



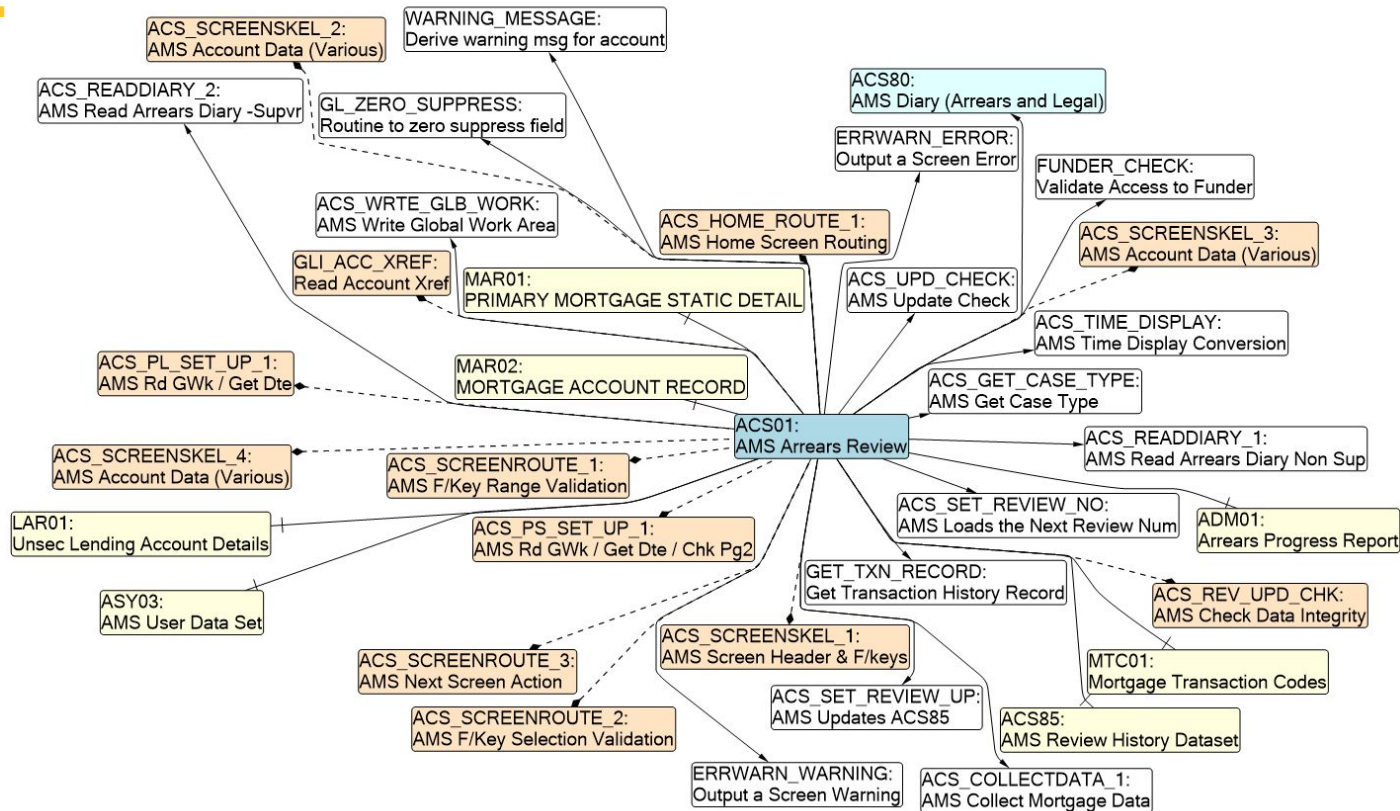
Why Is Testing Slowing Delivery?

Nigel.Tunncliffe@Unisys.com

Why testing can be so difficult

- EAE and AB Suite makes it easier to build and maintain complex applications
 - Often much larger than with COBOL, C#, Java etc
- The user executed untested code
 - Typically large number of paths through a unit of code
- The order of statements executed is different to what was expected
 - The user did something the developer did not expect
- The user applied a combination of un-tested values
 - The user entered data the developer did not expect
- Developers and Users think differently
- A successful test is one that finds a bug

Typical AB Suite Ispec





Measuring Code Complexity

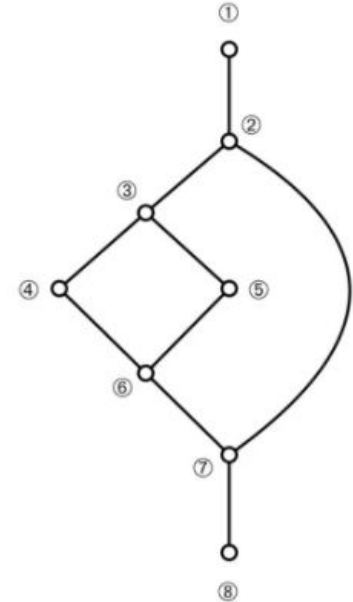
Measuring Complexity

- Function of number of paths through code
- More paths equals more test cases
- Things that make testing complex
 - Paths
 - Readability
 - Inputs/Outputs
 - Volume of logic
- Popular Measurements
 - Industry has a number of popular measurements

Cyclometric Complexity

- Indicates the number of possible paths through a unit of logic
 - Is a measure of testing effort
 - Ideally should be less than 10
- Based on 1976 Paper from Thomas McCabe
 - Determines the 'Nodes' and 'Edges' in a graph of the logic
 - Complexity = Edges – Nodes + 2
 - $Complexity = 9 - 8 + 2 = 3$.

```
① start
② if (X) then
③   if (Y) then
④     perform A
        perform B
        else
⑤     perform C
        perform D
⑥   endif
⑦ endif
⑧ end
```



[Cyclomatic complexity - Wikipedia](#)

Cognitive Complexity

- Measures the ‘readability’ of logic to determine how easy it is for another developer to understand
 - Ignore structures that allow multiple statements to be readably shorthanded into one
 - Increment (add one) for each break in the linear flow of the code
 - Increment when flow-breaking structures are nested
- The higher the number the less easy to read
- Simple code is easier to read and update
 - Less chance of introducing errors

Cyclometric vs Cognitive

- Both values have an impact on testability
- Each snippet of code shown has a Cyclometric Complexity of 4
- Second snippet is easier to read
 - First snippet has Cognitive Complexity of 7
 - Second snippet has Cognitive Complexity of 1

```
int sumOfPrimes(int max) {  
    int total = 0;  
    OUT: for (int i = 1; i <= max; ++i) {  
        for (int j = 2; j < i; ++j) {  
            if (i % j == 0) {  
                continue OUT;  
            }  
        }  
        total += i;  
    }  
    return total;  
}
```

```
String getWords(int number) {  
    switch (number) {  
        case 1:  
            return "one";  
        case 2:  
            return "a couple";  
        default:  
            return "lots";  
    }  
}
```


Halstead Complexity – Measure of Cognitive Complexity

Parameter	Meaning
n1	Number of unique operators
n2	Number of unique operands
N1	Number of total occurrence of operators
N2	Number of total occurrence of operands

Metric	Meaning	Mathematical Representation
n	Vocabulary	
N	Size	
V	Volume	
D	Difficulty	
E	Effort	
B	Errors	
T	Testing time	

[Halstead complexity measures - Wikipedia](#)

Visual Studio 'Maintainability' Index

- Maintainability Index formula
 - $\text{MAX}(0, (171 - 5.2 * \ln(\text{Halstead Volume}) - 0.23 * (\text{Cyclomatic Complexity}) - 16.2 * \ln(\text{Lines of Code})) * 100 / 171)$
- Returns a number between 0 and 100
 - Higher numbers equal easier to maintain
 - 0-9 = Red
 - 10-19 = Yellow
 - 20-100 = Green
- Microsoft Visual Studio only calculates maintainability for Microsoft languages
 - C#, VB.Net, F#
 - [Calculate code metrics - Visual Studio \(Windows\) | Microsoft Docs](#)

Component Coupling

- Measures
 - How many components are used by a particular component
 - How many other components use a particular component
- High coupling is bad as
 - A change in the component may break a referenced component
 - A change in a referenced component may break the component

Function Point Analysis

- Developed by Allan Albrecht from IBM in 1979
 - Measures the functionality of the system that is meaningful to users
 - Inputs
 - Outputs
 - Inquiries
 - Internal Files
 - Interfaces
- Creates a raw number which be used to determine the effort need to create new functionality
 - Function Points to development effort conversion is site specific
- Matrix can calculate Function Point Count for whole system or any folder

Function Point Analysis

	A	B	C	D	E	F
1	Function Point Summary for PERFORMDB generated on 29 July 2021 at 15:48					
2						
3		Data Communications	Average Influence ▼	3		
4		Distributed Data	Average Influence ▼	3		
5		Performance Objectives	Significant Influence ▼	4		
6		Tight Configuration	Average Influence ▼	3		
7		High Transaction Rate	Moderate Influence ▼	2		
8		On-line inquiry/data entry	Average Influence ▼	3		
9		End User Efficiency	Significant Influence ▼	4		
10		On-line Update	Significant Influence ▼	4		
11		Complex Processing	Moderate Influence ▼	2		
12		Code Re-usability	Average Influence ▼	3		
13		Conversion/Installation Ease	Average Influence ▼	3		
14		Multiple Site Installation	Moderate Influence ▼	2		
15		Facilitate Change	Significant Influence throughout ▼	5		
16						
17		Total Complexity		41		
18		Adjustment Factor	$0.65 + (\text{Total Complexity} * 0.01) =$	1.06		
19						
20		Total Unadjusted Function Points		1070		
21						
22		Adjusted Function Points		1134		
23						
24						
25						
26						
27						
28						



Identifying Complexity

Measuring Complexity in Practice

- Described the approaches in theory
- How do we do this in practice?



Model Analysis Tool for Research Investigation and XREF

Complexity Summary

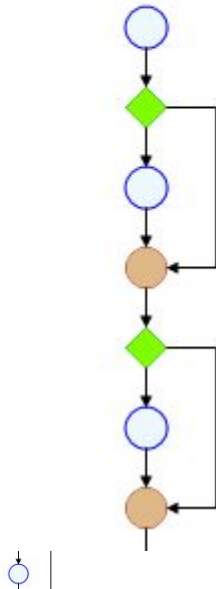
Cyclometric Complexity = 13

Cognitive Complexity = 18

Maintainability = 39

Predicate Complexity = 13

Show Predicates



```
51  MOVE UC_1      UC_READMAINFILED  
52  
53  GLP_CREATE_LET29 ()
```

System Complexity

- Matrix can calculate complexity for all methods in the application
- Can be run at periodic intervals to check complexity is decreasing
- Application size may increase but complexity should decrease over time

1	Name	Complexity	Maintainability
50	CommonLibrary.WFLNodeName	2	100
51	CommonLibTest.CreateVariableLengthFile	6	100
52	CommonLibTest.DecodeFileCheck	6	100
53	CommonLibTest.DecodeReportStatus	3	100
54	CommonLibTest.Main	27	25
55	CryptLib.CryptString	2	100
56	CryptLib.decryptstring	2	100
57	CryptLib.encryptstring	2	100
58	CUSTSQL.Construct	1	100
59	CUSTSQL.Destruct	1	100
60	CUSTSQL.Main	1	100
61	DBReadTest.DTTAMISP	84	9
62	DBReadTest.Main	2	17
63	DBReadTest.MultiStructureRead	19	34
64	DBReadTest.SingleStructureRead	5	100
65	DefaultExe.Call	2	100
66	DefaultLibrary.Call	2	100

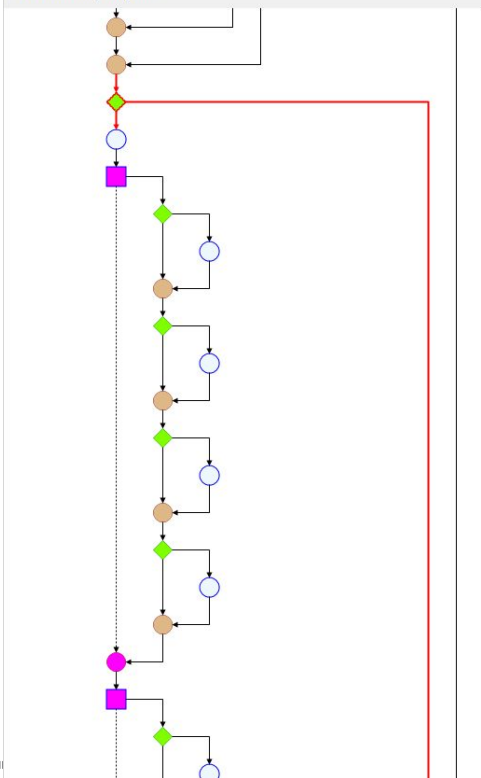
Component Coupling

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Method	Description	References	Referenced By	Reference List	Referenced By List								
2	AAD01	Alternative Audit Description	1	0	AAD02									
3	AAD02	Alternative Audit Descriptions	0	2		AAD01, AAD03								
4	AAD03	Alternative Audit Desc Summary	2	0	AAD02, DIG02									
5	ACC01	Account Xref	2	31	ACC01, ACC04	ACS05, API04, ASY81, CRC01, CTT62, CTT66, CTT84, CTT97, CUS05, CUS13, CUS83, CUS92, DDS07, INS60, LCO50, LET27, LET72, LET75, MAP01, MPL03, MPL04, MTG50, NOM28, SCR20, SEC01, SEC13, SRV64, SV025, SV036, TRI10, TRI65								
6	ACC02	Account Sequence Numbers	0	6		ACC03, IPR02, LAP01, MAP01, PAR61, TRI65								
7	ACC03	Maint Account Number Sequences	3	0	ACC02, LCO26, PAR02									
8	ACC04	Internal Accounts Available	0	0										
9	ACC05	Next Available Plan ID	0	0										
10	ACC99	A/C Number and Product Code	1	0	MPR01									
11	ACS01	AMS Arrears Review	8	0	ACS80, ACS85, ADM01, ASY03, LAR01, MAR01, MAR02, MTC01									
12	ACS02	AMS Telephone Screen	4	0	CUS74, LAR01, MAR01, PAR16									
13	ACS03	AMS Telephone Result	1	0	MAR01									
14	ACS04	AMS End Telephone Call	2	0	ACS75, MAR01									
15	ACS05	AMS Set Details for Nxt Review	13	0	ACC01, ACS70, ACS80, ACS85, ACS95, ADM54, ALE85, ASY03, ASY05, ASY60, ASY84, LPAB4, MAR01									
16	ACS09	AMS Arrears Reason Index	2	0	ACS80, ASY01									
17	ACS10	AMS Case Inquiry	2	0	LAR01, MAR01									
18	ACS12	AMS Ability to Pay	3	1	ACS12, MAR01, PAR16	ACS12								
19	ACS14	AMS Address Dataset	5	5	ACS14, CUS01, LAR01, MAR01, MAR02	ACS14, ACS15, ACS50, ACS51, ACS55								
20	ACS15	AMS Address Dataset Inquiry	1	0	ACS14									
21	ACS20	AMS Arrangement Update	25	0	ACS70, ACS77, ADM20, ADM32, ADM38, ADM51, ADM54, ASY03, ASY60, BAF11, CUS01, DDS01, DDS02, FLY98, LAR01, LCO26, LET28, MAR01, MAR02, MAR11, MTC01, PAD02, PAR16, SEC06, SEC09									
22	ACS22	AMS Primary/Secondary Accounts	7	0	ADM20, ADM32, ADM54, MAR01, MAR02, MPR01, PAR02									
23	ACS23	Maintain PTP Arrangement	16	0	ACS70, ACS77, ADM20, ADM51, ADM54, BAF11, DDS01, DDS02, LCO26, LET28, MAR01, MAR02, MAR11, PAD02, PAR02, PAR16									
24	ACS24	Promise to Pay Arrngmnt History	2	0	ADM54, PAR16									

Large Methods equals High Complexity

Cyclometric Complexity = 737
Cognitive Complexity = 4628
Maintainability = J
Predicate Complexity = 755

Show Predicates



```
441
442
443 Move GC_5      GG_CET_IN.GP_CET_LCAT
444 DoWhen (GW_WORK.GW_SUBACCNO.GW_INT_ACCT <> MAR11.PRIMEACNO) :CD41
445     Determine Every MAR11.MAR11_001 (GW_WORK.GW_SUBACCNO.GW_INT_F
446         Break
447     End
448 End
449 Move MAR11.CURRENTBAL      GG_CET_IN.GP_CET_TVAL
450 CREATE_EOE_TRIG()          :CD4128262 A
451 End
452 DoWhen (MPR_TYPE <> MAR02.MPR_TYPE)
453     Move GW_WORK.GW_SUBACCNO.GW_INT_ACCT      GG_CET_IN.GP_CET_ACCT
454     Move GW_WORK.GW_DATEREL      GG_CET_IN.GP_CET_DATE
455     Move GC_9      GG_CET_IN.GP_CET_LCAT
456     DoWhen (GW_WORK.GW_SUBACCNO.GW_INT_ACCT <> MAR11.PRIMEACNO) :CD41
457         Determine Every MAR11.MAR11_001 (GW_WORK.GW_SUBACCNO.GW_INT_F
458             Break
459         End
460     End
461     Move MAR11.CURRENTBAL      GG_CET_IN.GP_CET_TVAL
462     CREATE_EOE_TRIG()          :CD4128262 A
463 End
464 End
465
466 :::::End of first live sub account logic moved for CD4101290 A
467
468 :Product and cap frequency change      :CD4101290 A
469 DoWhen ((MPR01.MPR_CODE <> SW_WORK.SW_HOLD.SW_ORIG_MPR) AND \:CD4101290 A
470 (MPR01.CAP_FREQ <> SD_CAP_FREQ2)) :CD4101290 A
471 :Set old and new cap points
472 :Start of cap point setting logic moved for :CD4101290 A
473 BeginCase SD_CAP_FREQ2
474     Case GC_M
475         Move LCO26.FEE_CH_MTH      SD_OLD_CAP
476     Case GC_Q
477         Move LCO26.FEE_CH_QTY      SD_OLD_CAP
478     Case GC_H
479         Move LCO26.FEE_CH_HYR      SD_OLD_CAP
480     Case GC_A
481         Move LCO26.FEE_CH_ANN      SD_OLD_CAP
482 EndCase
483
484 BeginCase MPR01.CAP_FREQ
485     Case GC_M
486         Move LCO26.FEE_CH_MTH      SD_NEW_CAP
487     Case GC_Q
488         Move LCO26.FEE_CH_QTY      SD_NEW_CAP
489     Case GC_H
490         Move LCO26.FEE_CH_HYR      SD_NEW_CAP
491     Case GC_A
492         Move LCO26.FEE_CH_ANN      SD_NEW_CAP
493 EndCase
494
495 :::::End of cap point setting logic moved for :CD4101290 A
496
497 DoWhen ((PAR02.FEE_SUBACC = "N") OR \:CD4101290 A
```



Repetitions Adds Complexity

Programming Mantra

- If you write the same code twice you are doing it wrong
- Make it so simple it can't go wrong
- So how can Matrix help?

Identify Common Painted Fields

- Many Ispecs have the same group of fields defined on the painter
 - Address
 - Name
 - Sort code /Account number
 - Start/End Date
- In EAE had to paint each group of fields separately
- AB Suite allows a class to have an associated painter definition
 - Define common painter definition and logic once
 - Add to Ispec definition
 - Simplifies testing

Common Fields

Members Data Items Summary Methods Referenced Referenced By Updated By Read By Database Reads Updates Profiles Documentation GSD's Used Folders Fixed Screen Common Fields Graphical Screen Member Of Dynamic Search TFS Check Outs Audit

Team: **TEAMID** User: **USER** Next: **AMS NX**

SYSNAME: **A** **ADDITIONAL ADDRESSES** **DATE DATEAT**

Acct No: **BOKIDN** **EXT_DISP: AsString [28]** **POSTALNAME: AsString [40]**

Address Number: **SEQ**

Postal Name: **AMS_PNAME: AsString [30]**

Building Name: **BUILD_NAME: AsString [40]**

Number/ street: **BUILD_N** **STREET: AsString [40]**

District: **DISTRICT: AsString [40]**

Town: **TOWN: AsString [30]**

County: **COUNTY: AsString [25]**

Post Code: **POSTCODE: A**

Tel. Number: **TELEPHONE: AsString [30]** **TPS: T** Ext. **EXTENS: I**

Short Description: **SHORT_DESC**

Description: **DESC1: AsString [25]**
DESC2: AsString [25]

Use for Customer 1 **DE** Customer 2 **DE** Customer 3 **DE** Customer 4 **DE**

F11 F1_DISPLAY: AsString [35] F95 F5_DISPLAY: AsString [30]

F22 F2_DISPLAY: AsString [35] F66 F6_DISPLAY: AsString [30]

F33 F3_DISPLAY: AsString [35] F77 F7_DISPLAY: AsString [30]

F44 F4_DISPLAY: AsString [35] F88 F8_DISPLAY: AsString [30] F

Ispec	Description	Common Fields
ACS14	AMS Address Dataset	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ACS30	AMS Incoming Calls/Letters	COUNTY, DISTRICT, TOWN
ADM01	Areas Progress Report	COUNTY, DISTRICT, TOWN
ADM06	Summary of areas letters	COUNTY, DISTRICT, TOWN
ADM08	Summary of assurance policies	COUNTY, DISTRICT, TOWN
ADM10	Redemption notifications	COUNTY, DISTRICT, TOWN
ADM18	LIFE & NON-LIFE INSURANCE COM.	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ADM30	Areas Admin & Receivership	COUNTY, DISTRICT, TOWN
ADM53	History of Not of Post Jud Int	COUNTY, DISTRICT, TOWN
ADM70	Unsecured Areas Progress Rep	COUNTY, DISTRICT, STREET, TOWN
ALC01	COAST Interface	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ALE30	AMS Adhoc Case Review	COUNTY, DISTRICT, TOWN
ALM03	EMAT Application Enquiry	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ALM05	MAES Loan Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
AML01	ID and Address Verification	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ASY81	AMS Batch Requests	COUNTY, DISTRICT, TOWN
BAF01	Branch Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF07	Agency Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF20	Brand Details Maintenance	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF24	Alternative Brand Name & Addr	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BNK01	Bank account details	COUNTY, DISTRICT, TOWN
BNK14	Bank branch details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CRD01	CARD PAYMENTS	COUNTY, DISTRICT, STREET, TOWN
CTT18	Register/Cancel R85	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CTT46	Mortgage Inquiry	BUILD_NUM, COUNTY, STREET, TOWN
CUS01	Customer details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS06	CUSTOMER BY CORRES. ADDRESS	BUILD_NAME, BUILD_NUM
CUS12	Ad-Hoc Customer Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS15	Business Customer Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS20	CUSTOMER DETAILS	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS21	Mortgage Customer Details II	TOWN
CUS22	Applicant Related Addr Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS24	AVAILABLE REFERENCE ADDRESSES	TOWN
CUS35	Mortgage Customer Details III	TOWN
CUS64	Customer Merge Preview	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS83	CUSTOMER IVA/BANKRUPTCY DETAIL	COUNTY, DISTRICT, TOWN
CUS91	Maintain Cust FSCS Big Data	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
DBT02	Debt Agency Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
IAD06	Inv. A/C Correspondence Add.	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
IFA06	CHEQUEBOOK REQUESTS	DISTRICT, TOWN
IFA09	MAINTENANCE OF CARD	DISTRICT, TOWN
INQ01	Check digit calculation	COUNTY, DISTRICT, TOWN
INQ05	HLC CALCULATION	DISTRICT
LAP02	underwriting analysis/authoris	COUNTY, DISTRICT, TOWN
LAP03	LOAN OFFER	COUNTY, DISTRICT, TOWN
LAP05	LOAN CANCELLATION	COUNTY, DISTRICT, TOWN
LCO10	Unsecured Funding Cpry Details	COUNTY, DISTRICT, TOWN
LCO90	Funding Company detail - Part 1	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
LET46	Letter Reference Organisations	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
LON20	Unsecured Loan Settlement	BUILD_NAME, STREET

Common Fields

Members Data Items Summary Methods Referenced Referenced By Updated By Read By Database Reads Updates Profiles Documentation GSD's Used Folders Fixed Screen Common Fields Graphical Screen Member Of Dynamic Search TFS Check Outs Audt

Team: TEAMID User: USER Next AMS_NX

SYSNAME: A ADDITIONAL ADDRESSES DATE DATEAT

Acct No: BOKIDN EXT_DISP: AsString POSTALNAME: AsString [40]

DISP_LIT28: AsString [28]

Address Number SEQ

Postal Name AMS_PNAME: AsString [30]

Building Name BUILD_NAME: AsString [40]

Number/ street BUILD_N STREET: AsString [40]

District DISTRICT: AsString [40]

Town TOWN: AsString [30]

County COUNTY: AsString [25]

Post Code POSTCODE: A

Tel. Number TELEPHONE: AsString [30] Ext. EXTENS

Short Description : SHORT_DESC

Description DESC1: AsString [25]

DESC2: AsString [25]

Use for Customer 1 DE Customer 2 DE Customer 3 DE Customer 4 DE

F1 F1_DISPLAY: AsString [35] F5 F5_DISPLAY: AsString [30]

F2 F2_DISPLAY: AsString [35] F6 F6_DISPLAY: AsString [30]

Next NEXT_SC

BANK BRANCH DETAILS

Sort Code SORT_COD BACS_STQ: AsString [40]

Title Code TITLE

BANK_TITLE: AsString [78]

Branch Name TITLE_1: AsString [30]

(contd) TITLE_2: AsString [30]

Manager MANAGER: AsString [40]

Building Name BUILD_NAME: AsString [40] House Number BUILD_NO

Street STREET: AsString [40]

District DISTRICT: AsString [40]

Town TOWN: AsString [30]

County COUNTY: AsString [25]

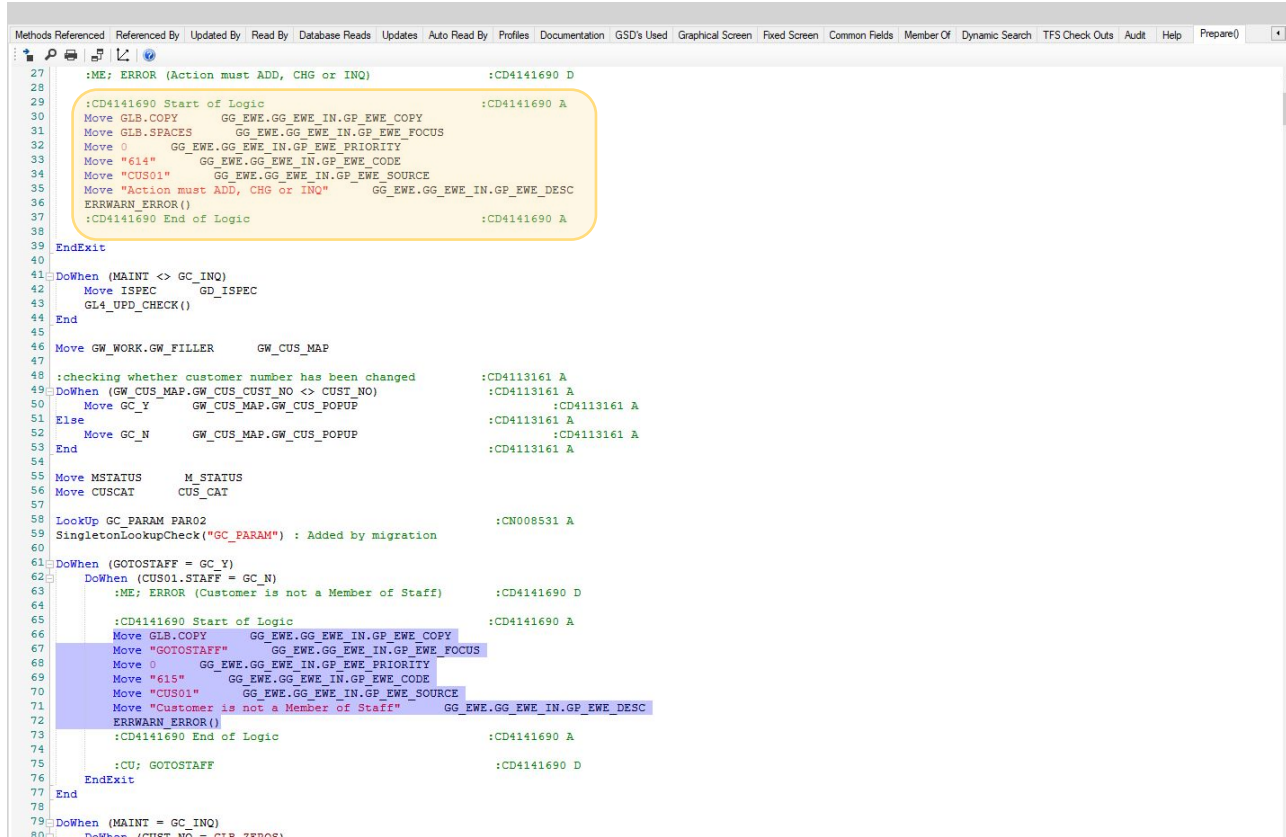
Postcode POSTCODE: A

Telephone TELEPHONE: AsString [30] Fax Number FAX_NO: AsString

BACS Exemption

Ispec	Description	Common Fields
ACS14	AMS Address Dataset	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ACS30	AMS Incoming Calls/Letters	COUNTY, DISTRICT, TOWN
ADM01	Areas Progress Report	COUNTY, DISTRICT, TOWN
ADM06	Summary of areas letters	COUNTY, DISTRICT, TOWN
ADM08	Summary of assurance policies	COUNTY, DISTRICT, TOWN
ADM10	Redemption notifications	COUNTY, DISTRICT, TOWN
ADM18	LIFE & NON-LIFE INSURANCE COM.	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ADM30	Areas Admin & Receivership	COUNTY, DISTRICT, TOWN
ADM53	History of Not of Post Jud Int	COUNTY, DISTRICT, TOWN
ADM70	Unsecured Areas Progress Rep	COUNTY, DISTRICT, STREET, TOWN
ALC01	COAST Interface	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ALC30	AMS Adhoc Case Review	COUNTY, DISTRICT, TOWN
ALM03	EMAT Application Enquiry	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ALM05	MAES Loan Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ALM01	ID and Address Verification	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
ASY81	AMS Batch Requests	COUNTY, DISTRICT, TOWN
BAF01	Branch Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF07	Agency Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF20	Brand Details Maintenance	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BAF24	Alternative Brand Name & Addr	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
BNK01	Bank account details	COUNTY, DISTRICT, TOWN
BNK14	Bank branch details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CRD01	CARD PAYMENTS	COUNTY, DISTRICT, STREET, TOWN
CTT18	Register/Cancel R85	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CTT46	Mortgage Inquiry	BUILD_NUM, COUNTY, STREET, TOWN
CUS01	Customer details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS06	CUSTOMER BY CORRES. ADDRESS	BUILD_NAME, BUILD_NUM
CUS12	Ad-Hoc Customer Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS15	Business Customer Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS20	CUSTOMER DETAILS	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS21	Mortgage Customer Details II	TOWN
CUS22	Applicant Related Addr Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS24	AVAILABLE REFERENCE ADDRESSES	TOWN
CUS35	Mortgage Customer Details III	TOWN
CUS64	Customer Merge Preview	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
CUS83	CUSTOMER IVA/BANKRUPTCY DETAIL	COUNTY, DISTRICT, TOWN
CUS91	Maintain Cust FSCS Big Data	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
DBT02	Debt Agency Details	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
JAD06	Inv. A/C Correspondence Add.	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
IFA06	CHEQUEBOOK REQUESTS	DISTRICT, TOWN
IFA09	MAINTENANCE OF CARD	DISTRICT, TOWN
INQ01	Check digit calculation	COUNTY, DISTRICT, TOWN
INQ05	HLC CALCULATION	DISTRICT
LAP02	underwriting analysis/authoris	COUNTY, DISTRICT, TOWN
LAP03	LOAN OFFER	COUNTY, DISTRICT, TOWN
LAP05	LOAN CANCELLATION	COUNTY, DISTRICT, TOWN
LC010	Unsecured Funding Copy Details	COUNTY, DISTRICT, TOWN
LC090	Funding Company detail - Part 1	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
LET46	Letter Reference Organisations	BUILD_NAME, BUILD_NUM, COUNTY, DISTRICT, STREET
LON20	Unsecured Loan Settlement	BUILD_NAME, STREET

Duplicate Code

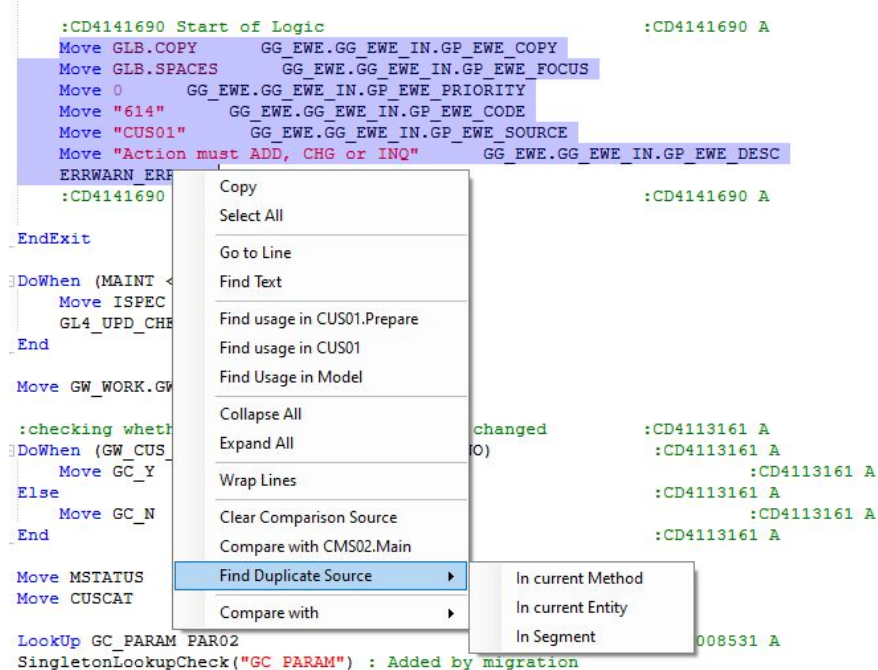


The screenshot displays a COBOL program in a code editor. A yellow box highlights a section of code from line 27 to 38, which is a logic block for error handling. This block is repeated later in the program, starting at line 61. The code includes comments and data moves, and is associated with a specific program ID.

```
27 :ME; ERROR (Action must ADD, CHG or INQ) :CD4141690 D
28
29 :CD4141690 Start of Logic :CD4141690 A
30 Move GLB.COPY GG_EWE.GG_EWE_IN.GP_EWE_COPY
31 Move GLB.SPACES GG_EWE.GG_EWE_IN.GP_EWE_FOCUS
32 Move 0 GG_EWE.GG_EWE_IN.GP_EWE_PRIORITY
33 Move "614" GG_EWE.GG_EWE_IN.GP_EWE_CODE
34 Move "CUS01" GG_EWE.GG_EWE_IN.GP_EWE_SOURCE
35 Move "Action must ADD, CHG or INQ" GG_EWE.GG_EWE_IN.GP_EWE_DESC
36 ERWARN_ERROR()
37 :CD4141690 End of Logic :CD4141690 A
38
39 EndExit
40
41 DoWhen (MAINT <> GC_INQ)
42 Move ISPEC GD_ISPEC
43 GL4_UPD_CHECK()
44 End
45
46 Move GW_WORK.GW_FILLER GW_CUS_MAP
47
48 :checking whether customer number has been changed :CD4113161 A
49 DoWhen (GW_CUS_MAP.GW_CUST_NO <> CUST_NO) :CD4113161 A
50 Move GC_Y GW_CUS_MAP.GW_CUS_POPUP :CD4113161 A
51 Else :CD4113161 A
52 Move GC_N GW_CUS_MAP.GW_CUS_POPUP :CD4113161 A
53 End :CD4113161 A
54
55 Move MSTATUS M_STATUS
56 Move CUSCAT CUS_CAT
57
58 LookUp GC_PARAM PAR02 :CN008531 A
59 SingletonLookupCheck("GC_PARAM") : Added by migration
60
61 DoWhen (GOTOSTAFF = GC_Y)
62 DoWhen (CUS01.STAFF = GC_N)
63 :ME; ERROR (Customer is not a Member of Staff) :CD4141690 D
64
65 :CD4141690 Start of Logic :CD4141690 A
66 Move GLB.COPY GG_EWE.GG_EWE_IN.GP_EWE_COPY
67 Move "GOTOSTAFF" GG_EWE.GG_EWE_IN.GP_EWE_FOCUS
68 Move 0 GG_EWE.GG_EWE_IN.GP_EWE_PRIORITY
69 Move "615" GG_EWE.GG_EWE_IN.GP_EWE_CODE
70 Move "CUS01" GG_EWE.GG_EWE_IN.GP_EWE_SOURCE
71 Move "Customer is not a Member of Staff" GG_EWE.GG_EWE_IN.GP_EWE_DESC
72 ERWARN_ERROR()
73 :CD4141690 End of Logic :CD4141690 A
74
75 :CU; GOTOSTAFF :CD4141690 D
76 EndExit
77 End
78
79 DoWhen (MAINT = GC_INQ)
80 DoWhen (CUST_MA = GC_YES)
```

Identifying Duplicate Code Patterns

- Highlight the common lines
- Select option to find 'Find Duplicate Source'
 - Matrix will perform a 'fuzzy' search
 - Ignores strings and integers
 - Searches for 'similar' data items using Levenshtein algorithm



The screenshot shows a code editor with a context menu open. The menu options are: Copy, Select All, Go to Line, Find Text, Find usage in CUS01.Prepare, Find usage in CUS01, Find Usage in Model, Collapse All, Expand All, Wrap Lines, Clear Comparison Source, Compare with CMS02.Main, Find Duplicate Source (highlighted), and Compare with. A sub-menu for 'Find Duplicate Source' is also visible, showing options: In current Method, In current Entity, and In Segment. The background code includes lines like ':CD4141690 Start of Logic', 'Move GLB.COPY GG_EWE.GG_EWE_IN.GP_EWE_COPY', 'Move GLB.SPACES GG_EWE.GG_EWE_IN.GP_EWE_FOCUS', 'Move 0 GG_EWE.GG_EWE_IN.GP_EWE_PRIORITY', 'Move "614" GG_EWE.GG_EWE_IN.GP_EWE_CODE', 'Move "CUS01" GG_EWE.GG_EWE_IN.GP_EWE_SOURCE', 'Move "Action must ADD, CHG or INQ" GG_EWE.GG_EWE_IN.GP_EWE_DESC', 'ERRWARN ERR', 'EndExit', 'DoWhen (MAINT <', 'Move ISPEC', 'GL4_UPD_CHR', 'End', 'Move GW_WORK.GW', ':checking wheth', 'DoWhen (GW_CUS', 'Move GC_Y', 'Else', 'Move GC_N', 'End', 'Move MSTATUS', 'Move CUSCAT', 'LookUp GC_PARAM PAR02', and 'SingletonLookupCheck("GC_PARAM") : Added by migration'.

Duplicate Code

Code To Match

```
1 Move GLB.COPY GG_EWE.GG_EWE_IN.GP_EWE_COPY
2 Move "GOTOSTAFF" GG_EWE.GG_EWE_IN.GP_EWE_FOCUS
3 Move 0 GG_EWE.GG_EWE_IN.GP_EWE_PRIORITY
4 Move "615" GG_EWE.GG_EWE_IN.GP_EWE_CODE
5 Move "CUS01" GG_EWE.GG_EWE_IN.GP_EWE_SOURCE
6 Move "Customer is not a Member of Staff" GG_EWE.GG_EWE_IN.GP_EWE_DESC
7 ERRWARN_ERROR()
```

CUS01.Prepare

```
88
89 Move "Please enter a customer number" GG_EWE.GG_EWE_IN.GP_EWE_DESC
90 ERRWARN_ERROR()
91 :CD4141690 End of Logic :CD4141690 A
92
93 :CU: CUST-NO :CD4141690 D
94 EndExit
95 DoWhen (GOTOSTAFF = GC_Y)
96 Recall CUS46
97 EndExit
98
99 :Below code is added as part of CR7020-WBBS :CD4144151 A
100 Move CUST_NO GG_ACS.GG_ACS_IN.GP_ACS_CUS
101
102 CHK_ACC_CUS_ST()
103
104 DoWhen (GG_ACS.GG_ACS_OUT.GP_ACS_MSG <> GLB.SPACES)
105 EndExit
106 :Above code is added as part of CR7020-WBBS :CD4144151 A
107
108 End
109
110 DoWhen ((GOTOSTAFF <> GC_Y) AND \
111 (GOTOSTAFF <> GC_N))
112 :ME; ERROR (Please enter Y or N in Go to staff) :CD4141690 D
113
114 :CD4141690 Start of Logic :CD4141690 A
115 Move GLB.COPY GG_EWE.GG_EWE_IN.GP_EWE_COPY
116 Move "GOTOSTAFF" GG_EWE.GG_EWE_IN.GP_EWE_FOCUS
117 Move 0 GG_EWE.GG_EWE_IN.GP_EWE_PRIORITY
118 Move "617" GG_EWE.GG_EWE_IN.GP_EWE_CODE
119 Move "CUS01" GG_EWE.GG_EWE_IN.GP_EWE_SOURCE
120 Move "Please enter Y or N in Go to staff" GG_EWE.GG_EWE_IN.GP_EWE_DESC
121 ERRWARN_ERROR()
122 :CD4141690 End of Logic :CD4141690 A
123
124 :CU: GOTOSTAFF :CD4141690 D
125 EndExit
126
127
128 :*****
129 :** Disability Code Validation **
130 :*****
131
132 DoWhen ((MAINT = GC_ADD) OR \
133 (MAINT = GC_CHG))
134
135 DoWhen ((APP_TO_AC <> GC_Y) AND \
136 (APP_TO_AC <> GC_N))
137 :ME; ERROR (Apply to linke :CD4141690 D
138 :d accounts must be Y or N) :CD4141690 D
139
140 :CD4141690 Start of Logic :CD4141690 A
```

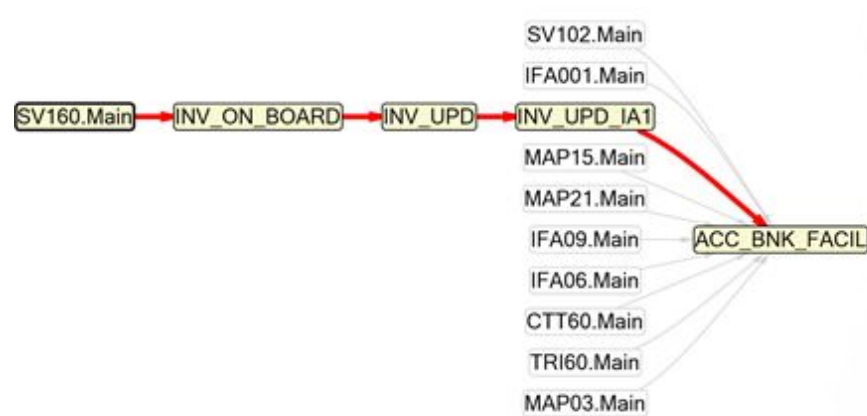
Method	Line No	Description
CUS01.Prepare	30	
CUS01.Prepare	65	
CUS01.Prepare	84	
CUS01.Prepare	115	
CUS01.Prepare	141	
CUS01.Prepare	165	
CUS01.Prepare	202	
CUS01.Prepare	232	
CUS01.Prepare	250	
CUS01.Prepare	335	
CUS01.Prepare	367	
CUS01.Prepare	400	
CUS01.Prepare	439	
CUS01.Prepare	457	
CUS01.Prepare	490	
CUS01.Prepare	509	
CUS01.Prepare	553	
CUS01.Prepare	572	
CUS01.Prepare	590	
CUS01.Prepare	608	
CUS01.Prepare	627	
CUS01.Prepare	645	
CUS01.Prepare	668	
CUS01.Prepare	691	
CUS01.Prepare	714	
CUS01.Prepare	733	
CUS01.Prepare	749	



What do I need to test?

What needs to be tested?

- If I change a method, what calls it?
 - Need to find all components which invoke this method, directly or indirectly
 - Can be difficult to determine if method is nested in other calls
- Matrix can perform 'reverse call stack' search
 - Given a method or list of method names can find who calls the method recursively
- Focus testing effort on what needs to be tested



Method Call Stacks

Summary ACC_BNK_FACIL ACS_CRT_REV

ACC_BNK_FACIL

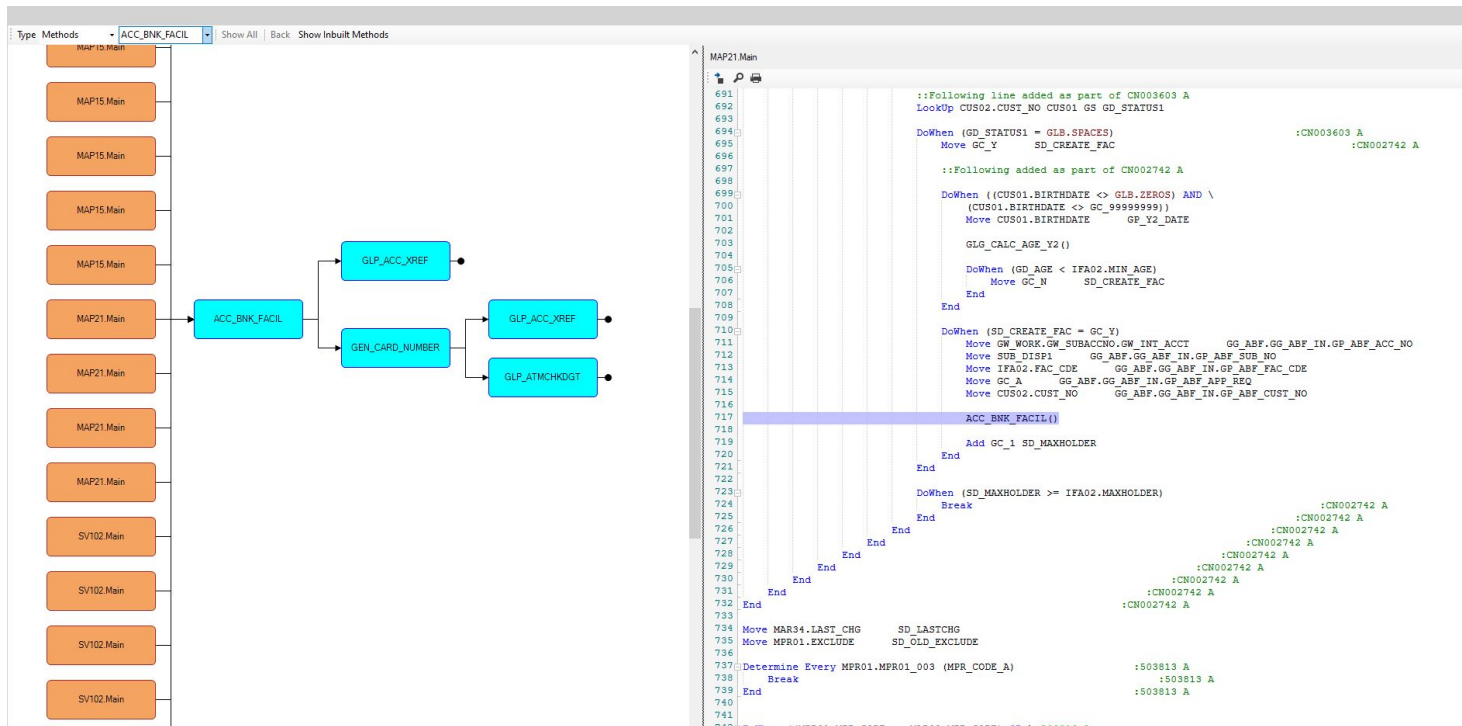
- CTT60 Main
- IFA001 Main
- IFA06 Main
- IFA09 Main
- INV_UPD_JA1 - Update Savings Account Details
 - INV_UPD - Update Savings Account
 - INV_ON_BOARD - Savings Ac On Boarding Wrapper
 - SV160 Main
- MAP03 Main
- MAP15 Main
- MAP21 Main
- SV102 Main
- TRI60 Main

Calling Method	Owner	Line No	Description
CTT60 Main	CTT60	3724	CASHIER ACCOUNT OPENING
IFA001 Main	IFA001	7759	Banking Facility Management
IFA06 Main	IFA06	119	CHEQUEBOOK REQUESTS
IFA09 Main	IFA09	1572	MAINTENANCE OF CARD
MAP03 Main	MAP03	491	ACCOUNT DETAILS
MAP15 Main	MAP15	38	Facilities Application
MAP21 Main	MAP21	717	ACCOUNT DETAILS (FURTHER LOAN)
SV102 Main	SV102	6192	Open Investment Account
SV160 Main	SV160	630	Savings Account On Boarding
TRI60 Main	TRI60	18832	Investors Account Opening

SV160 Main

```
618 :
619 :-----
620 Move SRV_TKTREF      GG_IOB.GG_IOB_IN.GP_IOB_TKTREF
621 Move SRV_ORGREF      GG_IOB.GG_IOB_IN.GP_IOB_ORGREF
622 Move "SV160"        GG_IOB.GG_IOB_IN.GP_IOB_SOURCE
623
624 DoWhen (SCREENSKEL_ADM5.SRV_USER <> GLB.SPACES)
625   Move SCREENSKEL_ADM5.SRV_USER      GG_IOB.GG_IOB_IN.GP_IOB_USER
626 Else
627   Move GW_WORK.GW_USERCODE      GG_IOB.GG_IOB_IN.GP_IOB_USER
628 End
629
630 INV_ON_BOARD()
631
632 : Error/Warning Handling
633 :-----
634 DoWhen ((GG_IOB.GG_IOB_WORK.GP_IOB_ERR = GC_Y) OR \
635 (GG_IOB.GG_IOB_WORK.GP_IOB_WRN = GC_Y))
636
637 :MV; GLB.ZEROS      GI-IOB-MSG      :CD4146208 D
638 Move GC_1      GI_IOB_MSG      :CD4146208 A
639
640
```


Graphical View of Method Call Stack

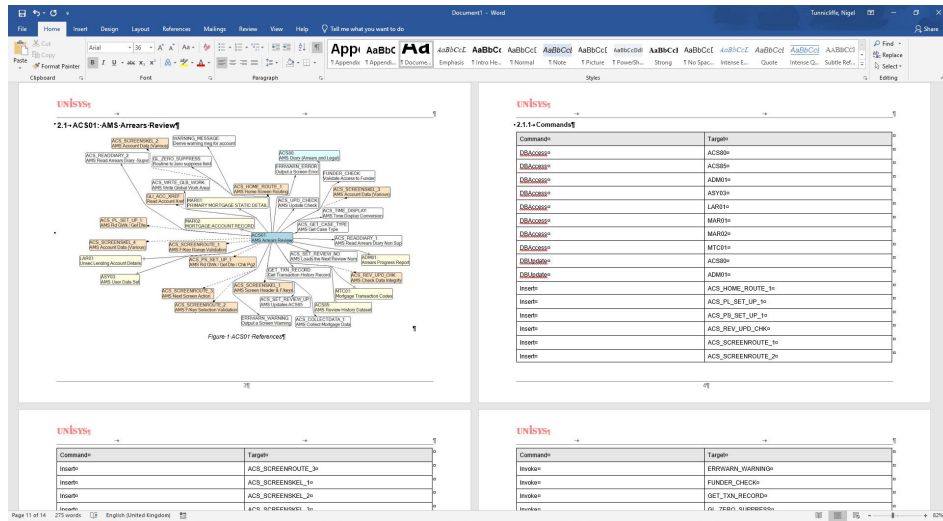




Matrix Futures

Possible Matrix Futures

- Matrix is constantly evolving/being updated
 - Most new features are user suggested enhancements/updates
- Possible future updates
 - Graphical view of Ispec References
 - Graphical view of Method call stack
 - Automated Documentation Production
 - ???
 - Let us know what would be useful





Summary

Summary

- Iterative Measurement is key
 - Are we improving?
- Address the complexity to reduce testing required
- Matrix can help identify complexity
 - How do we fix it?

**Thank
You**



