

Transaction calibration

On this page:

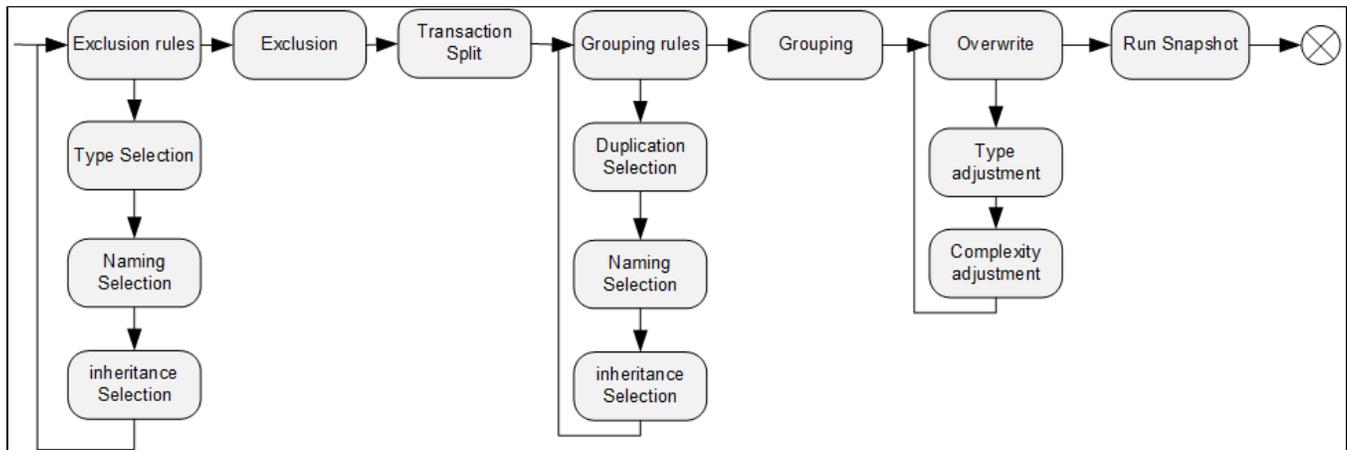
- [Introduction](#)
- [Calibrate Data Functions](#)
 - [Adjust the Data Function type](#)
 - [Adjust the FP value of Data Functions](#)
- [Calibrate Transactional Functions](#)
 - [Adjust the type of Transactional Functions](#)
 - [Adjust the FP value of Transactional Functions](#)
- [Automation: Insert custom treatments during the computation](#)
 - [Enabling custom treatments](#)
 - [Activating custom mode for CAST Storage Service \(CSS\)](#)
 - [Activating custom mode for SQL Server](#)
 - [Activating custom mode for Oracle](#)
 - [Customizing calibration](#)
 - [Customizing calibration Final Filter phase](#)
 - [Customizing Impact Factor in Enhancement Function Point calculation](#)

Introduction

Function Point calibration is done at the end of the process and requires [Transaction configuration](#) to be completed. Moreover, all the transactions that have been identified in the application must have already been validated by the SME. If the [Transaction review](#) pinpointed issues, then you should go back to the [Transaction configuration](#) step before starting the Function Point calibration.

Function Point calibration is done by **grouping**, **splitting** and **excluding** Transactional Functions and Data Functions in order to adjust the results. A [calibration kit](#) is available as an extension and provides dedicated rules that will help the AI administrator to calibrate Function Points faster.

The following image presents the process related to the Function Point calibration:



Calibrate Data Functions

Adjust the Data Function type

The Data Function type (EIF and ILF) is determined at computation and is based on the type of links pointing to Data Entities. Types that are taken in account are the following:

- **Use Select**, **Use Insert**, **Use Delete** and **Use Update**
- **Access Read** and **Access Write**



Where CAST AIP was not possible to fully determine the type of link (ex: only Use or Access), Data Functions are typed as **EIF by default**. After investigation, the AI administrator can change the type to **ILF** if necessary.

A set of predefined rules and built-in rules will automatically set up the correct types for Data Entities. However, it is often necessary to review the accesses to the data layer to refine the type of links at analysis time.

Figure 1:

Technical name	Object type	Functional name	Computed type	Custom type
Orders	Microsoft table	Orders	ILF	
AuthorIdCodes	Microsoft view	AuthorIdCodes	EIF	
publishers	Microsoft table	publishers	EIF	
titles	Microsoft table	titles	ILF	
roysched	Microsoft table	roysched	EIF	
discounts	Microsoft table	discounts	EIF	
stores	Microsoft table	stores	EIF	
authors	Microsoft table	authors	ILF	
StoreIdView	Microsoft view	StoreIdView	EIF	
sales	Microsoft table	sales	ILF	
titleview	Microsoft view	titleview	EIF	
titleauthor	Microsoft table	titleauthor	EIF	
Stock	Microsoft table	Stock	ILF	
Invoice	Microsoft table	Invoice	ILF	

Adjust the FP value of Data Functions

i This action should not need to be completed frequently and it is important to review and understand the root cause of improper complexity of Data Functions before adjusting the values.

In case of flat files or elements created by a Universal Analyzer configuration, you will have the default value of the type (column 3 of the Figure 1 below). In this case, and if you have access to additional information, you can overwrite the calculated value of the added components in order to secure the count.

Technical name	Object type	Functional name	FP value	Custom FP
Orders	Microsoft table	Orders	7	
AuthorIdCodes	Microsoft view	AuthorIdCodes	5	
publishers	Microsoft table	publishers	5	
titles	Microsoft table	titles	7	
roysched	Microsoft table	roysched	5	
discounts	Microsoft table	discounts	5	
stores	Microsoft table	stores	5	
authors	Microsoft table	authors	7	
StoreIdView	Microsoft view	StoreIdView	5	
sales	Microsoft table	sales	7	10
titleview	Microsoft view	titleview	5	
titleauthor	Microsoft table	titleauthor	5	
Stock	Microsoft table	Stock	7	
Invoice	Microsoft table	Invoice	7	

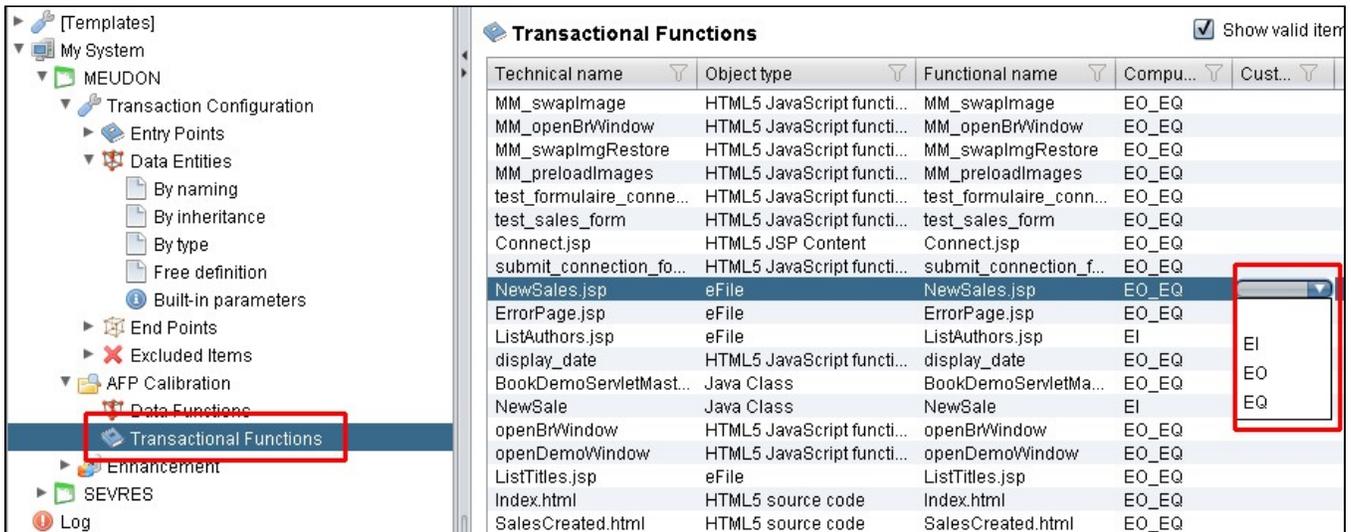
Figure 1:

Description	Type	DET	RET
Custom Data Function CICS - Semi persistent Data	CICS Transient Data		3
Custom Data Function CICS - Semi persistent Da...	Unknown CICS Transient Data		3
Custom Data Function Cobol External File	Cobol External File		3
Custom Data Function CICS/Cobol - Semi persist...	Cobol Transient Data		3
Custom Data Function GSAM	IMS FILE		3
Custom Data Function GSAM - Not delivered	IMS FILE - legacy		3

Calibrate Transactional Functions

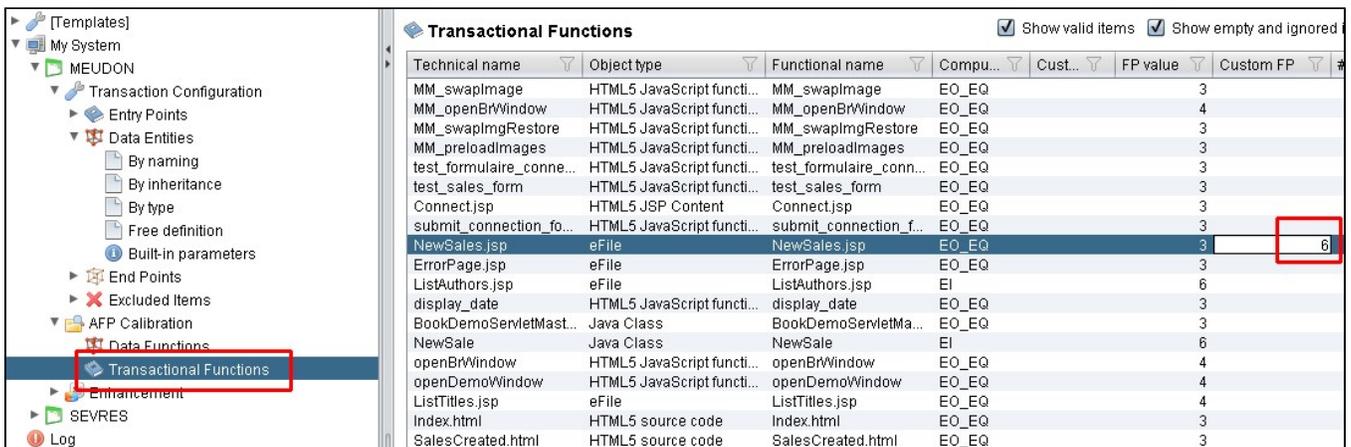
Adjust the type of Transactional Functions

By default, a Transactional Function is typed as **EO_EQ**. If it contains at least one **Update**, **Delete**, or **Insert** access to a Data Entity, then this is considered as managing input data flow and is typed as **EI**. Otherwise, if the Transactional Function only contains **Select** accesses or **undefined** access links, then it is typed as **EO_EQ**. In that case, after investigation, the AI administrator can change the type of Transactional Functions:



Adjust the FP value of Transactional Functions

This action should not need to be completed frequently and it is important to review and understand the root cause of improper complexity of Transactional Functions before adjusting the values.



Automation: Insert custom treatments during the computation

The computation process allows you to insert custom treatments in dedicated procedures executed at specific steps. This section presents which procedures are available and how some of them can be used.

Enabling custom treatments

The custom procedures will be executed only if the right option is activated in the Analysis Service. By default the custom mode is not activated and it is necessary to execute the following SQL code to enable custom treatments during the computation.

Activating custom mode for CAST Storage Service (CSS)

Execute the following SQL code in PostgreSQL DBMS to activate the custom mode:

```
select SET_TCC_OPTION_FPCUSTOM (1,1) --to set the option with trace
select SET_TCC_OPTION_FPCUSTOM (1,0) --to set the option with no trace
select SET_TCC_OPTION_FPCUSTOM (0,0) --to disable the custom execution
```

Activating custom mode for SQL Server

Execute the following SQL code in SQL Server DBMS to activate the custom mode:

```
exec SET_TCC_OPTION_FPCUSTOM 1,1 --to set the option with trace
exec SET_TCC_OPTION_FPCUSTOM 1,0 --to set the option with no trace
exec SET_TCC_OPTION_FPCUSTOM 0,0 --to disable the custom execution
```

Activating custom mode for Oracle

Execute the following SQL code in Oracle DBMS to activate the custom mode:

```
--to set the option with trace you should execute the
declare
RET int:=0;
Begin
RET :=SET_TCC_OPTION_FPCUSTOM ( 1,1);
End;

--to set the option with no trace
declare
RET int:=0;
Begin
RET :=SET_TCC_OPTION_FPCUSTOM ( 1,0);
End;

--to disable the custom execution
declare
RET int:=0;
Begin
RET :=SET_TCC_OPTION_FPCUSTOM ( 0,0);
End;
```

Customizing calibration

The table below shows the custom procedures that are available for the transaction calibration:

Procedure name	When it is executed
TCC_FP_USR_DF_DELETE_RULE	DSSAPP_FP_APP_DATAFUNC
TCC_FP_USR_DF_ADJ_DETRET_RULE	DSSAPP_FP_APP_DATAFUNC
TCC_FP_USR_DF_IGNORE_RULE	DSSAPP_FP_APP_DATAFUNC
TCC_FP_USR_DF_GROUP_RULE	DSSAPP_FP_APP_DATAFUNC
TCC_FP_USR_DF_ADJ_TYPE_RULE	DSSAPP_FP_APP_DATAFUNC
TCC_FP_USR_TF_DELETE_RULE	DSSAPP_FP_APP_TRANSACT
TCC_FP_USR_TF_IGNORE_RULE	DSSAPP_FP_APP_TRANSACT
TCC_FP_USR_TF_GROUP_RULE	DSSAPP_FP_APP_TRANSACT
TCC_FP_USR_TF_ADJ_DET_RULE	DSSAPP_SCOPE_IFPUG_DONE
TCC_FP_USR_TF_ADJ_FTR_RULE	DSSAPP_SCOPE_IFPUG_DONE
TCC_FP_USR_TF_ADJ_TYPE_RULE	DSSAPP_SCOPE_IFPUG_DONE
TCC_FP_USR_FINAL_RULE	DSSAPP_SCOPE_IFPUG_DONE

Customizing calibration Final Filter phase

The table below shows the custom procedures that are available for the transaction calibration Final Filter phase:

Procedure name	When it is executed
----------------	---------------------

FP_FILTER_DATAFUNCTIONS	Before computation
FP_FILTER_TRANSACTIONS	Before computation

Customizing Impact Factor in Enhancement Function Point calculation

The table below shows the custom procedures that are available for customizing the Impact Factor used to adjust the Transactional Functions and Data Functions in Enhancement Function Point computation:

Item type	Procedure name	When is it executed?
Data Function	EFP_USR_IF_DEL_DF_RULE	During snapshot computation
Data Function	EFP_USR_IF_MOD_DF_RULE	During snapshot computation
Data Function	EFP_USR_IF_ADD_DF_RULE	During snapshot computation
Transactional Function	EFP_USR_IF_DEL_TF_RULE	During snapshot computation
Transactional Function	EFP_USR_IF_MOD_TF_RULE	During snapshot computation
Transactional Function	EFP_USR_IF_ADD_TF_RULE	During snapshot computation