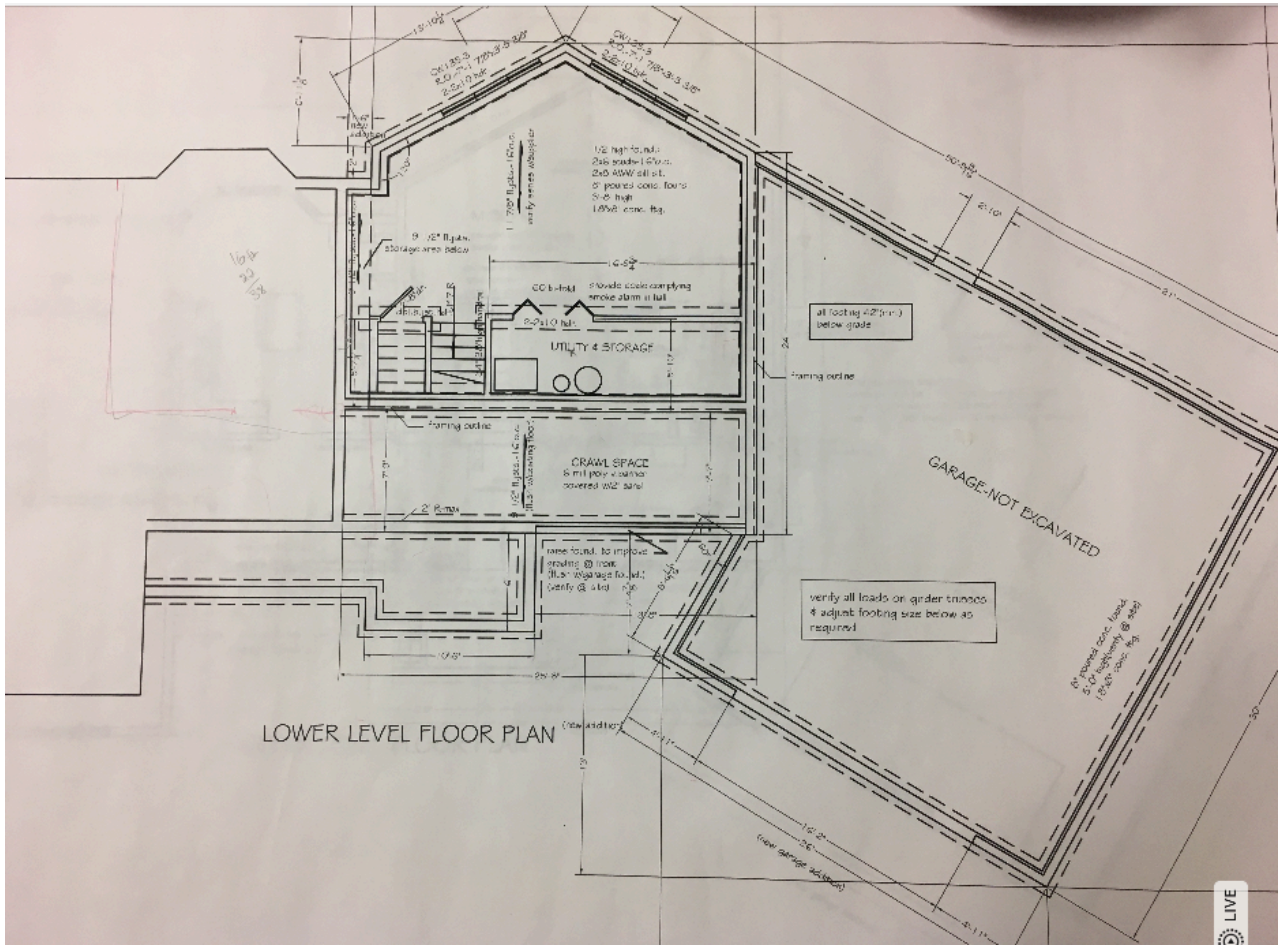
- Do you have a Site plan
- Have a BEFORE blueprint
- Have an AFTER blueprint
- Understand the plan before you start
- Use the correct material
- Use the right Tools!!!!
- Right people

What's the Plan??





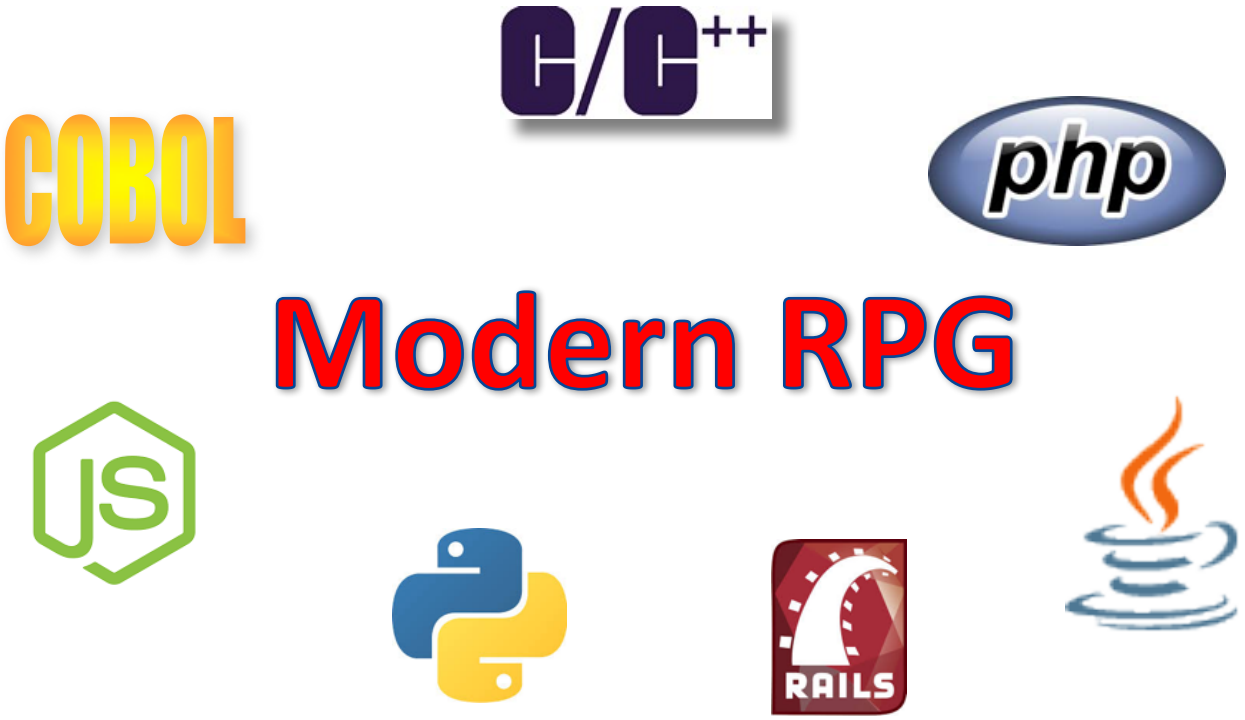
Joe Carpenter



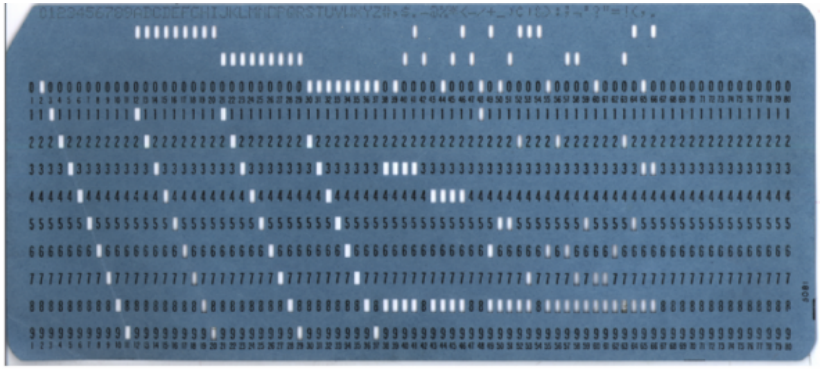
Modernize Applications…..

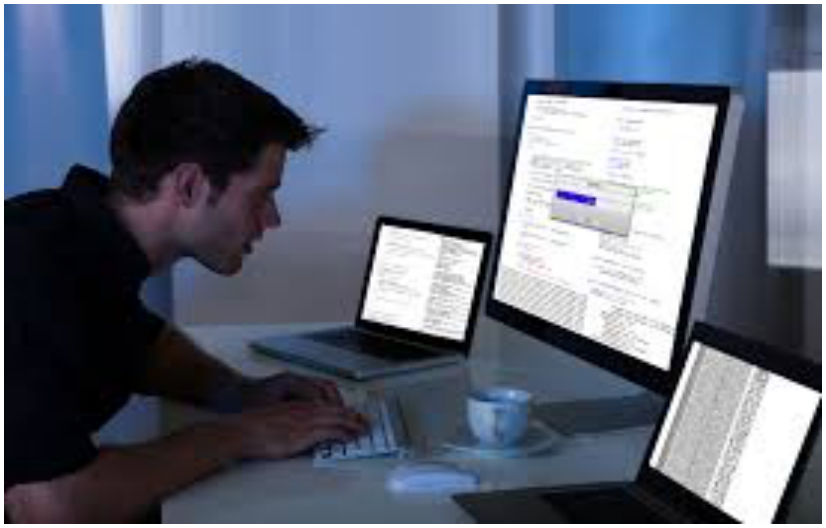
- Move to 'Modern Language'
- Self Documenting Code
- Testing anyone?
- Leverage Todays Developers
- Transition from Monolithic to Modular
  - Impact Analysis
  - Get a Blueprint
  - Data Maps
- Interactive User Interfaces
- Connecting to the Data
- Agile development and deployment

Languages of IBM i



Modern RPG – Constant Transformation





RPG IV - A Modern Business Language

- Interoperability
  - Java
  - XML
  - SQL
- Readability
  - Free form
  - Blank lines
  - Comments
- Functionality
  - Procedures
  - Data areas
  - Data structures
  - More data types
  - Extended file support
- Modern Tools
  - RD i, RTC, ARCAD Power Pack

```
    read  file;                 // Get next record
    dow not %eof(file);         // Keep looping with record
        if %error;
            dsply 'The read failed';
            leave;
        else;
            chain(n) name database data;
            time = hours * num_employees
                       + overtime_saved;
            pos = %scan (',' : name);
            name = %xlate(upper : lower : name);
            exsr handle_record;
            read file;
        endif;
    enddo;
begsr handle_record;
    eval(h) time = time + total_hours_array (empno);
    temp_hours = total_hours - excess_hours;
    record_transaction();
endsr;
```

Full Free Form RPG


```
ctl-opt bnddir('ACCRCV');

dcl-f custfile usage(*update);
dcl-ds custDs;
dcl-f report printer;

read custfile custDs;
dow not %eof;
   if dueDate > %date(); // overdue?
      sendOverdueNotice ();
      write reportFmt;
      exec sql insert :name, :duedate into
             mylib/myfile;
   endif;
   read custfile custDs;
enddo;
inlr = '1';

dcl-proc sendOverdueNotice;
    /copy invoices
   sendInvoice (custDs : IS_OVERDUE);
 end-proc;
```

Compiler Options

File Definitions

Data Definitions

Procedure Definitions

Calc Specs

Modern RPG

- 8-80 Column restriction now gone
- Nested Data Structures
- %MIN & %MAX
- ALIGN(*FULL)

- New requirements
  - Continue to adopt object oriented principles in RPG
  - Deal with JSON in a Native manner
  - Simply dealing with arrays
  - RPG is driven by you…
    - RFE community is where you get a voice
    - And where we get work to do!!!

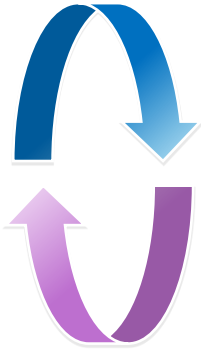
https://www.ibm.com/developerworks/rfe/execute?use_case=changeRequestLanding

How is this changing the game?

Hear how Modern RPG and Tools allows a 23 year old developer to be success today!

http://bit.ly/1WhM7uT

How Do I Convert to Modern RPG ??



April 2017 Announce…

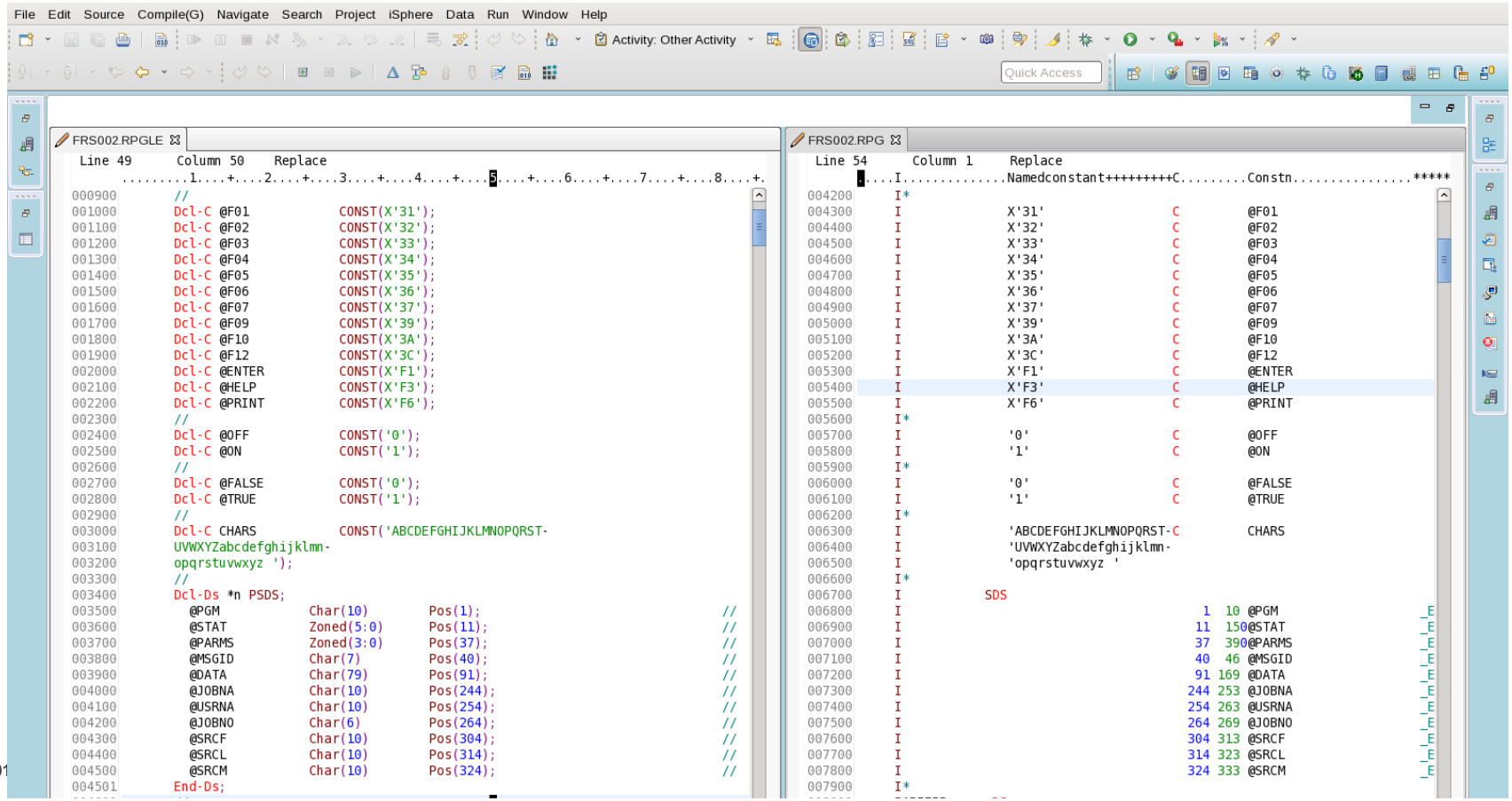
ARCAD Converter is now included in EES

ARCAD Converted for i - 5733-AC1

- Plugs into RDi
- Green Screen interface for MASS conversion
  - Convert your entire source file at once

https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS217-151&appname=lenovous&language=en

Input Specs and Data Structures -> Declarations




```
000900       //
001000       Dcl-C @F01            CONST(X'31');
001100       Dcl-C @F02            CONST(X'32');
001200       Dcl-C @F03            CONST(X'33');
001300       Dcl-C @F04            CONST(X'34');
001400       Dcl-C @F05            CONST(X'35');
001500       Dcl-C @F06            CONST(X'36');
001600       Dcl-C @F07            CONST(X'37');
001700       Dcl-C @F09            CONST(X'39');
001800       Dcl-C @F10            CONST(X'3A');
001900       Dcl-C @F12            CONST(X'3C');
002000       Dcl-C @ENTER          CONST(X'F1');
002100       Dcl-C @HELP           CONST(X'F3');
002200       Dcl-C @PRINT          CONST(X'F6');
002300       //
002400       Dcl-C @OFF            CONST('0');
002500       Dcl-C @ON             CONST('1');
002600       //
002700       Dcl-C @FALSE          CONST('0');
002800       Dcl-C @TRUE           CONST('1');
002900       //
003000       Dcl-C CHARS           CONST('ABCDEFGHIJKLMNOPQRST-
003100       UVWXYZabcdefghijklmn-
003200       opqrstuvwxyz ');
003300       //
003400       Dcl-Ds *n PSDS;
003500         @PGM            Char(10)       Pos(1);                //
003600         @STAT           Zoned(5:0)     Pos(11);               //
003700         @PARMS          Zoned(3:0)     Pos(37);               //
003800         @MSGID          Char(7)        Pos(40);               //
003900         @DATA           Char(79)       Pos(91);               //
004000         @JOBNA          Char(10)       Pos(244);              //
004100         @USRNA          Char(10)       Pos(254);              //
004200         @JOBNO          Char(6)        Pos(264);              //
004300         @SRCF           Char(10)       Pos(304);              //
004400         @SRCL           Char(10)       Pos(314);              //
004500         @SRCM           Char(10)       Pos(324);              //
004501       End-Ds;
```

```
004200     I*
004300     I              X'31'                C         @F01
004400     I              X'32'                C         @F02
004500     I              X'33'                C         @F03
004600     I              X'34'                C         @F04
004700     I              X'35'                C         @F05
004800     I              X'36'                C         @F06
004900     I              X'37'                C         @F07
005000     I              X'39'                C         @F09
005100     I              X'3A'                C         @F10
005200     I              X'3C'                C         @F12
005300     I              X'F1'                C         @ENTER
005400     I              X'F3'                C         @HELP
005500     I              X'F6'                C         @PRINT
005600     I*
005700     I              '0'                  C         @OFF
005800     I              '1'                  C         @ON
005900     I*
006000     I              '0'                  C         @FALSE
006100     I              '1'                  C         @TRUE
006200     I*
006300     I              'ABCDEFGHIJKLMNOPQRST-C         CHARS
006400     I              'UVWXYZabcdefghijklmn-
006500     I              'opqrstuvwxyz '
006600     I*
006700     I          SDS
006800     I                                      1  10 @PGM                _E
006900     I                                     11  150@STAT               _E
007000     I                                     37  390@PARMS              _E
007100     I                                     40  46 @MSGID              _E
007200     I                                     91 169 @DATA               _E
007300     I                                    244 253 @JOBNA              _E
007400     I                                    254 263 @USRNA              _E
007500     I                                    264 269 @JOBNO              _E
007600     I                                    304 313 @SRCF               _E
007700     I                                    314 323 @SRCL               _E
007800     I                                    324 333 @SRCM               _E
007900     I*
```

Creates Case Statements

```
021600       INZSR();
021700       //
021800       DoU @EXIT = @TRUE;
021900       //
022000         Write MSGCTL;                                    //
022100         Exfmt FRSUPD;                                    //
022200       //
022201         Select;                                          //
022300           When KEY = @F02;                               //
022301             @F02SR();
022400           When KEY = @F03;                               //
022401             @F03SR();
022500           When KEY = @F04;                               //
022501             @F04SR();
022600           When KEY = @F05;                               //
022601             @F05SR();
022700           When KEY = @F07;                               //
022701             @F07SR();
022800           When KEY = @F09;                               //
022801             @F09SR();
022900           When KEY = @F12;                               //
022901             @F12SR();
023000           Other;                                         //
023001             @ERRCK();
023100         EndSl;
023200       //
023300         If KEY = @F06                                    //
023400           and *IN50 = *OFF;                              //
023500           @F06SR();                                      //
023600         EndIf;                                           //
023700       //
023800         If KEY = @F10                                    //
023900           and *INLR <> *ON                               //
024000           and *IN50 = *OFF;                              //
024100           @F10SR();                                      //
024200         EndIf;                                           //
024300       //
024400       EndDo;                                             //
024500       //
024700       //
024800       *INLR = *ON;                                       //
083000       //
083100       //
```

```
021600     C                   EXSR INZSR
021700     C*
021800     C           @EXIT   DOUEQ@TRUE
021900     C*
022000     C                   WRITEMSGCTL                     * WRITE MSGCTL
022100     C                   EXFMTFRSUPD                     * DISPLAY SCREEN
022200     C*
022300     C           KEY     CASEQ@F02       @F02SR          * F02 REFRESH
022400     C           KEY     CASEQ@F03       @F03SR          * F03 EXIT
022500     C           KEY     CASEQ@F04       @F04SR          * F03 EXIT
022600     C           KEY     CASEQ@F05       @F05SR          * F03 EXIT
022700     C           KEY     CASEQ@F07       @F07SR          * F03 EXIT
022800     C           KEY     CASEQ@F09       @F09SR          * F03 EXIT
022900     C           KEY     CASEQ@F12       @F12SR          * F12 CANCEL
023000     C                   CAS             @ERRCK          * CK FOR ERRORS
023100     C                   ENDCS
023200     C*
023300     C           KEY     IFEQ @F06                       * F03 EXIT
023400     C           *IN50   ANDEQ*OFF                       * 50 OFF=NO ERRS
023500     C                   EXSR @F06SR                     * CREATE RECRD
023600     C                   ENDIF                           * END IF
023700     C*
023800     C           KEY     IFEQ @F10                       * F10 PRESSED
023900     C           *INLR   ANDNE*ON                        * F10 PRESSED
024000     C           *IN50   ANDEQ*OFF                       * 50 OFF=NO ERRS
024100     C                   EXSR @F10SR                     * CREATE RECRD
024200     C                   ENDIF                           * END IF
024300     C*
024400     C                   ENDDO                           * END DO
024500     C*
024600     C           END     TAG
024700     C*
024800     C                   MOVE *ON        *INLR           * LAST RECORD=ON
024900     C*
025000     C*
025100     C*
025200     C           INZSR   BEGSR                           * SET OUTPT SCREE
025300     C*
025400     C                   SETOF                     39    * UNPROT TIL F6
025500     C                   SETOF                 404450    * ERROR IND. SC 1
025600     C                   SETOF                 515253    * ERROR IND. SC 1
025700     C                   SETOF                 545556    * ERROR IND. SC 1
025800     C*
```

Subroutines converted to Procedures

```
Dcl-Proc INZSR;                                                    //
//
  *IN39 = '0';                                                     //
  *IN40 = '0';                                                     //
  *IN44 = '0';
  *IN50 = '0';
  *IN51 = '0';                                                     //
  *IN52 = '0';
  *IN53 = '0';
  *IN54 = '0';                                                     //
  *IN55 = '0';
  *IN56 = '0';
//
  %Subst(MSGID:7:1) = ' ';
  MSGDTA = *Blanks;
  MSGQUE = @PGM;
//
  @EXIT = @FALSE;
  PGMQ = @PGM;
//
  ORDNUM = PORDER;                                                 //
//
  Chain ORDNUM ORDNAME;
  *IN44 = not %Found;
//
  FILDPK = DEPART;
  DOFYR = %Dec(%XLate(' ':'0':
        FILYY):4:0);
  DOFMON = %Dec(%XLate(' ':'0':
        FILMM):2:0);
  DOFDAY = %Dec(%XLate(' ':'0':
        FILDD):2:0);
//
  CUSTKY = CNAME;
  CUSNAM = CUSTNM;
//
  CLASSK = CLASS;
//
  COSFIR = ' ';
  COSBUS = ' ';
  COSECO = ' ';
  If CLASSN = '1';                                                 //
    COSFIR = 'X';
```

```
025200     C           INZSR     BEGSR                          * SET OUTPT SCREE
025300     C*
025400     C                     SETOF                        39 * UNPROT TIL F6
025500     C                     SETOF                    404450 * ERROR IND. SC 1
025600     C                     SETOF                    515253 * ERROR IND. SC 1
025700     C                     SETOF                    545556 * ERROR IND. SC 1
025800     C*
025900     C                     MOVE ' '       MSGID   7
026000     C                     MOVE *BLANKS   MSGDTA 80
026100     C                     MOVE @PGM      MSGQUE 10
026200     C*
026300     C                     MOVE @FALSE    @EXIT   1
026400     C                     MOVE @PGM      PGMQ
026500     C*
026600     C                     Z-ADDPORDER    ORDNUM                       ACTIVE
026700     C*
026800     C           ORDNUM    CHAINORDNAME               44
026900     C*
027000     C                     MOVE DEPART    FILDPK
027100     C                     MOVE FILYY     DOFYR
027200     C                     MOVE FILMM     DOFMON
027300     C                     MOVE FILDD     DOFDAY
027400     C*
027500     C                     MOVE CNAME     CUSTKY
027600     C                     MOVELCUSTNM    CUSNAM
027700     C*
027800     C                     MOVE CLASS     CLASSK
027900     C*
028000     C                     MOVE ' '       COSFIR
028100     C                     MOVE ' '       COSBUS
028200     C                     MOVE ' '       COSECO
028300     C           CLASSN    IFEQ '1'                         * RUN INTERACTIVE
028400     C                     MOVE 'X'       COSFIR
028500     C                     ENDIF
028600     C           CLASSN    IFEQ '2'                         * RUN INTERACTIVE
028700     C                     MOVE 'X'       COSBUS
028800     C                     ENDIF
028900     C           CLASSN    IFEQ '3'                         * RUN INTERACTIVE
029000     C                     MOVE 'X'       COSECO
029100     C                     ENDIF
029200     C*
029300     C                     Z-ADDTCKTNO    TCKETS                       ACTIVE
029400     C*
```

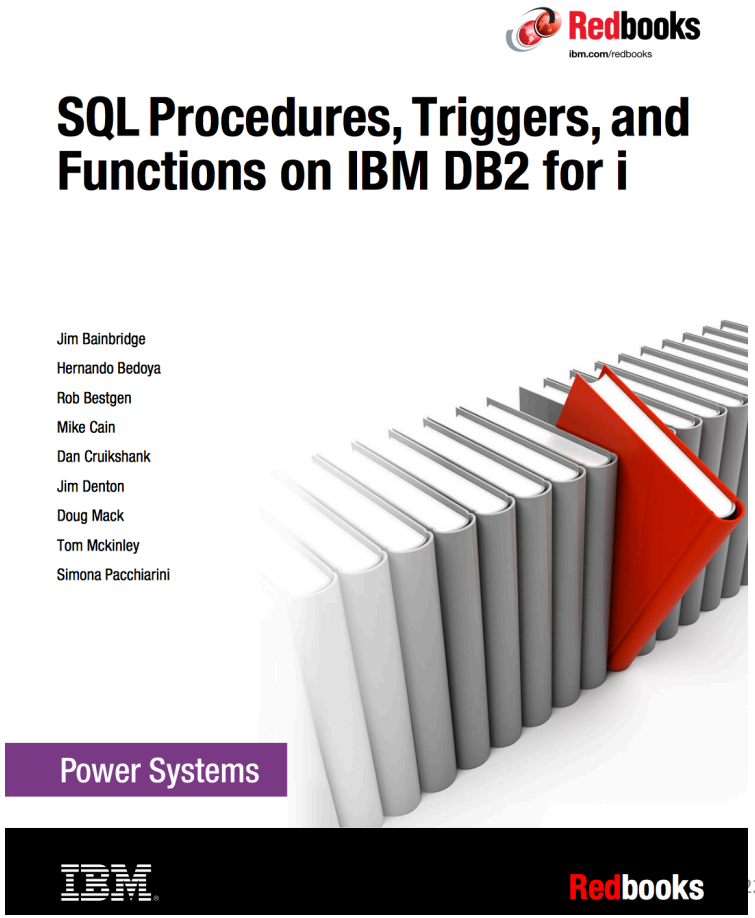
Latest in SQL

New in Apr 2016

Updates from the original

Many of the same expert Authors, plus a few New ones!

http://www.redbooks.ibm.com/abstracts/sg248326.html?Open



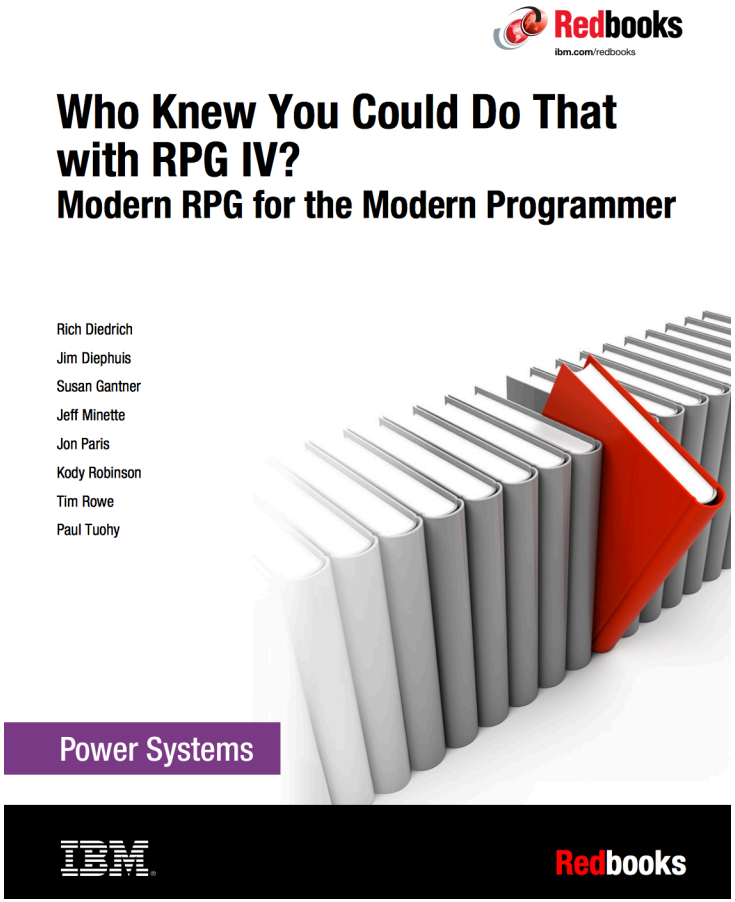
Modern RPG

New in Dec 2016

Updated from the original book from 2000

Many of the same expert Authors, plus a few New ones!

https://www.redbooks.ibm.com/redbooks/pdfs/sg245402.pdf



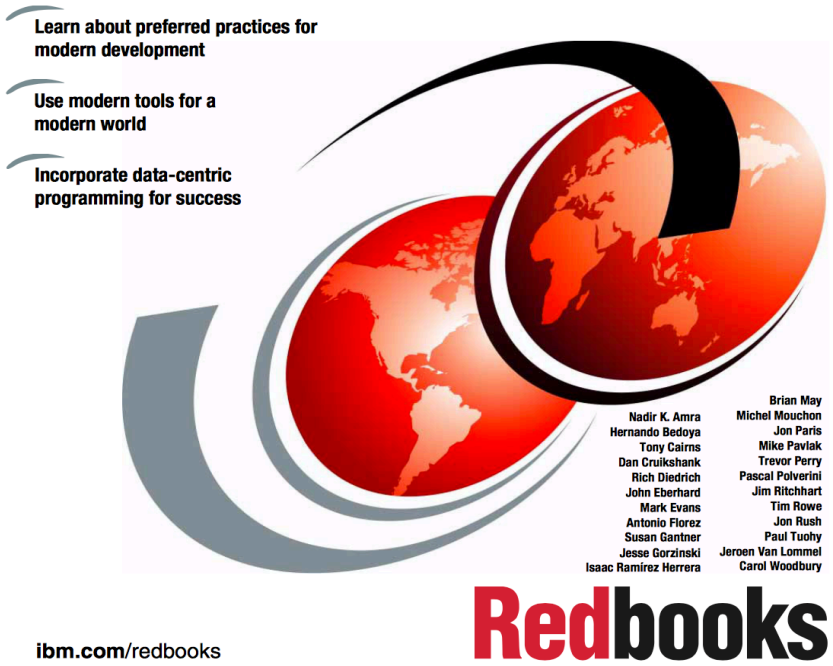
Modern IBM i Application from the Database up to the User Interface and Everything in Between

New in June 2014

Many Experts from the IBM i Community

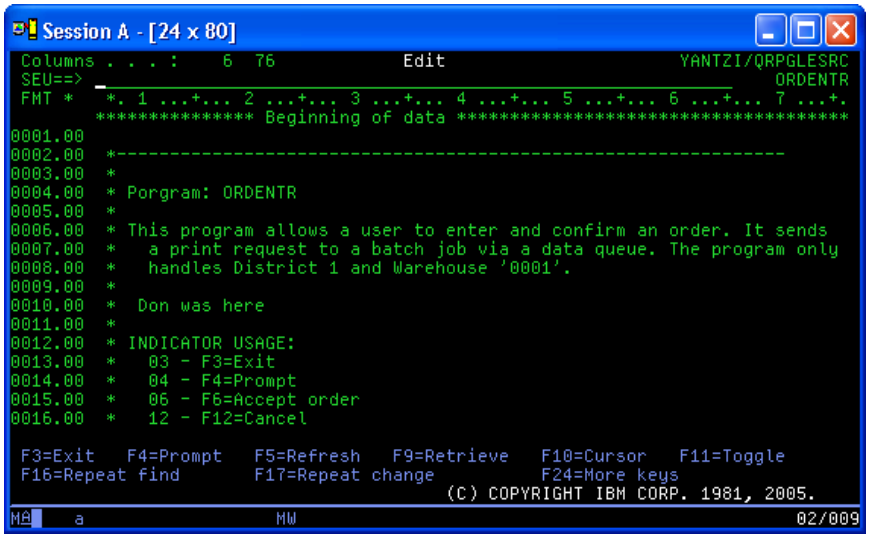
http://www.redbooks.ibm.com/redbooks/pdfs/sg248185.pdf

22,000 + Downloads!!!!



Development tools….

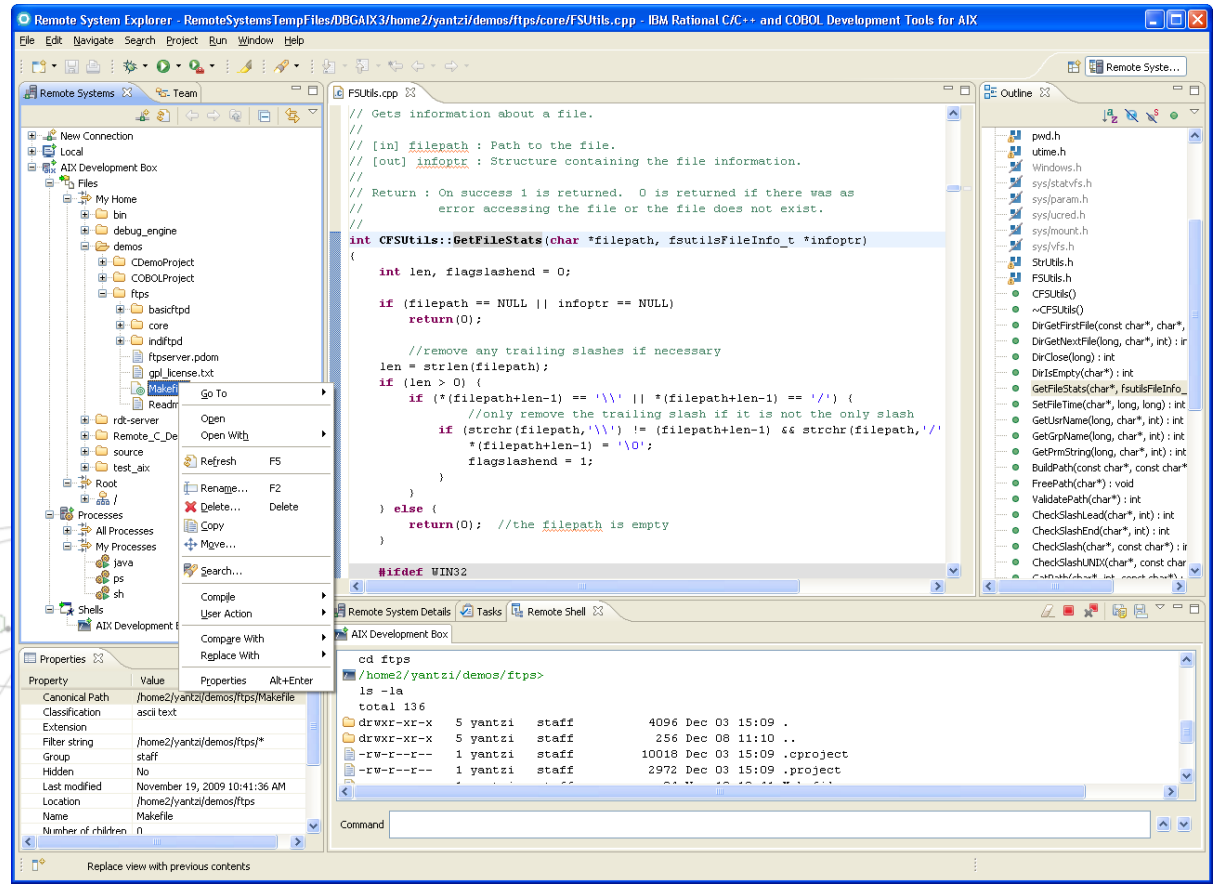
=



Tried and True, Yes but also SLOW, OLD, No new features

Rational Developer for i

- *Modern*
- *Integrated*
- Analysis
- Debugger
- Visual
- Supports RPG, COBOL, CL, C, C++, SQL, DDM



Developer Productivity

- Return on Investment
- Better able to respond to the needs of the business
- Higher quality
- Positioned to adopt new talent

20%→50% Productivity improvement

RDi Latest and Greatest 9.5.1

SSO

Kerbrose

Run SQL Scripts

Format SQL

Visual Explain

RPGILE Outline

O-Spec I-Spec

Indicators

Unreferenced

Synchronize IFS Projects

Comments

RDi Latest and Greatest 9.5.1.2

- Fix pack, so easy to install
- Nested Data Structures
- %MIN, %MAX
- Align(*FULL)
- Trim Leading spaces on line join ALT-J
- Kerberos Updates
- Content Assist updates
- Outline view updates

Rational Developer for i

RDi 9.5.1.2
Delivers Name Refactoring

Renaming

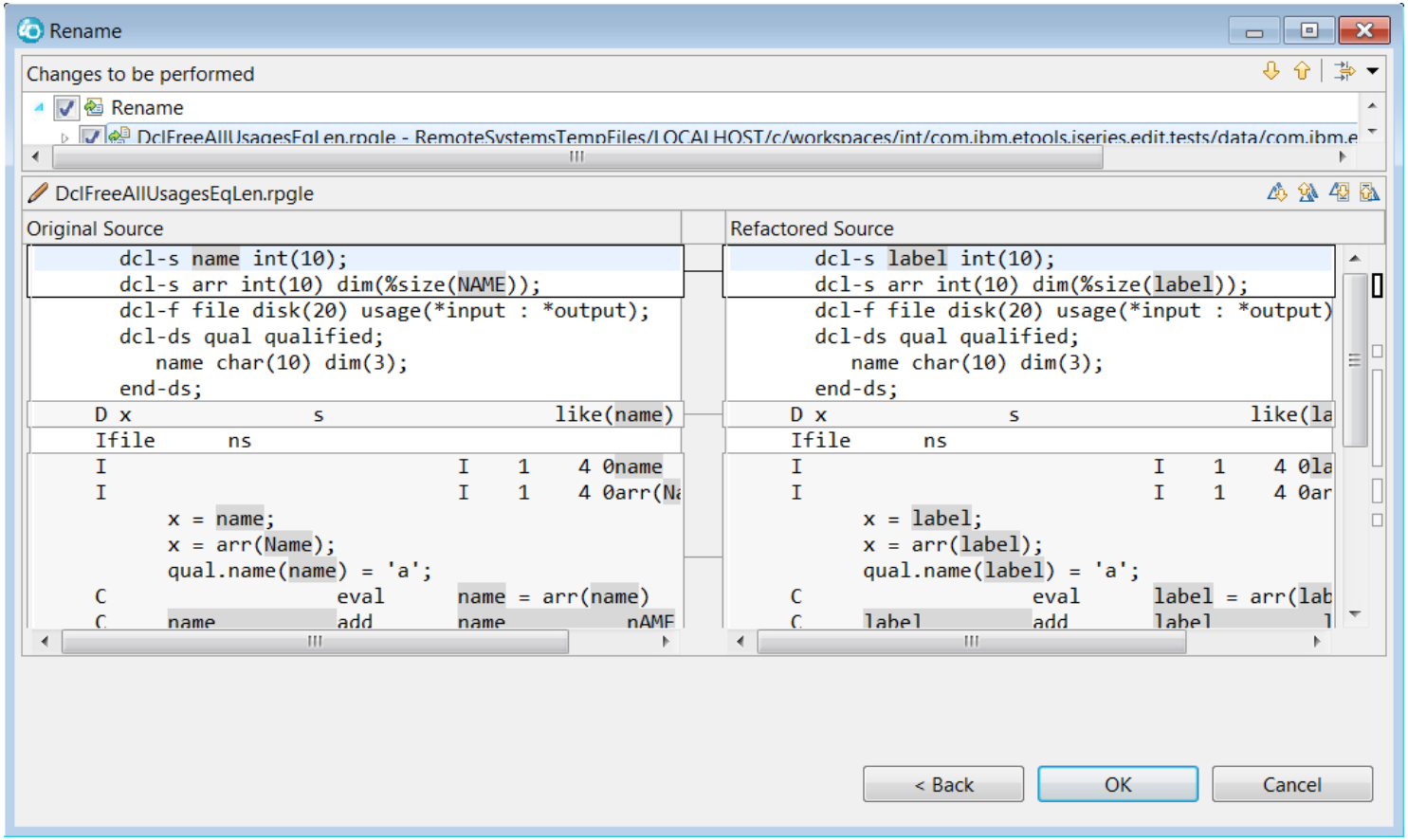
Any fool can write code a computer can understand…. Good programmers write code that a human can understand.

Martin Fowler

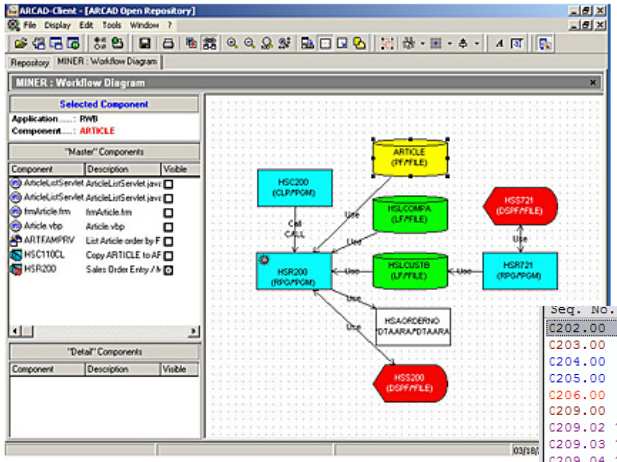
Names

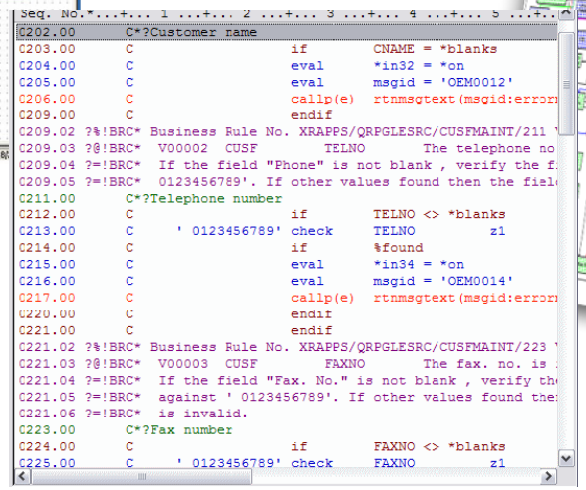
- Names are critical to
  - Revealing the programmer's intent
  - Provides meaning
  - Making the code read like prose
  - Maintainability
- Changing code without changing the function
- Since the tool makes renaming easy and error-free there is no excuse to tolerate cryptic or out-of-date names any more
- It works within fixed, free and fully free (**FREE) source
- Better then Find -> Replace

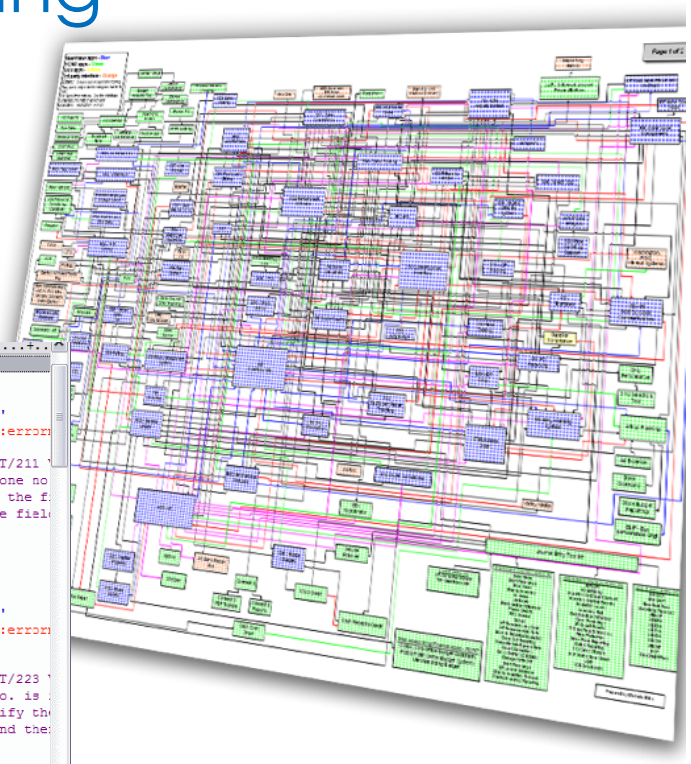
Rename Refactoring



Tools for Program Understanding







ARCAD – Observer

FRESCHE – X-Analysis

http://arcadsoftware.com/products/arcad-observer-application-analysis/

Why a Tool for Analysis?

- Rapid analysis for hot fixes
- Application Change Studies (cost estimation)
- Redesign/re-architecting/SOA
- Extraction of business rules
- Application modernization
- Skill transfer
  - Help new people learn the application
- Generation of documentation required by regulatory constraints

Provides the Blueprint

Time to start understanding your code

April 2017 Announce…

ARCAD Observer is now included in EES

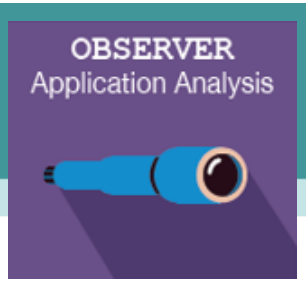
ARCAD Observer for i - 5733-AO1

- Plugs into RDi

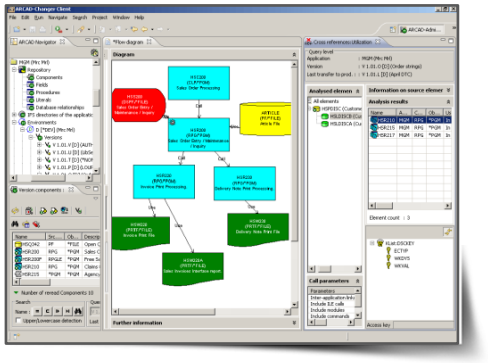
https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS217-151&appname=lenovous&language=en

ARCAD-Observer

Application analysis



Summarize application components in an easy to digest way



More than 50% of developers time is spent understanding application's

An immediate solution for missing or obsolete documentation

- Diagrams: I/O, database relationship, chain
  - Observe and analyze applications with diagrams
- Flowcharts
  - Observe and analyze applications with flowcharts
- Macroscopic views
  - Drill down to all dependencies between components
- Source code analyzer
  - Enhance source code graphically
- List manager
  - Analyze custom groups of components
- Documenter
  - Automatically generate diagrams, documentation & HTML
- Eclipse based RDi Plug-in
  - Accessible directly from RDi
- Ready for Rational
  - IBM validated integration with RDi

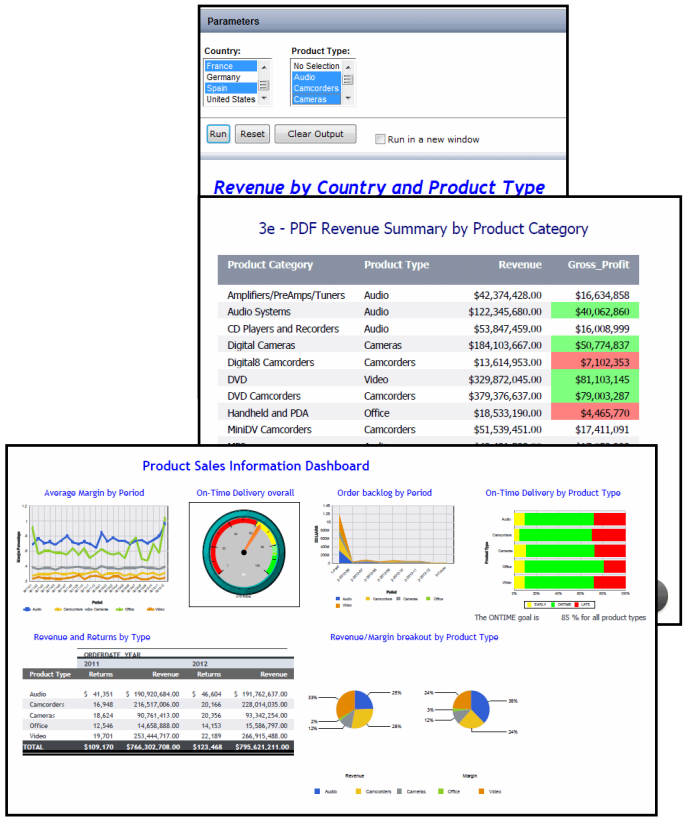
Data
&
Data Access

Why Modernize Your Reporting Environment

- Turn DATA into INTELLIGENCE
- Delivered at the right time, in the right formats
- Take advantage of SQL Capabilities
- Document the database, and standardize on data meaning (single version of the truth)

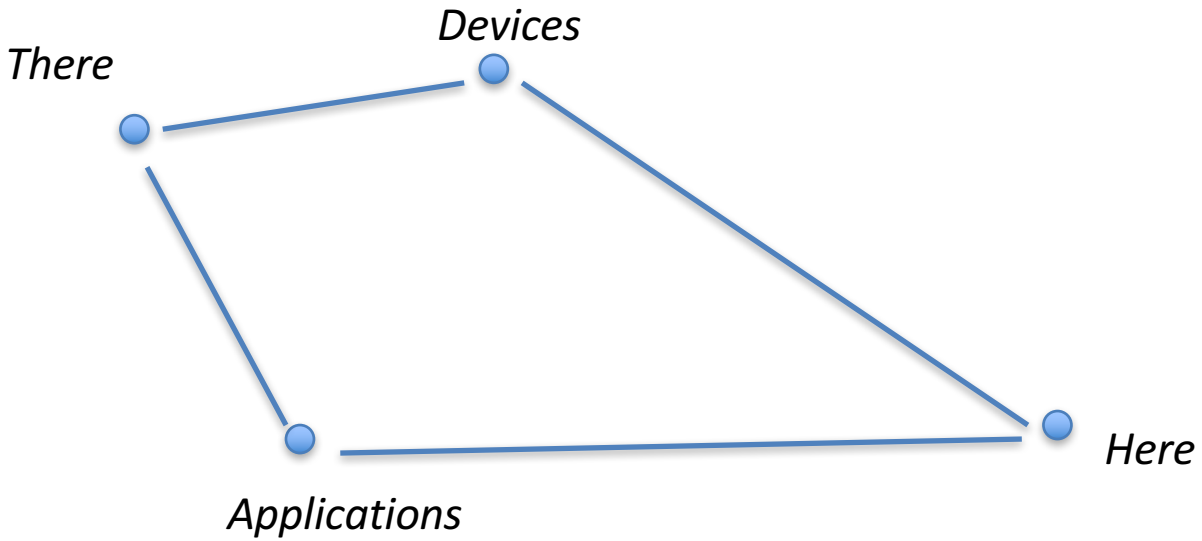
- Get I/T out of the Report Writing Business
  - With self service reporting solutions
- Eliminate redundancy
- Adapt more quickly to changes in the operational systems/databases
- Deploy reports in a variety of highly intuitive ways
  - Mobile, Excel, Dashboards, OLAP

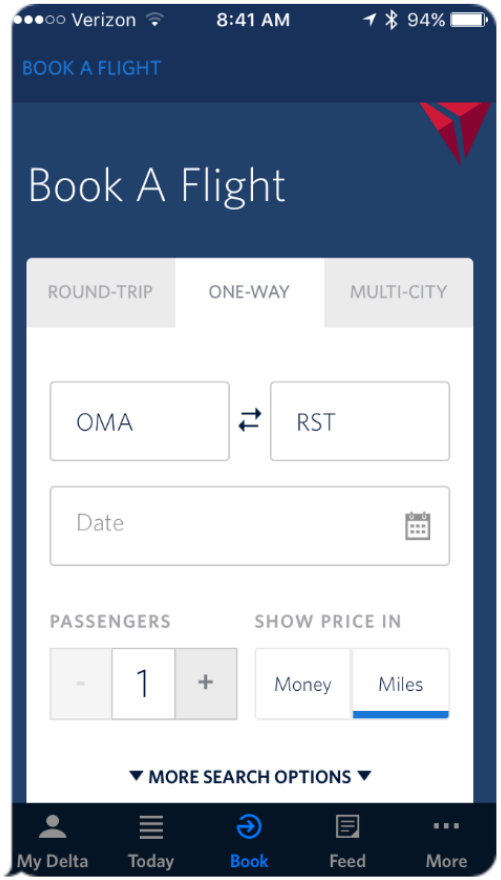
- Improve the perception of the “AS/400”

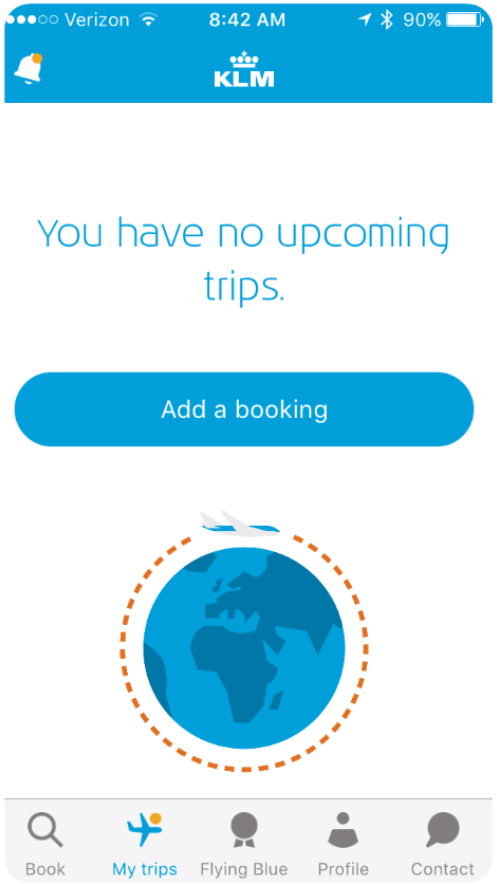


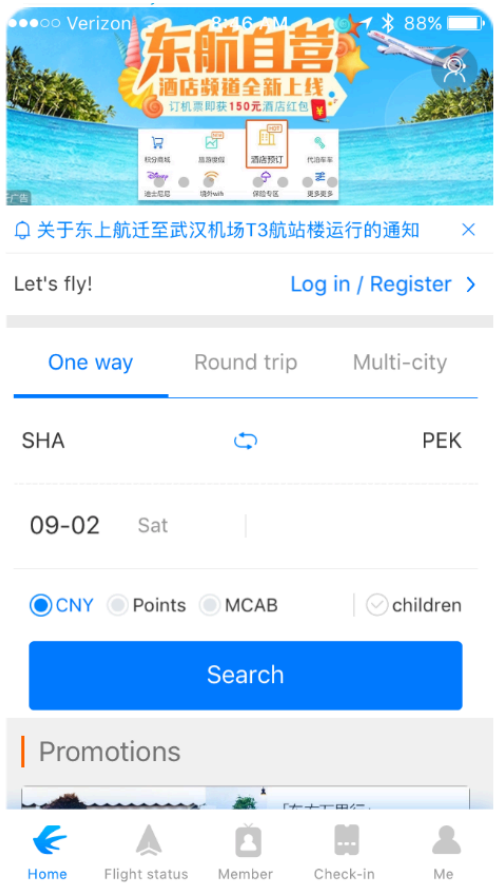
DB2 Web Query

Connections









API Definition

Application

Programming

Interface

API Definition

Application programming interface

From Wikipedia, the free encyclopedia

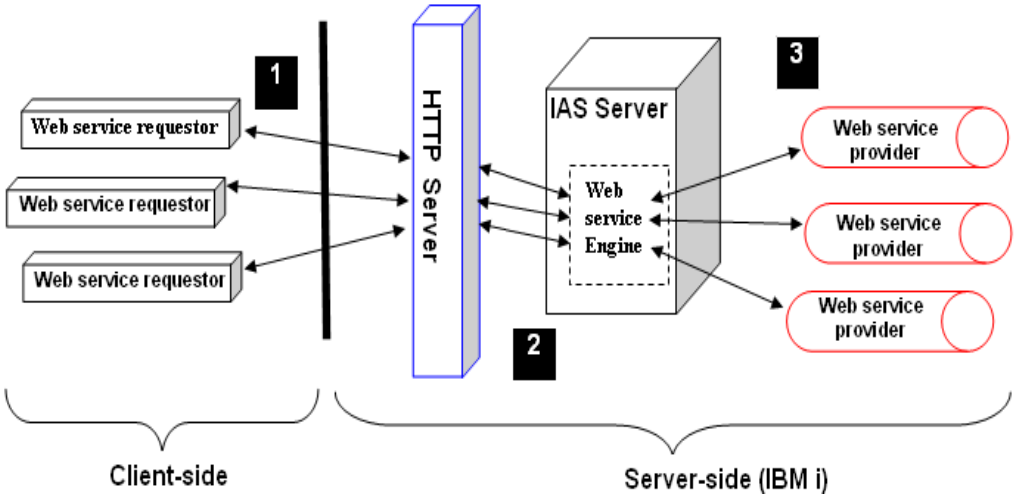
"API" redirects here. For other uses, see API (disambiguation).

In computer programming, an **Application Programming Interface** (**API**) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables or remote calls. POSIX, Microsoft Windows API, the C++ Standard Template Library and Java APIs are examples of different forms of APIs. Documentation for the API is usually provided to facilitate usage.

APIs - Simple

Simple way to connect endpoints. Send a request and receive a response.

Web service client/server flow does not change



- Create and Host APIs – IWS Server
  - SOAP Web Services (JAX-WS)
  - REST Web Services (JAX-RS)
- Call Web Services from your native RPG/COBOL. - IWS Client

http://www.ibm.com/systems/i/software/iws/

How do you access IBM i Information ?

Past

- Write code to call
  - CL Commands
  - System APIs
- Write more code to sort, filter, find what you are after

Today with SQL Services

- With a single SQL statement
  - Retrieve information
  - Leveraging the power of SQL
    - Sort, Filter, find what you are after

Built in DB2 Services – Use SQL to get info from the system!

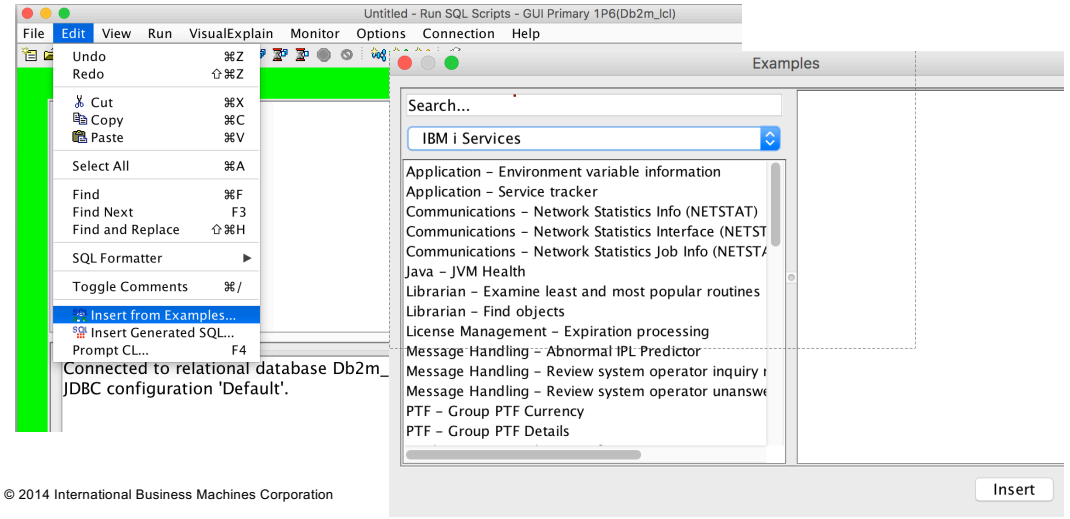
Who's password is about to expire

```
SELECT * from qsys2.user_info where status = '*ENABLED' AND
DATE_PASSWORD_EXPIRES BETWEEN CURRENT TIMESTAMP AND CURRENT TIMESTAMP + 7 DAYS;
```

SELECT * from qsys2.user_info where status = '*ENABLED' AND DATE_PASSWORD_EXPIRES BETWEE ... - X1423p2.rch.stglabs.ibm.com(X1423p2)

AUTHORIZATION_NAME	PREVIOUS_SIGNON	DAYS_UNTIL_PASSWORD_EXPIRES	DATE_PASSWORD_EXPIRES
APASSE	2014-05-01 12:03:5...	3	2014-05-05 00:00:00.0...
BOT5001USR	2013-10-31 00:49:2...	2	2014-05-04 00:00:00.0...
LJAMESON	2014-04-30 13:50:0...	3	2014-05-05 00:00:00.0...

Examples included in ACS



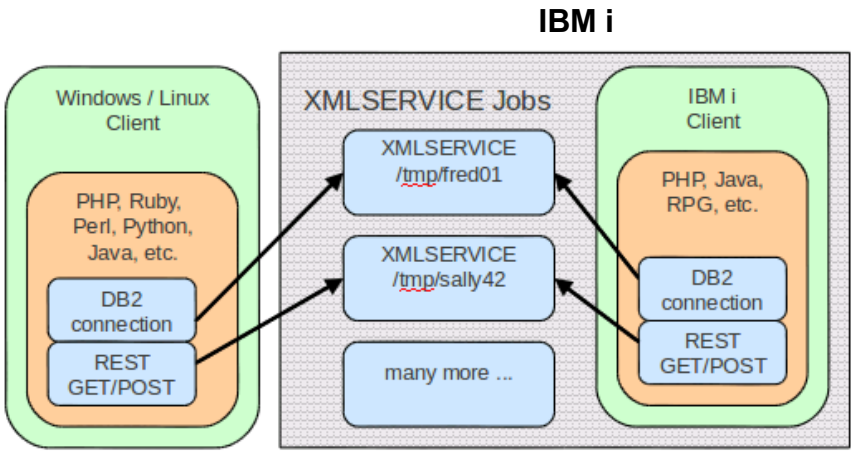
http://ibm.biz/DB2foriServices

XML Service

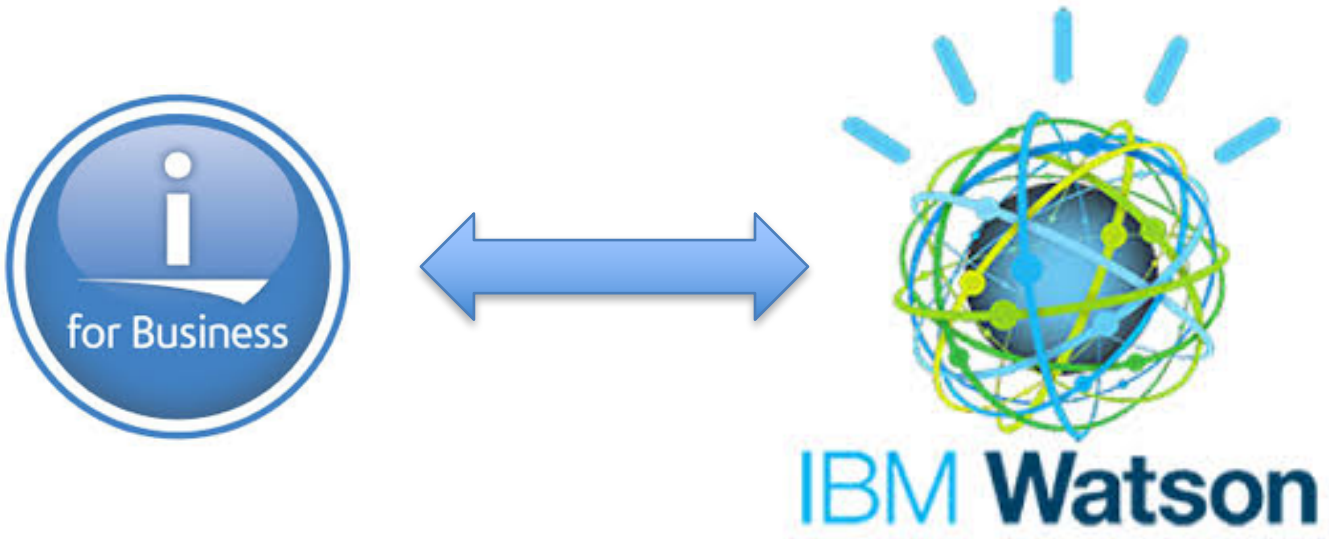
Easy access to IBM i data and resources from PHP

(similar to IBM ToolBox for Java)

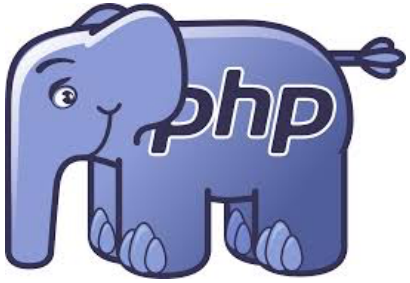
- –DB2 for i – SQL and Native
- –Program call
- –Procedure call
- –Data Area
- –Data Queue
- –Message Queue
- –Commands
- –System values
- –Spool files



http://www.youngiprofessionals.com/wiki/index.php/XMLSERVICE/XMLSERVICE



Some Key Open Source Technolgies



What are our Customers Doing with it ?

- New ways to leverage their data
- Modern UI
- Opening new Markets or expanding existing

https://scotsburn.com/

https://www.wengercorp.com/

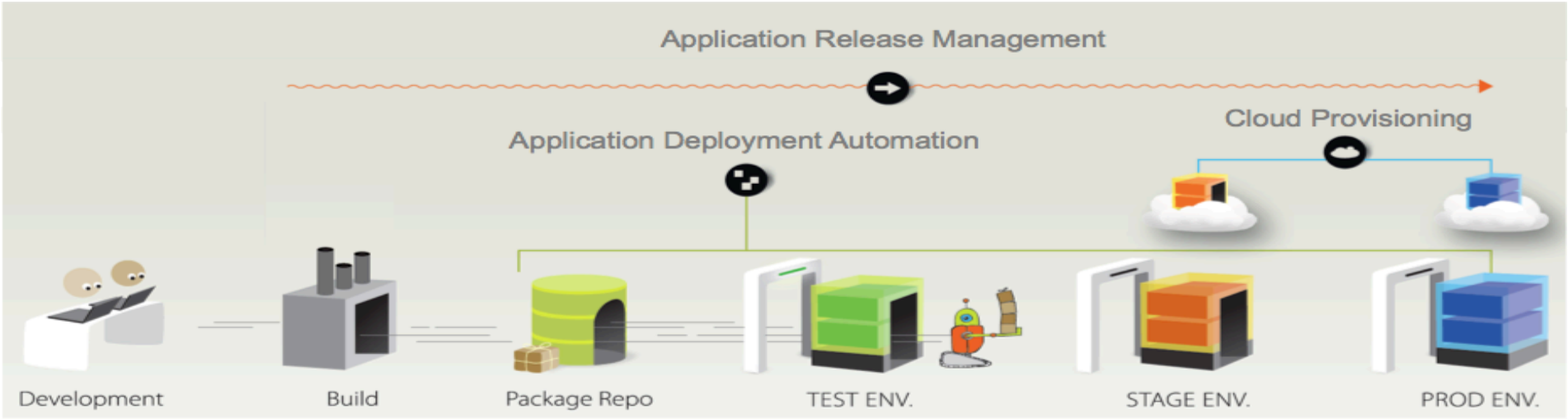
https://www.ksturnpike.com/

http://www.htbendix.dk/#

https://is.alliedrefrigeration.com/arhome.php

DevOps

Helping development meet the needs of the business at the speed of business



What is DevOps

Accelerate software delivery – *for faster time to value*

Balance speed, cost, quality and risk *for increased capacity to innovate*

Reduce time to customer feedback **–** *for improved customer experience*

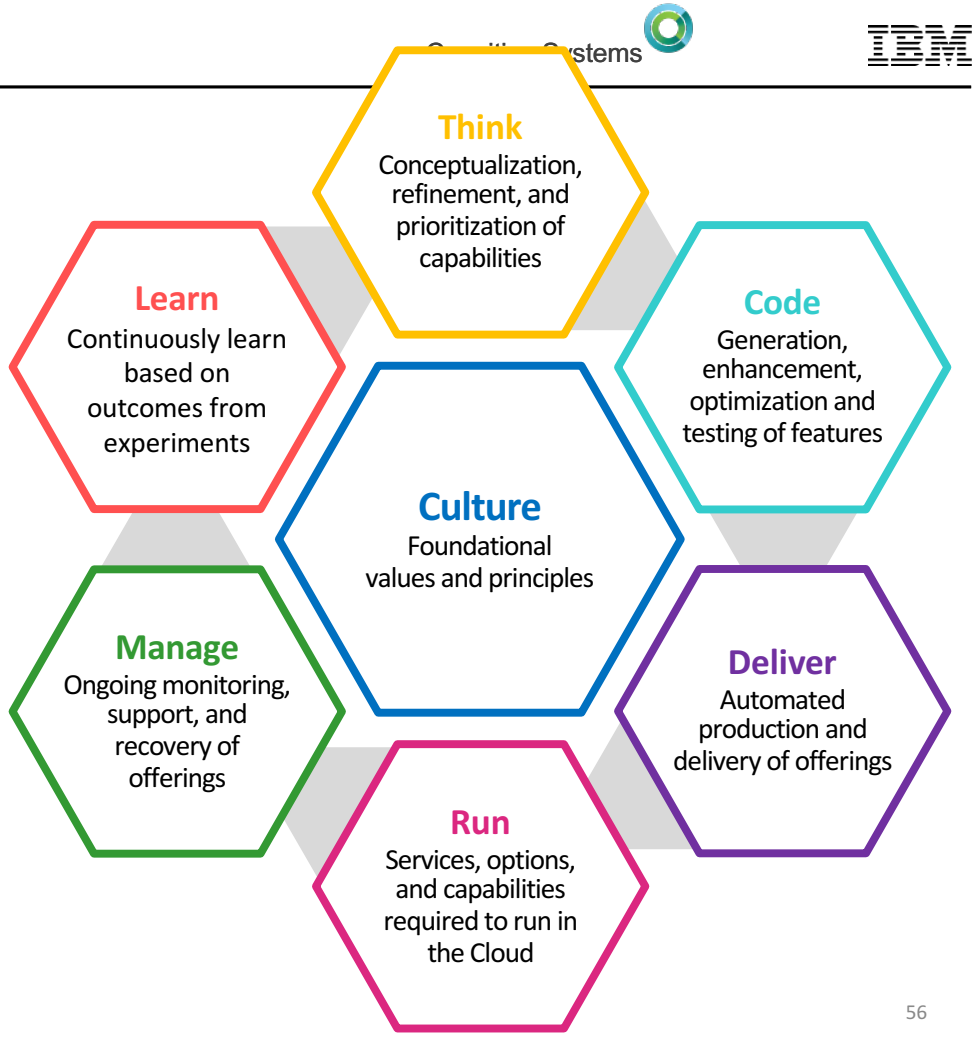
Process

Culture

Technology

DevOps

To learn more visit:
https://www.ibm.com/devops/method



The Systems is NOT Old….its what you have choosen to do with the system that defines its ‘Age’

Art of the Possible

IBM i 7.1 Concerns

- IBM i 7.1 has been out for 7 years
- Many parts are getting very long in the tooth….

SMB1

SSL/TLS

IBM i 7.1 Concerns

- Java
  - Default version of Java is Java 6
  - Just announced - Java 6 will reach EOL Dec 2017
    - No more fixes
    - No more security patches
    - Only option is to move to newer version of Java
  - Oracle ended Java 6 (for premium customers) Dec 2016
  - IBM went an additional year with support

IBM i 7.1 Concerns

- Apache
  - DG1 – HTTP Server Powered by Apache
    - Ships Apache 2.2
    - Apache Foundation announced EOL for 2.2 for Dec 2017
    - No more security patches available
      - This applies even if you purchase extended service
    - Only option – Move to new IBM i OS

IBM i 7.1 Concerns

SMB1

- SMB1
  - Recent US-CERT and Microsoft advisories recommend disabling SMB1
  - Internal MAD vulnerability scan failures based on the advisories
  - Microsoft plans to uninstall SMB1 in future builds of Windows 10

- This effects
  - QNTC
  - NetServer

- Resolution – Move to newer IBM i OS release

System SSL - New in 7.2 (PTFs back to 7.1)

- Transport Layer Security version 1.1 & 1.2 protocol (TLSv1.1 and TLSv1.2) RFC 4346 & RFC 5246
  - SHA2 support

WARNING: Payment Card Industry (PCI) will require TLS 1.1 or TLS 1.2 in June, 2018. IBM i 6.1 does not support TLS 1.1 or TLS 1.2.

- TLS1.2 is the current "gold" standard
  - PCI requirement in June 2018 (TLS 1.1 or TLS 1.2)
  - TLS 1.1 & 1.2 support exists in 7.1 and 7.2
  - IBM i 6.1 and earlier will not meet PCI requirements in 2018
  - Latest support has been PTFed to 7.1

NOTE: All versions of SSL and TLS 1.0 have been deemed unsecure and should never be used!

Questions and Answers

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Revised September 26, 2006

Special notices (cont.)

Revised December 2, 2010