IDENTIFYING RET'S, DET'S, FTR'S

4

Objective of Section:

Learn the necessary techniques to identify a RET, a DET and a FTR. Understanding how to identify DET's and FTR's is critical to distinguish one transaction from another. While in practice understanding the exact number of DET's and FTR's may not impact a function point count, understanding DET's and FTR's can help understand how to count function points for enhancement function point counts. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

Record Element Type (RET): A RET is user recognizable sub group of data elements within an ILF or an EIF. It is best to look at logical groupings of data to help identify them. The concept of RET will be discussed in detail in the chapters that discuss internal logical file and external interface files. Additionally, a short article, *Understanding RET's* can be found at record types.

File Type Referenced (FTR): A FTR is a file type referenced by a transaction. An FTR must also be an internal logical file or external interface file.

Data Element Type (DET): A DET is a unique user recognizable, non-recursive (non-repetitive) field.   A DET is information that is dynamic and not static.   A dynamic field is read from a file or created from DET's contained in a FTR. Additionally, a DET can invoke transactions or can be additional information regarding transactions. If a DET is recursive then only the first occurrence of the DET is considered not every occurrence.

Understanding the FTR's and DET's helped distinguish one transaction from another transactions.   This concept will be discussed in detail later in this book.

Rating:

All of the components are rated based upon DET's, and either RET's or FTR's.

Component	RET's	FTR's	DET's
External Inputs (EI)		✓	✓
External Outputs (EO)		✓	✓
External Inquiries (EQ)		✓	✓
External Interface Files (EIF)	✓		✓
Internal Logical Files (ILF)	✓		✓

Transaction DET's:

External Inputs: Data Input Fields, Error Messages, Calculated Values, Buttons
External Outputs: Data Fields on a Report, Calculated Values, Error Messages, and Column Headings that are read from an ILF. Like an EQ and EO can have an input side and output sides.
External Inquiries: Input Side - field used to search by, the click of the mouse. Output side - displayed fields on a screen.

Record Element Types (RET's):

Record element types are one of the most difficult concepts in function point analysis. Most record element types are dependent on a parent - child relationship. In this case, the child information is a subset of the parent information. In a parent child relationship there is a one to many relationship. That is, each child piece of information is linked directly to a field on the parent file. More will be discussed about RET's in the internal logical file and external interface file sections.

Tips to Identify RET's and DET's early in the life cycle:

RET's and DET's may be difficult to evaluate early in the software life cycle. Since RET's and DET's are essential to rating components, several techniques can be used to rate components.

- Rate all transactional function types and data function types as Average.
- Determine how are transactional function type and data function types rated in similar type applications. Are the majority of data function types rated as low in similar type applications?

DET's for GUI

Using the strict definition of a data element provided by IFPUG's Counting Practices Manual. "A data element is a user recognizable, non recursive field." Unfortunately this does not provide much guidance when counting GUI applications. In fact, the IFPUG Counting Practices manual does not provide much detail on counting, radio buttons, check boxes, pick list, drop downs, look ups, combo boxes, so on and so forth.   In GUI applications, a data element is information that is stored on an internal logical file or that is used to invoke a transaction. A comprehensive article on the application of Function Points to New and Emerging Technologies exist at Website\Articles\index.htm

Radio Buttons

Radio Buttons are treated as data element types. Within a group of, a frame, radio buttons the user has the option of selecting only one radio button; so only one data element type is counted for all the radio buttons contained in the frame.

Check Boxes

Check Boxes differ from radio buttons in that more than one check box can be selected at a time. Each check box, within a frame, that can be selected should be treated as a data element.

Command Buttons

Command buttons may specify an add, change, delete or inquire action. A button, like OK, may invoke several different types of transactions.
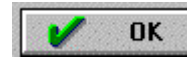
According to IFPUG counting rules each command button would be counted as a data element for the action it invokes.   In practice this data element will not impact the rating of the transaction, but it does help understand and dissect a screen full of transactions.

A button like next may actually be the input side of an inquiry or another transaction.

For example, a simple application to track distributors could have fields for Distributor Name, Address, City, State, Zip, Phone Number, and Fax Number. This would represent seven fields or (seven data elements) and the add command button would represent the eighth data element.   In short, the "add" external input represent a one external input with eight data elements, the "change" external input represents another external input with eight (seven data elements plus the "change" command button), and the "delete" external input represents the last external input with eight data elements (seven fields plus the "delete" command button).

Display of Graphical Images or Icons

A display of a graphical image is simply another data element. An inventory application, for example, may contain data about parts. It may contain part name, supplier, size, and weight and include a schematic image of the part. This schematic is treated as a single data element.

Sound Bytes

Many GUI applications have a sound byte attached. This represents one data element. The number of notes played is simply recursive information. If the length of the sound byte increases, then the data element remains one.   For example, you can play the "Star Spangled Banner" for two seconds or four seconds, but you'll still count the sound bytes as one data element. The longer it is played the more recursive information it has.

Photographic Images

A photographic image is another data element, and is counted as one. A human resource application may display employee name, start date, etc. and a photograph of the employee. The photograph is treated the same as employee name or employee start date. The photograph is stored and maintained like any other piece of information about the employee.

Messages

There are three types of messages that are generated in a GUI application: error messages, confirmation messages and notification messages. Error messages and confirmation messages indicate that an error has occurred or that a process will be or have been completed.   They are not an elementary or independent process alone, but they are part of another elementary process. A message that would state, "zip code is required" would be an example of an error message. A message that would state, "are you sure you want to delete customer" is an example of a

confirmation message. Neither type of message is treated as a unique external output, but each is treated as a data element for the appropriate transaction.

On the other hand, a **notification messages** is a business type message. A notification is an elementary process, has some meaning to the business user and is independent of other elementary processes. It is the basis of processing and a conclusion being drawn. For example, you may try to withdraw from an ATM machine more money than you have in your account and you receive the dreaded message, "You have insufficient funds to cover this transaction." This is the result of information being read from a file regarding your current balance and a conclusion being drawn.   A notification message is treated as an External Output.

DET's For Real Time Systems

Using the strict definition of a data element provided by IFPUG's Counting Practices Manual. "A data element is a user recognizable, non-recursive field." Unfortunately this does not provide much guidance when counting real time or embedded systems. In fact, the IFPUG Counting Practices manual does not provide any detail on counting these types of systems.

Some traditional definitions can be applied directly to real time and embedded systems. The fields on a diagnostics file: time of diagnostics, hardware state during diagnostics, temperature, voltage, so on and so forth would all be examples of data elements.

Real Time Systems may not have any "traditional user interface." That is, the stimulus for the Real Time System may be it's own output – or state.   A real time or embedded systems can signal to determine current Hardware State (or location) and determine the appropriate adjustment (input) based on the current state.

Navigation

Navigation is moving from one transaction to another.

Skill Builder:

1. The following information is heard in the Rome Train Station. How many data elements are heard?   That is, what information varies from one train arrival to the next?

The train arriving from Florence will arrive on Track 46 at 8:30 a.m.
The train arriving from Naples will arrive on Track 43 at 11:00 a.m.

2. The totals on a particular report change colors depending if the amount is above or below $ 500.
3. For example if the amount is -$250 it appears as **$250**, but if the amount is over 0 then the value appears blue. For example if the amount is **$1,000**.   How many data elements are represented by the number and by the color?

EXTERNAL INPUTS

<div style="text-align: right">5</div>

Objective of Section:

Describe and define the concepts necessary to identify and rate External Inputs. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

External Inputs (EI) - is an elementary process in which data crosses the boundary from outside to inside. This data may come from a data input screen or another application. The data may be used to maintain one or more internal logical files. The data can be either control information or business information. If the data is control information it does not have to maintain an internal logical file.

If an external input adds, changes and deletes (maintains) information on an internal logical file, then this represents three external inputs. External inputs (especially change & delete) may be preceded by an external inquiry (see the section on external inquiries). Hence a full function screen is add, change, delete and inquiry (more will be discussed about inquiries later in the book).

Rating:

Like all components, EI's are rated and scored. The rating is based upon the number of data element types (DET's) and the file types referenced (FTR's). DET's and FTR's are discussed earlier. The table below lists both the level (low, average or high) and appropriate score (3, 4 or 6).

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Counting Tips:

Try to ask the question, do external inputs need more or less than 2 files to be processed? For all the EI's that reference more than 2 FTR's, all that is needed to know is if the EI has more or less than 4 data element types referenced. If the EI has more than 4 DET's the EI will be rated

as high; less than 4 DET's the EI will be rated as average. Any EI's that reference less than 2 FTR's should be singled out and counted separately.
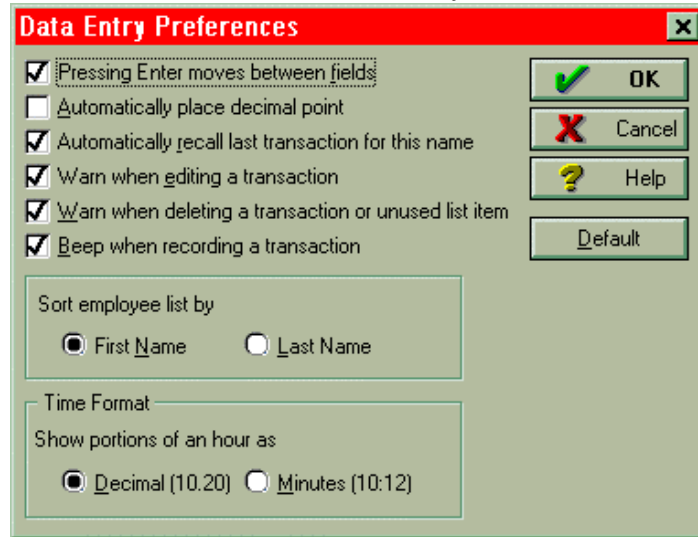
Examples:

EI's can be business data, control data and rules based data.

Business Data: Customer Name, Address, Phone, and so on and so forth.

Control Data:

The data elements are those that invoke the transaction or change the behavior of the application. Each check box represents a data element. Additionally, the sort employee list radio buttons represents one data element as well as the time format radio buttons.

Control information changes or alters the state (or behavior) of the application. Control information specifies how, what, and when data will be processed.

Data Elements:

Unique sets of data elements help distinguish external input from other external input.

- Data Input Fields
- Calculated Values or Derived Data that are stored
- Error Messages
- Confirmation Messages
- Recursive fields are only counted as one DET.
- Action keys (command buttons such as OK, Next, so on and so forth)
- Multiple Action Keys that perform the same function are counted only as one DET.

File Types Referenced (FTR's):

Unique FTR's helps distinguish external input from other external input. An FTR must be either an Internal Logical File and/or External Interface File. Each internal logical file that an external input maintains is counted as an FTR. Any internal logical file or external interface file that is referenced by an external input as part of the elementary process of maintaining an internal logical file would be considered an FTR also. For example, an External Input may update an

EXTERNAL OUTPUTS

6

Objective of Section:

Describe and define the concepts necessary to identify and rate External Outputs. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

External Outputs (EO) - an elementary process in which **derived data** passes across the boundary from inside to outside.   Additionally, an EO may update an ILF. The data creates reports or output files sent to other applications. These reports and files are created from information contained in one or more internal logical files and external interface files.

External Output
(1 EO, 2 FTR's)
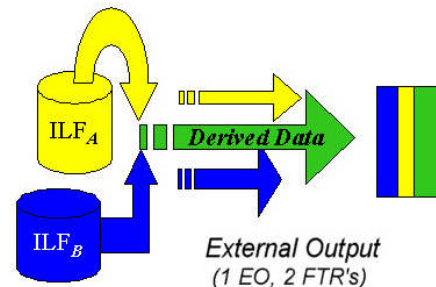
Derived Data is data that is processed beyond direct retrieval and editing of information from internal logical files or external interface files. Derived data is usually the result of algorithms, or calculations. *Derived data occurs when one or more data elements are combined with a formula to generate or derive an additional data element(s).* This derived data does not appear in any FTR (internal logical file or external interface file).

An algorithm is defined as a mechanical procedure for performing a given calculation or solving a problem in a series of steps.

A calculation is defined as an equation that has one or more operators. An operator is a mathematical function such as addition, subtraction, multiplication, and division (+, -,x, /).

Rating:

Like all components, EO's are rated and scored.   The rating is based upon the number of data elements (DET's) and the file types referenced (FTR's). The rating is based upon the **total number of unique** (combined unique input and out sides) data elements (DET's) and the file types referenced (FTR's) (combined unique input and output sides). DET's and FTR's were discussed earlier in this section.   The table below lists both the level (low, average or high) and appropriate score (4, 5 or 7).

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Counting Tips:

You may ask the question, Do external outputs need more or less than 3 files to be processed? For all the EO's that reference more than 3 files, all that is needed to know is if the EO has more or less than 5 data element types. If the EO has more than 5 data element types then the EO will be rated as high, less than 5 the EO will be rated as average. Any EO's that reference less than 3 files should be singled out and counted separately.

Terminology:

The definition states an EO contains information, which ***derived data*** passes across the boundary from inside to outside. Some confusion may arise because an EO has an input side. The confusion is the definition reads data passes across the boundary from inside to outside. The input side of an EO is search criteria, parameters, etc does not maintain an ILF. The information that a cross from outside to inside (input side) is not permanent data, but it is transient data. The intent of the information coming from outside the application (input side) is not to maintain an ILF.

Examples:

Unlike other components EO's almost always contain business data. Rule base data and control based "outputs" are almost always considered External Inquiries. This is true due to the fact that rule data and control type data is not derived (or derivable).

Notification Messages are considered EO's. A notification message differs from an error message. A notification message is an elementary process, while an error message (or confirmation message) is part of an elementary process. A notification message is the result of some business logic processing. For example, a trading application may notify a broker that the customer trying to place an order does not have adequate funds in their account.

Derived Data displayed in textual fashion (rows and columns) and graphical format is an example of two external outputs.

Data Elements:

Unique sets of data elements help distinguish one external output from another. Keep in mind that a DET is something that is dynamic.
(A DET is a unique user recognizable, non-recursive (non-repetitive) field)
- Error Messages
- Confirmation Messages

- Calculated Values (derived data)
- Values on reports that are read from an internal logical file or external interface file.
- Recursive values or fields (count only once)
- Generally, do not count report headings (literals) as data elements unless they are dynamic. That is, if the report headings are read from files that are maintained they may be DET's also.
- System generated dates that are on the tops or reports or are displayed are normally not counted as DET's. If system generated dates is part of business information of the external output they should be counted as DET's. For example, the date an invoice is printed or the date a check is printed.

File Types Referenced (FTR):

Unique FTR's help distinguish one external output from another. An FTR must be either an Internal Logical File and/or External Interface File.

The elementary process associated with an external output may update an internal logical file or external interface file. For example, the elementary process that produces as payroll check may include an update to a file to set a flag to indicate that the payroll check was produced. This is not the same as maintaining the file. Maintained is the process of modifying data (adding, changed and deleting) via an elementary process (via an External Input). The primary intent of an EO is not to maintain an ILF.

Uniqueness:

A unique set of data elements, and/or a different set of FTR's, and/or a unique set of calculations makes one external output unique or different from other external outputs. That is, one of the following must be true:

- Unique or different set of data elements
- Unique or different set of FTR's
- Unique or different calculations
- Unique processing logic

Understanding Enhancement Function Points:

Modification of any of the items, which make an External Output unique from other external outputs, causes the EO to be "enhanced." If any of the following are true:

- DET's added to an EO
- DET's modified on an EO. The DET was included in the last FP Count.
- A New FTR
- Modifications to a calculation that appears on an EO.

EXTERNAL INQUIRIES

Objective of Section:

Describe and define the concepts necessary to identify and rate External Inquiries. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

External Inquiry (EQ) - an elementary process with both input and output components that result in data retrieval from one or more internal logical files and external interface files. The input process does not update or maintain any FTR's (Internal Logical Files or External Interface Files) and the output side does not contain derived data.

1 EQ, 1 FTR
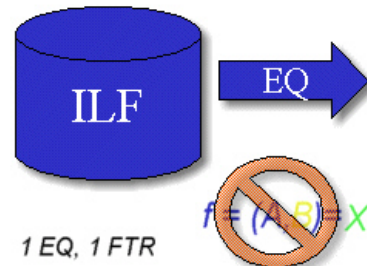
Rating:

Like all components, EQ's are rated and scored. Basically, an EQ is rated (Low, Average or High) like an EO, but assigned a value like and EI.   The rating is based upon the **total number of unique** (combined unique input and out sides) data elements (DET's) and the file types referenced (FTR's) (combined unique input and output sides). DET's and FTR's were discussed in earlier chapter. If the same FTR is used on both the input and output side, then it is counted only one time. If the same DET is used on both the input and output side, then it is only counted one time.

Functional Complexity Matrix (shared table between EO and EQ)

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Examples:

EQ's can contain business data, control data and rules based data.

Business Applications: An example of Business data is customer names, addresses, phone number, so on and so forth. An example of Rules Data is a table entry that tells how many days a customer can be late before they are turned over for collection.

Drop Down List (a listing of customers by name) would be an example of an EQ.

A screen full of customer address information would be an example of an EQ.

Terminology:

The definition states that an EO contains information, which ***derived data*** passes across the boundary from inside to outside.   Some confusion may arise because an EO has an input side. The confusion is the definition reads data passes across the boundary from inside to outside. The input side of an EO is search criteria, parameters, etc does not maintain an ILF. The information that a cross from outside to inside (input side) is not permanent data, but it is transient data. The intent of the information coming from outside the application (input side) is not to maintain an ILF.

Data Elements:

Unique sets of data elements help to distinguish one external inquiry from another external inquiry.

- Input Side
    - ☆ Click of a the mouse
    - ☆ Search values
    - ☆ Action keys (command buttons)
    - ☆ Error Messages
    - ☆ Confirmation Messages (searching)
    - ☆ Clicking on the an action key
    - ☆ Scrolling
    - ☆ Recursive fields are counted only once.
- Outside
    - ☆ Values read from an internal logical file or external interface file
    - ☆ Color or Font changes on the screen
    - ☆ Error Messages
    - ☆ Confirmation Messages
    - ☆ Recursive fields are counted only once.
- The combined (unique) total input and outside DET's are used when rating EQ's.

Like an EI, action keys that perform the same function but appear multiple times are counted as only one DET.

INTERNAL LOGICAL FILES

9

Objective of Section:

Describe and define the concepts necessary to identify and rate Internal Logical Files. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

Internal Logical Files (ILF) - a user identifiable group of logically related data that resides entirely within the application boundary and is maintained through External Inputs.

Even though it is not a rule, an ILF should have at least one external output and/or external inquiry. That is, at least one external output and/or external inquiry should include the ILF as an FTR. Simply put, information is stored in an ILF, so it can be used later. The EO or EQ could be from another application. It is worth noting that it is possible that a specific ILF is not referenced by EO or EQ, but it is used by an EI (other than the EI that maintains it).

Again, even though it is not a rule, an ILF should have at least one external input.

Rating:

Like all components, ILF's are rated and scored. The rating is based upon the number of data elements (DET's) and the record types (RET's). DET's and RET's were discussed earlier. The table below lists both the level (low, average or high) and appropriate score (7, 10 or 15).

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Counting Tips:

Determine the appropriate row first then the column. Ask the question, *do all files contain one record type or more than one record type?* If all or many of the files only contain one record type, then all that is needed to know if the file contains more or less than 50 data elements types (DET's). If the file contains more than 50 data elements the file will be rated as average, if less than 50 data element types the file will be considered low. Any files that contain more than one record type can be singled out and counted separately.

Examples:

ILF's can contain business data, control data and rules based data.   The type of data contained in an ILF is the same type of data an EI to contains and maintains.

It is common for control data to have only one occurrence within an ILF. For example control data file may only contain parameter settings, or a status setting. For example, part of the on board automobile system only contains current information, oil pressure, engine temperature, so on and so forth. This particular process of the on board system does not care about historical data – only the current instance. When the status changes the file is updated with current information and there is no historical information. The on board system may keep track of historical changes in diagnostics files, but this would be a totally separate process.   This process is not used to keep the car running, but to help a mechanic understand what has been going on with the engine.

Real Time and Embedded Systems: For example, Telephone Switching is made of all three types, Business Data, Rule Data and Control Data. Business Data is the actual call, Rule Data is how the call should be routed through the network, and Control Data is how the switches communicate with each other. Like control files it is common real time systems will have only one occurrence in an internal logical file.

Business Applications: An example of Business data is customer names, addresses, phone number, so on and so forth. An example of Rules Data is a table entry that tells how many days a customer can be late before they are turned over for collection.

Record Element Types:

The idea behind RET's is to quantify complex data relationships maintained in a single FTR.

Record element types are one of the most difficult concepts in function point analysis. Most record element types are dependent on a parent - child relationship. The child information is a subset of the parent information. In a parent child relationship there is a one to many relationship.
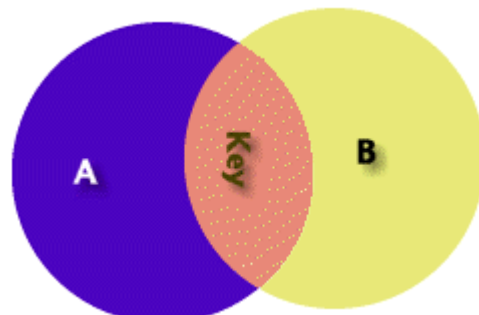

Figure 3 (two ILF, one RET each)

Figure 3 represents two separate logical groups of data A and B. In this case *some A are B*.

Figure 4 represents one logical group of data A two record types. In this case *All B are A*.

Imagine a customer file that contains Name, Address, so on and so forth. In addition all the credit cards and credit card numbers of the customer are contained in the file. This would be an example of 2 record types. There would be multiple occurrences of credit cards and numbers for each customer. The credit card and numbers are meaningless when not linked to the customer.
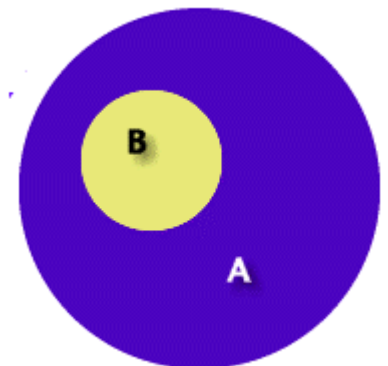
Figure 4 (two RET, one ILF)

EXTERNAL INTERFACE FILES

Objective of Section:

Describe and define the concepts necessary to identify and rate External Interface Files. The exercises at the end of the section help the student demonstrate that they have gained the basic knowledge required.

Definition:

External Interface Files (EIF) - a user identifiable group of logically related data that is used for reference purposes only. The data resides entirely outside the application boundary and is maintained by another applications external inputs. The external interface file is an internal logical file for another application.   An application may count a file as either a EIF or ILF not both.

Each EIF included in a function point count must have at least one external output or external interface file against it. At least one transaction, external input, external output or external inquiry should include the EIF as a FTR.

Every application, which references the EIF, needs to include it in their FP Count. Some organizations have a pull theory and others have a push theory of data. The pull theory is an external application "reaching into" another applications and retrieving data. Those organizations which have push theory require applications to create interfaces (EO or EQ) which other applications read.

Rating:

Like all components, EIF's are rated and scored.   The rating is based upon the number of data elements (DET's) and the record types (RET's). DET's and RET's were discussed earlier in this section.   The table below lists both the level (low, average or high) and appropriate score (5, 7 or 10).

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (5)	Low(5)	Average (7)
2 to 5 RET	Low (5)	Average (7)	High (10)
6 or More RET	Average (7)	High (10)	High (10)

Counting Tips:

Only count the part of the file that is used by the application being counted not the entire file. The internal logical file, of another application, that you access may have a large amount of data, but only consider the DET's and/or RET's that are used when rating an EIF.

Determine the appropriate row first then the column. Ask the question, *do all files contain one record type or more than one record type?* If all or many of the files only contain one record type, then all that is needed to know if the file contains more or less than 50 data elements types (DET's). If the file contains more than 50 data elements the file will be rated as average, if less than 50 data element types the file will be considered low. Any files that contain more than one record type can be singled out and counted separately.

Examples:

EIF's can contain business data, control data and rules based data.

Real Time and Embedded Systems: For example, Telephone Switching is made of all three types, Business Data, Rule Data and Control Data. Business Data is the actual call, Rule Data is how the call should be routed through the network, and Control Data is how the switches communicate with each other.

Business Applications: An example of Business data is customer names, addresses, phone number, so on and so forth. An example of Rules Data is a table entry that tells how many days a customer can be late before they are turned over for collection.

Technology Issues:

Lotus Notes refers to data stores as "forms." Client/Server Applications may store information on the host or client. Count it only one time. COBOL Applications may use a variety of data stores such as IMS, DB2 etc.… It is important to view data from the "logical model."

Standard Documentation:

- Table Layouts
- Interface Diagrams
- Database descriptions
- Logical data models
- Field sizes and formats
- Design Documentation
- Functional Specifications
- User Requirements

Tips to Identify EIF's early in the life cycle:

The following types of documentation can be used to assist in counting external interface files prior to system implementation.