iTip	Guidance from the Functional Sizing Standards Committee on topics important to you	

Logon

iTip # 3 – (version 1.1 04/26/2018)

Author: Peter Thomas		
Reviewers:		
Diana Baklizky	Dan French	Roopali Thapar
Bonnie S. Brown	Steve Keim	Adri Timp
E. Jay Fischer	Tammy Preuss	Charles Wesolowski

iTips provide guidance on topics important to the FPA community. They explain the application of IFPUG FPA method in a particular situation. iTips are not rules, but interpretation of the rules, and provide guidance using a realistic example to explain the topic being covered.

This iTip is focused on describing the IFPUG FPA method as it applies to counting basic logon functionality in a typical application. This iTip includes a series of examples but is not an exhaustive examination of the subject.

Background

The four examples described below cover the different alternative functionality of a logon process (sometimes referred to as login) required to include the components of security: determining who can logon to an application, determining what the user can do (typically referred to as role-based security), and tracking when the user has accessed the application (accountability), in various combinations.

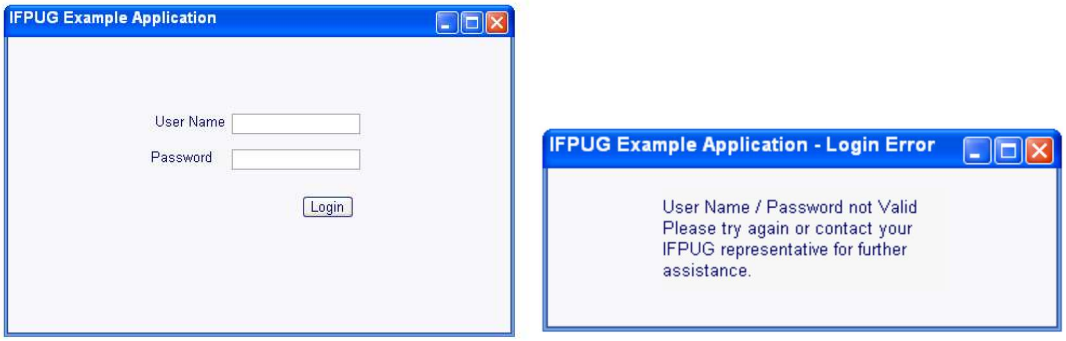
For these examples, the application places the user on a "landing page" (menu) after logon. The examples show the logon processing meeting the rules to be measured as a transactional function. Each example introduces more complex processing and additional DETs.

The examples assume that administrative functions required for maintenance of the Valid Users ILF are part of the application being measured. The measurement of these maintenance transactions is outside the scope of this iTip, however guidance is given on the analysis of the ILF which stores the logon data.

The iTip assumes that the FP analysis step for determining the counting scope, purpose, boundary and functional user requirements was performed and that the result indicated a need to count the Logon functionality as part of a development project function point count.

Example 1

An application contains a logon function to validate the user name and password. A sample screen is shown below. The user enters a user name and password and clicks the Login button. The application validates the entered details against the Valid Users file and either allows the user access to the application or provides the user a message that the details are incorrect.



The primary intent of the logon transaction is to present to the user success or failure of the validation of the user name and password. The Logon processing leaves the application ready for the user to proceed, for example by selecting a menu option.

The logon function meets the rules for an EQ. It has four DETs: user name, password, the action of clicking the Login button, and a message. To perform the validation, the function reads the Valid Users ILF which is counted as one FTR for the logon function. No ILFs are updated. A single low complexity EQ is counted for logon.

The Valid Users file is counted as one low complexity ILF containing the two DETs (user name and password).

Function	Type	DETs	FTRs
Logon	EQ	4 – user name, password, login button, message	1 – Valid Users

Function	Type	DETs	RETs
Valid Users	ILF	2 – user name, password	1

Example 2

In this example the logon function is expanded to include determining what functionality the user can perform. The same Logon screen as shown in the previous example is used. As illustrated in Example 1, the user enters a user name and password and clicks the Login button. The application validates the entered details against the Valid Users file and either allows the user access to the application or provides the user a message that the details are incorrect. There is an additional requirement that as part of the logon, the user is provided appropriate access to the application functions based on the user role (role-based security) stored in the Valid Users file.

The primary intent of the logon transaction is to present to the user success or failure of the validation of the user name and password. During the processing of the transaction it reads an additional attribute from the "Valid Users" ILF which holds a value of either "standard" or "admin." This value is used to alter the behavior of the application to limit access to specific functionality. No ILFs are updated.

The logon function meets the rules for an EO (i.e., alters the behavior of the application) and has four DETs: user name, password, the action of clicking the login button, and a message.

The Valid Users ILF is counted as one low complexity ILF with three DETs (user name, password and user role).

Function	Type	DETs	FTRs
Logon	EO	4 – user name, password, login button, message	1 – Valid Users

Function	Type	DETs	RETs
Valid Users	ILF	3 – user name, password, user role	1

Example 3

In this example the logon function described in Example 1 is expanded (differently from example 2) to track user logon dates (provide an audit trail). The same Logon screen as shown in the first example is used. As described in Example 1, the user enters a user name and password and clicks the Login button. The application validates the entered details against the Valid Users file and either allows the user access to the application or provides the user a message that the details are incorrect. There is an additional requirement that, as part of the logon, a value, the last logon date, is written to the Valid Users ILF.

The primary intent of the logon transaction is still to present to the user success or failure of the validation of the user details. As the process now includes maintaining an ILF, it meets the rules for an EO and has four DETs: user name, password, the action of clicking the login button, and a message.

The additional attribute recording the logon date is counted as an additional DET in the Valid Users ILF, if it is user recognizable. The additional DET leaves the ILF unchanged as a low complexity.

Note: this additional attribute recorded in the data function for the logon date does not cross the boundary and is therefore not counted for the transactional function.

Function	Type	DETs	FTRs
Logon	EO	4 – user name, password, login button, message	1 – Valid Users

Function	Type	DETs	RETs
Valid Users	ILF	3 – user name, password, last logon date	1

Example 4

In this example the logon function described in Example 2 is expanded to illustrate independently maintained role-based permissions (access rights) for the user. As previously described, the user enters a user name and password and clicks the Login button. The application validates the entered details against the Valid Users file and either provides the user a message that the details are incorrect or completes the processing by referencing an additional file to determine the permissions enabled for that user role. The screen displays the previous logon timestamp along with a message confirming that logon was successful. There is an additional requirement that, as part of the logon, a value, the last logon date, is written to the Valid Users ILF.

The primary intent of the logon transaction is still to present to the user success or failure of the validation of the user details. During the processing of the transaction it reads additional attributes from the "Valid Users" ILF which hold the previous logon timestamp and user role. The user role is used to read the "Role-based Permissions" ILF to determine which functions (access rights) the user is allowed to perform. The role-based permissions for the assigned role are used to alter the behavior of the application to limit access to specific functionality.

As the process now includes altering the behavior of the application and maintaining an ILF, it meets the rules for an EO and has five DETs: user name, password, the action of clicking the login button, a message, and previous logon timestamp. Two logical files are used.

Function	Type	DETs	FTRs
Logon	EO	5 – user name, password, login button, message, previous logon timestamp	2 – Valid Users, Role-based Permissions

Function	Type	DETs	RETs
Valid Users	ILF	3 – user name, password, user role	1
Role-based Permissions	ILF	2 – user role, permissions	1

Summary

Logon has been shown to be an EQ or an EO depending on the functionality.

Frequently Asked Questions (FAQ)

Is logon the only way to implement role-based access to application functionality?

There are other more sophisticated implementation techniques e.g., using the relational database or virtualization middleware. Example 2 shows the functional requirement being logically satisfied at logon via a simple implementation technique. Example 4 shows the functional requirement being logically satisfied via a more sophisticated technique (e.g., using a relational database).

Are multiple ILFs counted if additional data about the user is being stored?

To measure the data functions correctly, entity dependencies must be determined. The data must be entity dependent to be measured in the same ILF and maintained by the application. Additional information on this topic is provided in the logical files chapter in the CPM.

Is an extra FTR counted for the security ILF/EIF for screens providing functions limited to some users?

No. The security ILF/EIF is generally only counted as an FTR for security administration transactions (e.g., maintaining the user role values). Other transactions (e.g., check printing) generally do not reference the security ILF/EIF logically, so they do not count an extra FTR. Who is accessing the transaction is different from what the transactional function is doing.

What if additional information is displayed on the logon landing page?

If the landing page displays additional information to the user, then additional DETs are counted according to the CPM DET rules.

What if user information and/or role-based permissions are maintained in another application?

If the user information and/or role-based privileges are maintained in another application, the data function is classified as an EIF as long as the application being counted does not also maintain the data function.

Further Reading

IFPUG Counting Practices Manual,

Part 1, Section 5.5 – Measure Transactional Functions.

Part 2, Chapter 7 – Measure Transactional Functions.

Part 3, Chapter 2 – Logical Files

Part 4, Chapter 2 – Example: Add Window Security (Page 2-100)

An article on Computer Security http://en.wikipedia.org/wiki/Access_controls