# How to Find a Fault on your Pick System

Faults come in many guises:

1. Dictionary
2. Data
3. Menus
4. Programs
5. Procs
6. Access to your Pick System

This document will lead you through all of them.

## Dictionary Faults

This is where a list does not display data correctly (e.g. from £100.00 to £10  
 00.0

0)

### Cause 1 The dictionary in the report has been changed

It looks as if the width has been changed from 10 to 3 columns to display the data in

Pick Dictionaries are created for quick on the fly, reporting onto a printer. Printers have columns, and Dictionaries use this feature to pre-set the width of each column

The #10 of all dictionary items arfe reserved for column width.

e.g.

#0 (ID) Cost.Price (The name in User ENGLISH)

#1 (Type) S (Type “S” is Standard)

#2 (Attribute) 12 (Display the 12th field – attribute from the data file)

#9 (Justification R (Right)

#10 Length 10 (allow 10 characters for this column width)

You will note that Microsoft Excel, which was created 20 years later, copies this format exactly, but because of copyright, uses different descriptions.

In this instance, if the width was 10, then the output will be “bbb£100.00” where b is a blank space.

In this instance, if the width was changed to 3, then the output will be “£10”

And underneath “0.0”

And underneath “bb0”

Where ‘b’ is a blank space. Also note, that the field is still “R”ight justified, and that all of the characters fill form the Right hand side of the column, towards the left.

In this instance, if the width was changed to 4, then the output will be “£100”

And underneath “0.00”

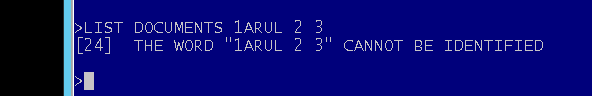
*This can happen when a dictionary is changed from 10 to 3, so that another column can be added to a report.*

The change can be from several months or a few years ago for reports run annually, or every few years.

### Cause 2 The Dictionary on the report is missing

This happens when the dictionary item was removed – often many months or even years ago.

The report is run, and looks like this:

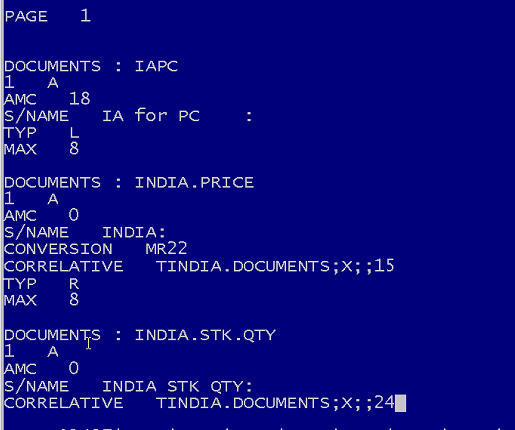


We do not know which of the three dictionary items (1ARUL, 2, or 3) are missing. We have to search for all 3 of them.

### To find a dictionary item

Firstly, we list run command like “LIST DICT DOCUMENTS = “[IA]”

This will give us a result like this:



You can see is:

Dictionary name = “IAPC”, “INDIA.PRICE”, and so on.

Choose a likely Dictionary that looks like the one we want.

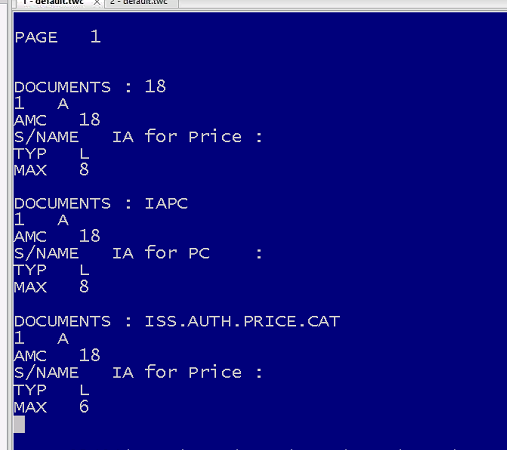
e.g. If we are looking for the Issuing Authority, then we would choose “IAPC”

Now, look the the Attribute number = The AMC – Attribute Mark Counter.

In this instance it is 18.

Type in LIST DICT DOCUMENTS WITH \*A3 = “18”

Now we find all of the dictionary items that look at Attribute 18 in the data file.



We have Dictionaries which have AMC (Attribute) = “18”

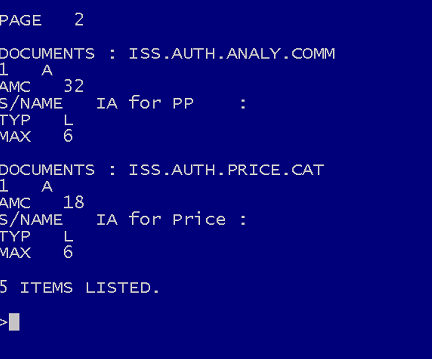
The first one is ‘”18” and the heading is “IA for Price”

The Next one is the one we chose.

The third one is ISS.AUTH.PRICE.CATT which very much looks like Issuing Authority.

To confirm, type in LIST DICT DOCUMENTS = “[ISS]” “[AUTH]”

We will look for Dictionaries with ISS or AUITH in their name



This will show

AMC of 18 for Issuing Authority for Price

AMC of 32 for Issuing Authority for Analysis (PP)

AMC of 31 for Issuing Authority for Sales Analysis

AMC of 38 for Issuing Authority for Ordering

There may be more.

There may be Correlatives or Conversions which point to Issuing Authority in a different file.

### Correlatives.

Describe how to find them here

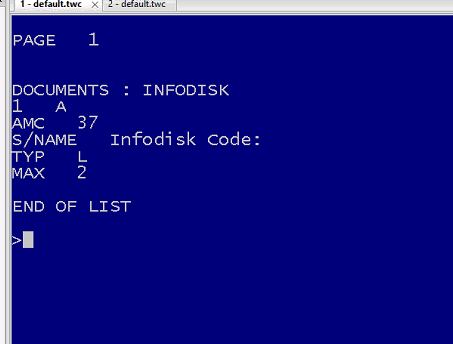
### Correlatives.

Describe how to find them here

We also need to know those documents with an INFODISK code of “01”

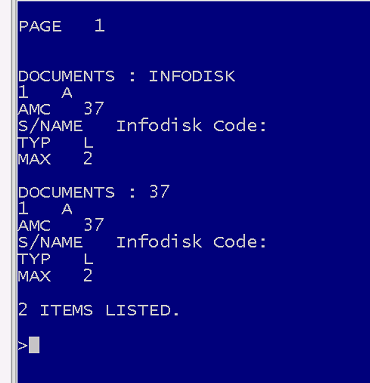
We need to find the dictionary for INFODISK

Type in LIST DICT DOCUMENTS = “INFO]”



We need to confirm it is #37 (AMC = 37).

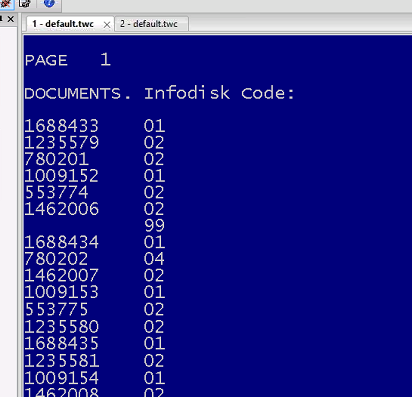
Type in LIST DICT DOCUMENTS WITH \*A2 = “37”



This confirms that #37 (AMC 37) is the attribute in the data file that stores the INFODICK code.

Simply list them to view the data (Use <Ctrl> X to escape.

LIST DOCUMENTS 37



This shows Documents with their Infodisk Code.

We can now run a sample:

## Data Faults

This is where either a piece of data has been added, altered, or removed

### Cause 1: Additional Data

The result of this is usually in a SELECT list, and looks like this:

>SELECT MERRY WITH 1 = "1!

2 ITEMS SELECTED.

>SAVE-LIST ARULS\_LIST

[239] 'ARULS\_LIST' SAVED

>LIST MERRY 1

PAGE 1

MERRY..... 1

A 1

B 1

C 2

3 ITEMS LISTED.

>GET-LIST ARULS\_LIST

2 ITEMS SELECTED.

>LIST MERRY

PAGE 1

MERRY.....

A

B

2 ITEMS LISTED.

**>ED MERRY X**

**NEW ITEM**

**TOP**

**.I**

**001+1**

**002+**

**TOP**

**.F**

**TOP**

**.FI**

**'X' FILED.**

>LIST MERRY 1

PAGE 1

MERRY..... 1.........

A 1

B 1

C 2

**X 1**

4 ITEMS LISTED.

>GET-LIST ARULS\_LIST

2 ITEMS SELECTED.

>LIST MERRY

PAGE 1

MERRY.....

A

B

2 ITEMS LISTED.

The above shows that although a LIST is used to display all of the items with a value of 1, that the **NEW item (X)** is not included.

The solution, if unsure, is to reselect the list before running the program.

This is not always possible: e.g.

1. Select all of the records to update (10 mins) and SAVE-LIST
2. GET-LIST - Create invoices (20 mins)
3. GET-LIST - Create stock lists (10 mins)
4. GET-LIST - Create letters (1 hour)

During this whole process, new records to update may be added.   
However, we do not want to include them. We only want to create letters for those items selected at the very beginning of the process, in stage 1.

### Cause 2: Amended Data

aaaaa

### Cause 3: Deleted Data

Aaaaa

## Menu Faults

This is where either a piece of data has been added, altered, or removed

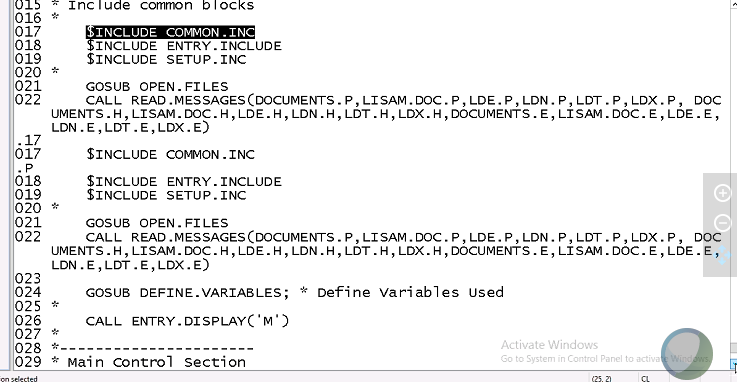
## Program Faults

This is where either a piece of data has been added, altered, or removed

In the example chosem, we are looking at the program ENTRY.MAIN in the database LI-BP on the JLL (LISAM) data warehouse.

The first 16 lines are comments

The the next 22 lines look like this



The start is:

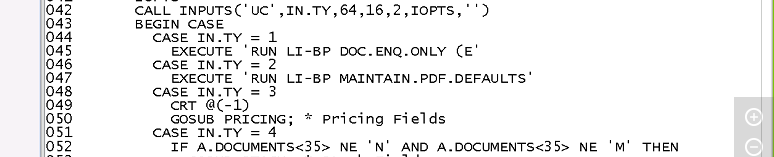
$INCLUDEs to run subroutines external to ENTRY.MAIN (6 of them)

Goto internal subroutine – and OPEN .FILES

CALL and external subroutine ENTRY.DISPLAY with a parameter of “M”

Loops around until it picks up a number – this is a Program written to look like a menu.

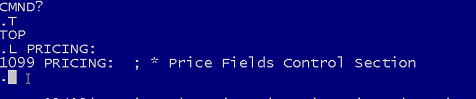
We spot very quickly that the Input program we want (Option number 3) is shown as



Which is shown as GOSUB PRICING

Further down the program we find the SUBROUTINE called PRICING:

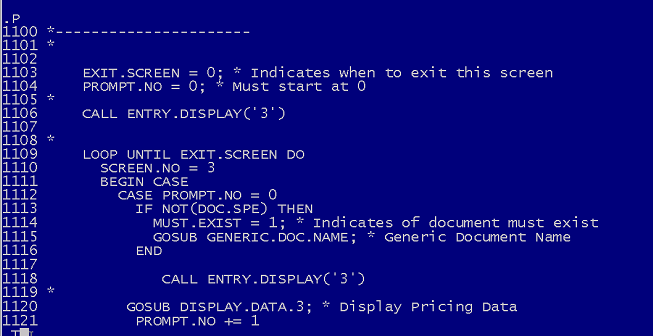
(Please note the ‘:’ character).



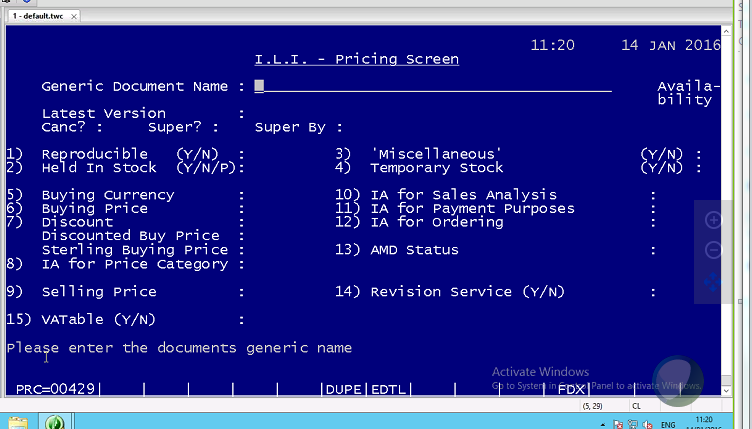
Using the “L” for Locate command, go to Top (Use the “T” command)

And Locate “PRICING:”

The PRICING: subroutine looks like this:



No we have found the program we are looking for, we run it



Find the required field. IN the example we need to find the IA ffor Price Category

It is field number 8

We also need to find:

Infodisk Code Not in this program screen

Availability Top Right hand corner – Not a field to be changed – it has no Field Number

Reproducible Field 1

Held in Stock Field 1

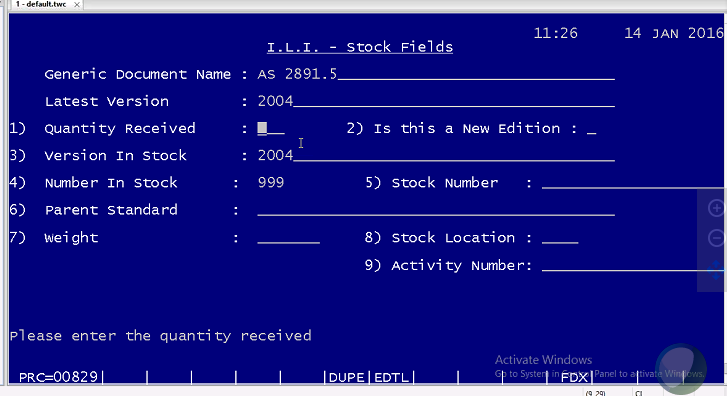
Miscellaneous Field 3

Temporary Stock Field 4

Number in Stock Not in program screen

Using “N” for Next Screen

Number in Stock Field 4 in this new Program/Screen heading of “I.L.I. – Stock Fields”



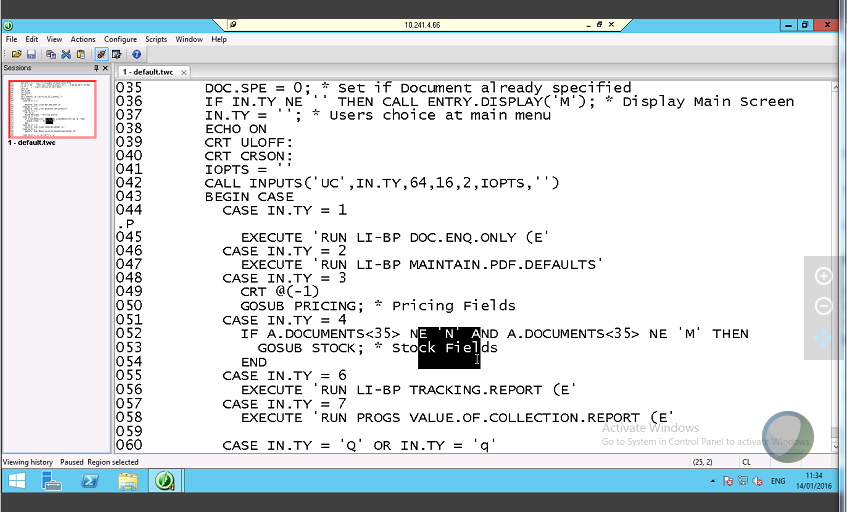
Infodisk Code we get from a Dictionary Item – there was only One field with this description.

Looking for the Number in Stock

We go back to the calling program (INPUT.MAIN)

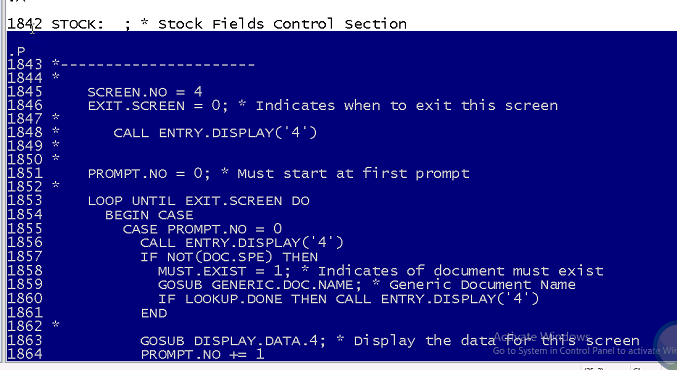
List 44 lines, and we see that the option4 is STOCK FIELDS, and is at sub routine STOCK:

By locating STOCK:



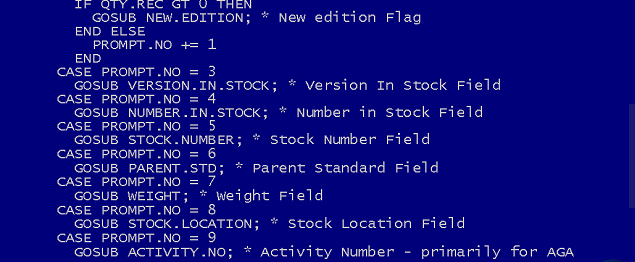
Locate the Subroutine STOCK:

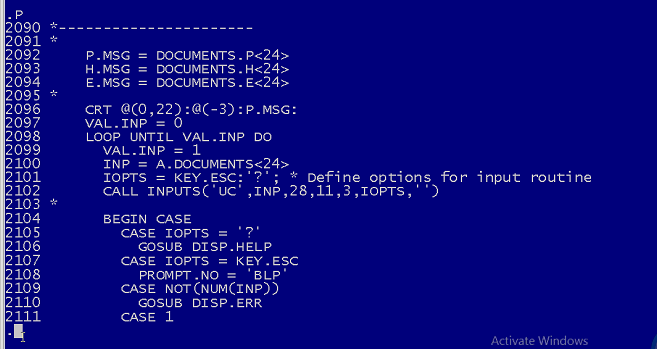
(Line 1842)



Is the STOCK: Subroutine

We are looking for field 4 (Number in Stock) in this subroutine



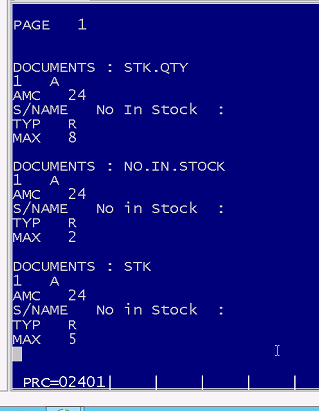


From here we note that the A.DOCUMENTS variable, attribute 24 (line 2100) is holding the variable used to update the “number in stock field”.

By going back to TCL and running the command

LIST DICT DOCUMENTS WITH \*A” (Attribute number) = “24”

We get:



Which confirms that 24 is the attribute holding the Number in Stock.

## Proc Faults

What is a PROC?

A Proc is a data item

It begins (Line1) is always PQ

This stands for PROCESS QUERY

(Most PROCS are Queries).

Some Procs are commands – e.g. Logon PROCS, and System Commands

A sample is like this:

PQ

Comment sample PROC to list only the MD

H SORT MD NOPAGE

P

This PROC will SORT the MD into ID order, and display all 3061 MD records in the MD database in order of Name, with 154 pages (153 carriage returns – NOPAGE).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

First we need to

Find the PROC.

Follow the User commands, and find the Menu and process used.

e.g. Janet logged into JANET, and went to TCL, and typed in HOCKEY

This logged her onto another account.

She got a new menu, and chose option 3.

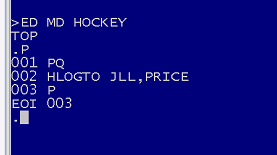
We will Log onto JANET, then same

Drop to TCL

Instead of running the PROC called HOCKEY, we examine it.

ED MD HOCKEY

P (List 22 lines)

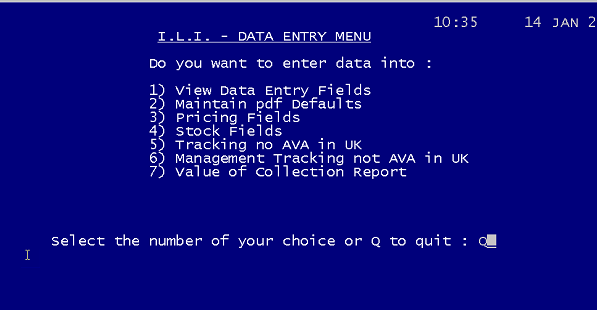


This shows us the command HOCKEY

It is a LOGTO command, and nothing else.

Therefore, this means all of JANET account’s Security settings (SYS2) are carried over.

Type in HOCKEY, or, exit (logoff) and logon to JLL with the correct password

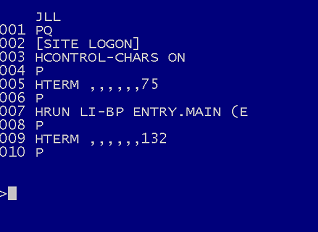


On this occasions, on this account we go to TCL when choosing Quit. Other account will send you to LOGON PLEASE promt.

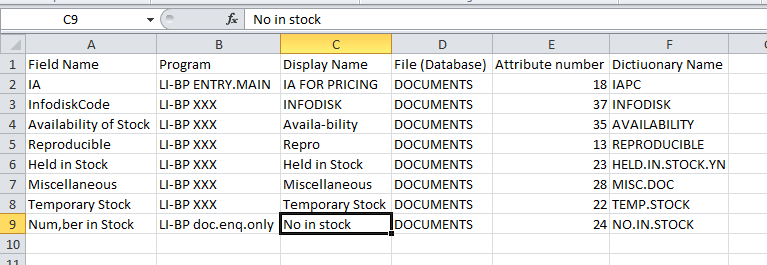
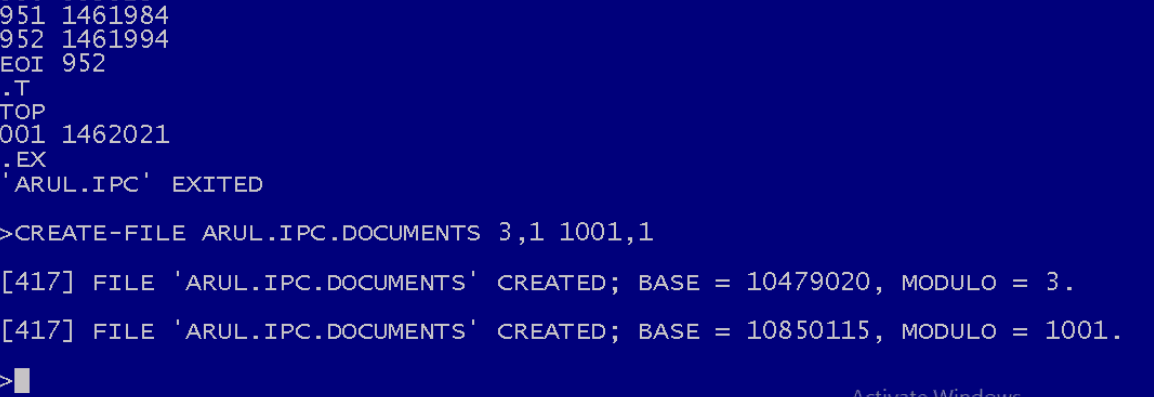
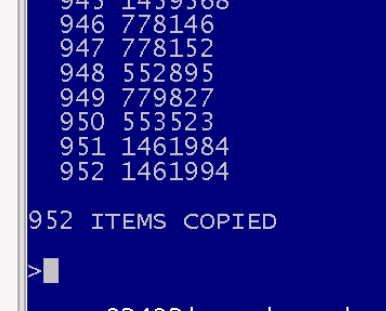
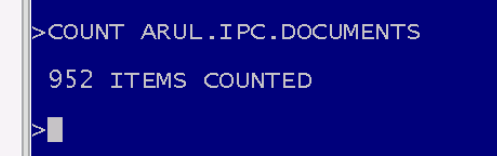
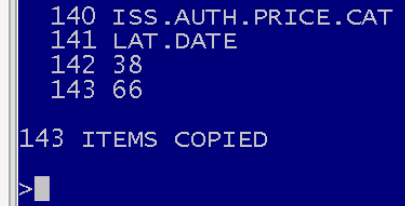
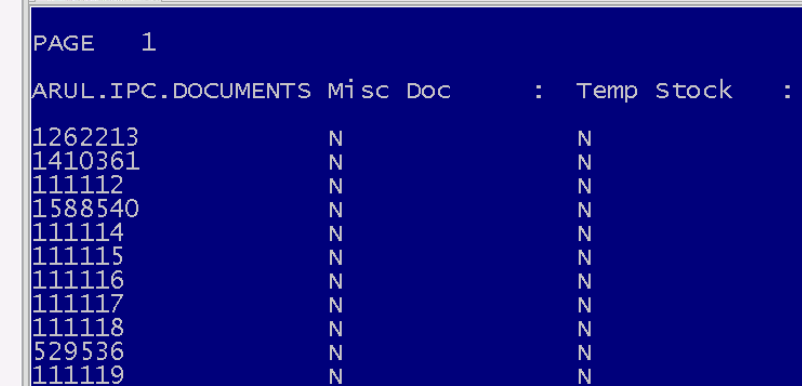
Type in WHO, to confirm you are still on JLL account

Type in CT MD JLL – this is the first command that is used when you logon to the account JLL

Another Proc appears



This PROC will:

1. Run another PROC subroutine called LOGON in the SITE database
2. Puts the CONTROL CHARACTERS ON – This will enable Printing control characters to be used e.g. ESC<ctrl>&100sk will enable Landscape Also, Control characters for display like <ctrl>G will make the buzzer/bell work
3. Changes the Terminal Emulator (makes Pick look look an old Green and Black system when it is actually html, PHP, C#, C++, .net, etc system) into 75 columns from 80 columns wide.
4. Runs a program called ENTRY.MAIN in the LI-BP database, using the “E” option which Hides All Errors. This is a bad move. We should never use this. If there is an error, we should either use it (e.g.1020 items selected is an error message – use the information) or we should fix the error. Never hide an error message.
5. Sets the Terminal to 132 columns for display, after the program has completed.
6. We continue as before to find th attributes, in which files we need to change.
7. Until we get a list like this:
8. 
9. All of the above is to find the correct items to change.
10. There 2 SELECT lists:
    1. ARUL.IPC
    2. ARUL.IPC-E.01
11. Due to not having a test system, we have to create a test file.
12. Type in the following:
13. CREATE-FILE ARUL.IPC.DOCUMENTS 3,1 1001,1
14. 
15. This creates a Temporary Test File System
16. We can now change the Production system
    1. If something goes wrong, then we
    2. Copy from the Test System back in to the Production.
17. We need to make the Test System the same as the Production System
18. Type in
19. GET-LIST ARUL.IPC
20. COPY DOCUMENTS
21. (ARUL.IPC.DOCUMENTS
22. This will copy the 952 DOCUMENTS from DOCUMENTS into ARUL.IPC.DOCUMENTS
23. 
24. To prove it is correct, type in:
25. COUNT ARUL.IPC.DOCUMENTS
26. 
27. Next is the Dictionaries
28. COPY DICT DOCUMENTS \*
29. (DICT ARUL.IPC.DOCUMENTS
30. 
31. Move to the new ‘Test System’ file
32. Type LIST ARUL.IPC.DOCUMENTS MISC.DOC TEMP.STOCK
33. 
34. We choose a few examples:
    1. 1462021 IPC 2612
    2. 111078 IPC A 600
    3. 111160 IPC A 610
35. Using Janet’s choice (111078 – IPC A 600) we note that this has already been updated.
36. Also, 2 new attributes are required to complete the request:
    1. Fields 0, 1, 2, 3, and 4 in the first screesn are all required to be “N”
    2. Field 11, and 12 should be null and IPC
    3. Number in Stock should be “0” and Version in stock should be null.
    4. The 2 new fields are “Version in stock” and field 12 (IA for Ordering) to be IPC.
37. We need to find the attributes in which files for these 2 information fields.

## Access Faults

This is where either a piece of data has been added, altered, or removed

## System Faults

This is where either a piece of data has been added, altered, or removed

## Other Faults

This is where either a piece of data has been added, altered, or removed