

8.3.22 - Additional notes

- [CAST Transaction Configuration Center](#)
 - [Saving empty transaction objects - new feature](#)
 - [Upgrade behavior](#)
 - [Call graph of empty Transactions](#)
 - [Function Points computation report](#)
 - [Summary of Function Points computation report changes](#)
 - [Assessing Function Points for empty Transactional Functions based on non-empty Transactional Functions](#)
 - [Computing settings](#)
 - [Results](#)
 - [AFP Calibration](#)
 - [Transactional functions](#)
 - [Enhancement Panel](#)
 - [Impacts](#)
 - [Limitations](#)
 - [View the list of objects of a transaction](#)
 - [View the list of Data Functions of an assessed transactional function](#)
 - [Items that are considered datafunctions of a transaction](#)



Additional notes for the features and improvements introduced in **AIP Core 8.3.22**.

CAST Transaction Configuration Center

Saving empty transaction objects - new feature

Transactions are more and more frequently used for purposes other than AFP/AEP counting. However, the concept of "Transaction" was until now strongly Function Points-oriented, meaning Transactions which did not access any Data Entity or End Point did not always have their objects saved. This new feature aims to change the way such Transactions are collected and managed.

All Transactions are still identified by their Entry Points, but objects that are part of Transactions which do not access any Data Entity or End Point **can now be saved**, whichever enhancement measure is selected for their application: **EFP** (Enhancement Function Points) or **AEP** (Automated Enhancement Points) - see [TCC - Enhancement node - Right hand panel](#).

In terms of **AFP**, saying that a Transaction is not valid can still be done by using the existing **Delete** or **Ignore** action. If a Transaction does not make sense, then the corresponding Entry Point must be removed.

Transactions considered as valid for AFP counting but neither accessing any Data Entity nor End Point will still have both their FTR and DET properties set to 0.

Having this type of empty Transactions in AFP/AEP counting is controlled by the new Function Point computation setting called [Save objects of empty Transactions](#). The value of this setting will by default be set to "**ALWAYS**" for a new application, whichever enhancement measure is enabled (EFP or AEP), meaning that objects of its empty Transactions will always be saved.

To prevent variations in computation duration and volume of saved data for an application whose AIP version has just been upgraded to **8.3.21**, the value of this setting will be set to "ONLY AEP" by the upgrade process. This setting value means that objects of empty Transactions will always be saved if the application enhancement measure is AEP, and never be saved if it is EFP, which is the behavior up to and including **8.3.20**.

Before computing a new Snapshot or launching a Function Points computation, the [Save objects of empty Transactions](#) setting value can be changed in the following dialog boxes of the CAST Transaction Configuration Center:

- [Application Computation Settings](#)
- [Global Computation Settings](#)

Global Computation Settings

Data Function default type: EIF

Filter look-up tables: YES

Data Functions automatic FK/PK merge: YES

Save objects of empty Transactions (those that do not reach any Data Entity or End Point): ALWAYS

FP value for empty Transactional Functions:

Automatic configuration refresh: YES

Ok Cancel

Upgrade behavior

To prevent variations in computation duration and volume of saved data for an application whose AIP version has just been upgraded to 8.3.21 - the version where objects of empty Transactions can be saved - the upgrade process will set the value of the **Save objects of empty Transactions** setting to "ONLY AEP". However, since for new applications the default value of this setting is "ALWAYS", then the value of this setting in the **Global Computation Settings** dialog will be set to "ALWAYS" too. Consequently, where the upgraded application uses **Global Computation Settings** (instead of **Application Computation Settings**) it is necessary to override these global settings, which is indicated by a "*" in front of the setting along with a change in the label of the "[x] Use Global Computation Settings" check-box:

'Monster_3_AEP' Computation Settings ✕

Use Global Computation Settings (the values below will be ignored except those whose name starts with *)

Data Function default type: EIF ▾

Filter look-up tables: YES ▾

Data Functions automatic FK/PK merge: YES ▾

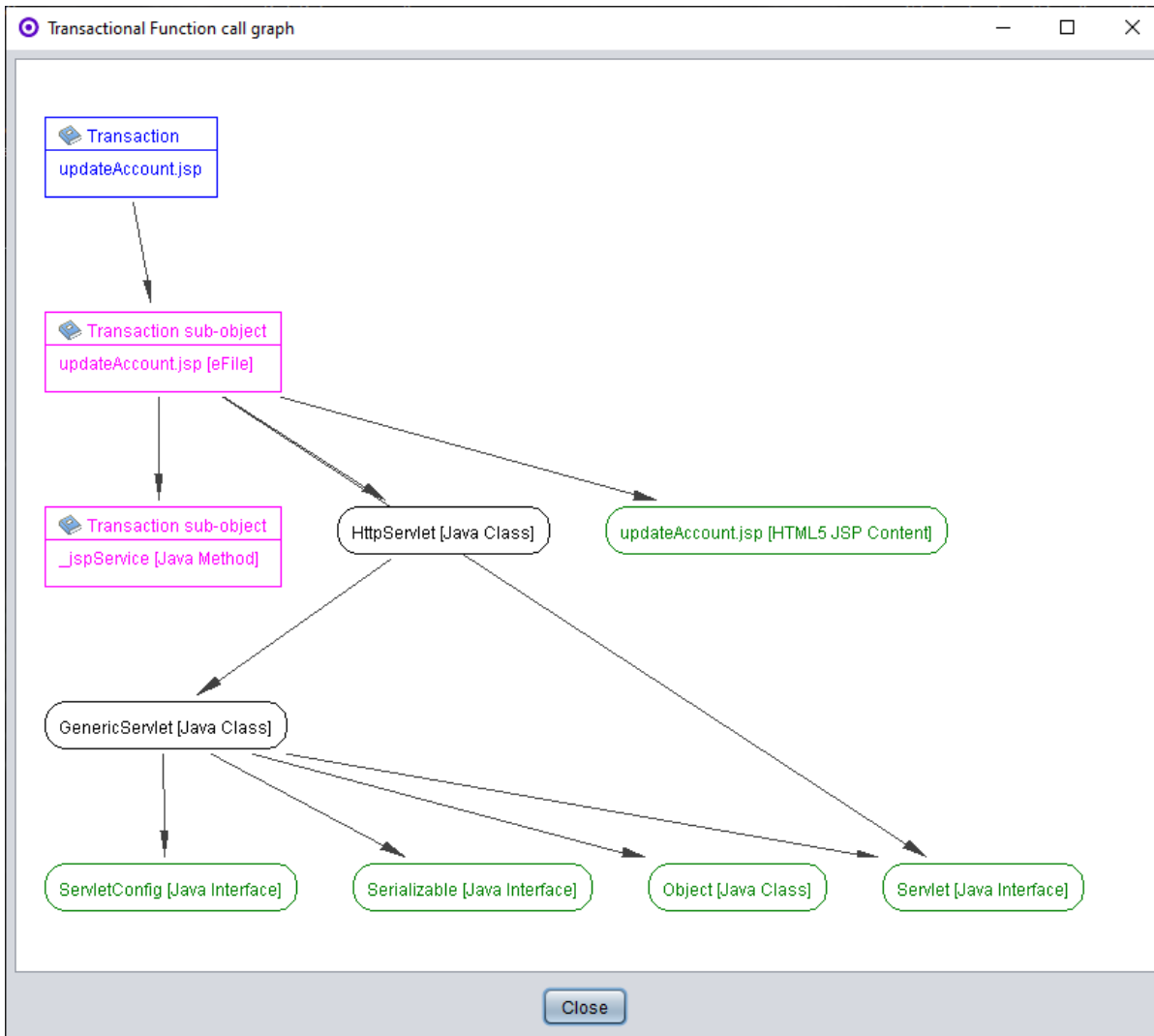
Save objects of empty Transactions (those that do not reach any Data Entity or End Point): ONLY AEP ▾

Default FP value for empty Transactional Functions: 0 ▾

Automatic configuration refresh: YES ▾

Call graph of empty Transactions

If objects of empty Transaction are saved and there are any, their **call graph** and their **full call graph** can be viewed in the same way as for those of non-empty Transactional Functions. Below is an example of what such a call graph looks like:



Function Points computation report

The Function Points computation report (generated during the **Compute** action) has been refined to provide information regarding:

- the per-application setting about whether objects of empty Transactions will be saved, or won't
- how many empty Transactions have been found
- among the above empty Transactions, how many will have their Function Point value estimated (depending upon the **Default FP value for empty Transactional Functions** setting)

Below is an example of the Function Points computation report in previous releases, taking the example of FPs computed for 3 applications in a single shot (2 applications are using AEP, 1 application is using EFP):

```

Number of applications with full call graph      : 2 (#33949, #34110)
Number of applications with reduced call graph  : 1 (#34065)
Number of relevant objects in repository       : 5323
Number of OMG AFP-links in repository         : 11043
Number of DF records / TR end points          : 248
Number of transactions                         : 150
Number of empty / estimated transactions      : 13
Number of links -> DF records / TR end points : 1864
Number of links -> other objects in full graph : 6039
Number of transaction links generated (sum)    : 7903
Number of objects in largest transaction      : 180
Number of elements in largest SCC group       : 12
Saving graph results...
Saved full call graphs of transactions in application #33949 (6563 details).
Saved call graphs of transactions in application #34065 (164 details).
Saved full call graphs of transactions in application #34110 (1176 details).
Computation finished.

```

Below is an example of the Function Points computation report in 8.3.21, taking the same applications as above:

- the 2 applications whose Enhancement measure is set to AEP will have the objects of their empty Transactions saved,
- the objects of empty Transactions not being saved for the single application whose Enhancement measure is set to EFP (as would be the case after an upgrade to 8.3.21),
- a default FP value for empty Transactional Functions having also been set for one of the applications:

```

Number of applications with full call graph      : 2 ( #33949 (empty TRs saved), #34110 (empty TRs saved)
)
Number of applications with reduced call graph  : 1 ( #34065 (empty TRs not saved) )
Number of relevant objects in repository       : 5323
Number of OMG AFP-links in repository         : 11043
Number of Data Entities + End Points          : 248
Number of empty Transactions with estimated FP value : 2
Number of non-empty + empty Transactions      : 150 ( = non-empty: 137 + empty: 13 )
Number of links -> Data Entities + End Points : 1864 ( = DE+EP: 812 + intermediate objects: 1052 )
Number of links -> other objects in full graphs : 6039 ( includes 81 links in empty Transactions )
Number of Transaction links generated (sum)    : 7903 ( = 1864 + 6039 )
Number of objects in the largest non-empty Transaction : 180
Number of objects in the largest empty Transaction : 22
Number of elements in the largest SCC group    : 12
Number of elements allowed in SCC groups      : 1000 at most
Number of elements in largest used SCC group   : 3
Number of ignored SCC groups (too large)      : 0
Saving graph results...
Saved full call graphs of transactions in application #33949 (6563 details).
Saved reduced call graphs of transactions in application #34065 (164 details).
Saved full call graphs of transactions in application #34110 (1176 details).
Computation finished.

```

Summary of Function Points computation report changes

Changed lines:

- Number of applications with full call graph
- Number of applications with reduced call graph
-> in addition of the formerly reported application# (because their name is generally unavailable from the context where Transaction graphs are computed), indication of whether objects of empty Transactions will be saved: "empty TRs saved", or not: "empty TRs not saved"
- Number of Data Entities + End Points
-> renamed from "Number of DF records / TR end points" to avoid acronyms
- Number of non-empty + empty Transactions : N (= non-empty: n1 + empty: n2)
-> replaces values formerly displayed in "Number of transactions" and in "Number of empty / estimated transactions" (or one of its 2 other possibilities "Number of empty transactions" and "Number of estimated transactions" reported depending upon the "Default FP value for empty Transactional Functions" computation setting was, respectively, "0" for all applications or different from "0" for all applications whose Function Points were computed). As of now, only the counts of empty and non-empty Transactions are reported (N is always equal to n1 + n2), another line in the report being in charge of displaying, among all empty Transactions, the count of only those which are estimated
- Number of links -> Data Entities + End Points : N (= DE+EP: n1 + intermediate objects: n2)
-> replaces the formerly displayed "Number of links -> DF records / TR end points", with values now indicating the counts of each of the sorts of objects composing Transactional Functions AFP graphs (N is always equal to n1 + n2):
 - the number of data entities (aka data function records), included in the count for n1

- the number of transaction end points, also included in the count for n1
- the number of objects on paths between transaction entry points (included in the count for n2) and the deepest data entities or end points (excluded from the count for n2) existing on these paths
- Number of links -> other objects in full graphs : N (includes K links in empty Transactions)
-> since objects of empty Transactions can now be saved on option, the mention "(includes K links in empty Transactions)" makes explicit that such K objects are part of the reported count N
- Number of objects in the largest non-empty Transaction
- Number of objects in the largest empty Transaction
-> these two lines replace the formerly displayed "Number of objects in largest transaction", ambiguous because it wasn't explicitly stating that empty Transactions were not taken into account in the computing of the reported maximum value even though empty Transactions could exist and have their objects be saved for applications whose Enhancement measure was "AEP"
- Saved reduced call graphs of transactions in application #ID (N details)
- Saved full call graphs of transactions in application #ID (N details)
-> these messages indicate that objects belonging to the Transactions of the application whose ID is reported have been saved. The overall number of objects that have been saved is given by N, which includes objects belonging to empty Transactions if their objects have to be saved as well. Messages will start with "Saved reduced call graphs of ..." for applications whose Enhancement measure is EFP or if Function Points are computed using the CAST Transaction Configuration Center, whereas messages will start with "Saved full call graphs of ..." for applications whose Enhancement measure is AEP and Function Points are computed during a Snapshot.

New lines:

- Number of empty Transactions with estimated FP value
-> reports the number of empty Transactions whose FP value will be estimated. Estimation will happen if the "Default FP value for empty Transactional Functions" computation setting has been changed from its default value "0" to another value
- Number of elements allowed in SCC groups
- Number of elements in largest used SCC group
- Number of ignored SCC groups (too large)
-> these lines used to be displayed only if at least one SCC group (Strongly Connected Component grouping many objects all having strongly interconnected dependencies) was containing more than 1000 objects - or the number of objects defined in the SYS_SITE_OPTIONS table by the GRAPH_MAX_SCC_ELEMENTS option - for warning about the existence of some parts of the Transactions graphs having been wiped out from the calculations because of too large SCCs (aka "LSCC" or "big potatoes"). As of now, the default or the overridden maximum number of objects allowed in SCC groups is always reported (this value possibly being "unlimited" if the GRAPH_MAX_SCC_ELEMENTS option has been configured suitably). Still, the value reported as the "Number of ignored SCC groups (too large)" is the indicator of whether none or some too large SCC groups have been excluded from the computation of the Transactions graphs, and therefore as well from the objects that will be accounted as being part of the affected Transactions.

Assessing Function Points for empty Transactional Functions based on non-empty Transactional Functions

It is now possible to configure the CAST Transaction Configuration Center to assess Function Points for all Transactional Functions for which the call graph does not reach a data entity nor an end-point based on existing Function Points that have been computed.

CAST Transaction Configuration Center determines the average number of nodes in the call graph of non-empty Transactional Functions corresponding to one DET. This ratio allows CAST Transaction Configuration Center to assign the number of DET to empty Transactional Functions, depending on the number of nodes in their associated call graph. The Function Point value will be then assessed based on that number of DET and with an FTR=1.

Computing settings

A new value (**ASSESS**) is available to assess the Function Point value in the **Compute** settings:


The screenshot shows a configuration window with a label "Default FP value for empty Transactional Functions:". To the right of the label is a dropdown menu. The dropdown menu is currently open, displaying a list of options: 0, 3, 4, 5, 6, 7, and ASSESS. The "ASSESS" option is highlighted in blue at the bottom of the list. The dropdown menu has a small downward arrow on its top right corner.

 It is not possible to set ASSESS when **Save objects of empty Transactions** is set to NEVER:

Save objects of empty Transactions (those that do not reach any Data Entity or End Point): NEVER ▾

FP value for empty Transactional Functions: ASSESS ▾

Wrong Combination! ✕

 It is not possible to assess the FP value for empty Transactions when the objects of empty Transactions are not saved!

OK

Results

AFP Calibration

When computing Function Points with the parameter set to ASSESS empty transactional functions will no longer exist. Their number and Function Point counts are displayed in the [AFP Calibration page](#) (see the lines in blue):

AFP Calibration

Automated Function Points: 1700

Transactional Functions		Data Functions	
Number of Functions	293	Number of Functions	75
Non-empty Functions	156 (53%)		
Assessed Empty Functions	137 (47%)		
Computed FP	1318	Computed FP	382
Calibrated FP	1318	Calibrated FP	382
Calculated FP	770 (58%)		
Assessed FP	548 (42%)		
Average Nb of objects per DET	12.31		

✳️ Compute function points and load results

The average number of objects per DET ratio is computed on the number of object details in the non-empty transactional functions call graph and the corresponding DET. This ratio is then applied to empty transactional functions to determine their DET. A tooltip is available on blue lines:

Assessed FP **639 (49%)**

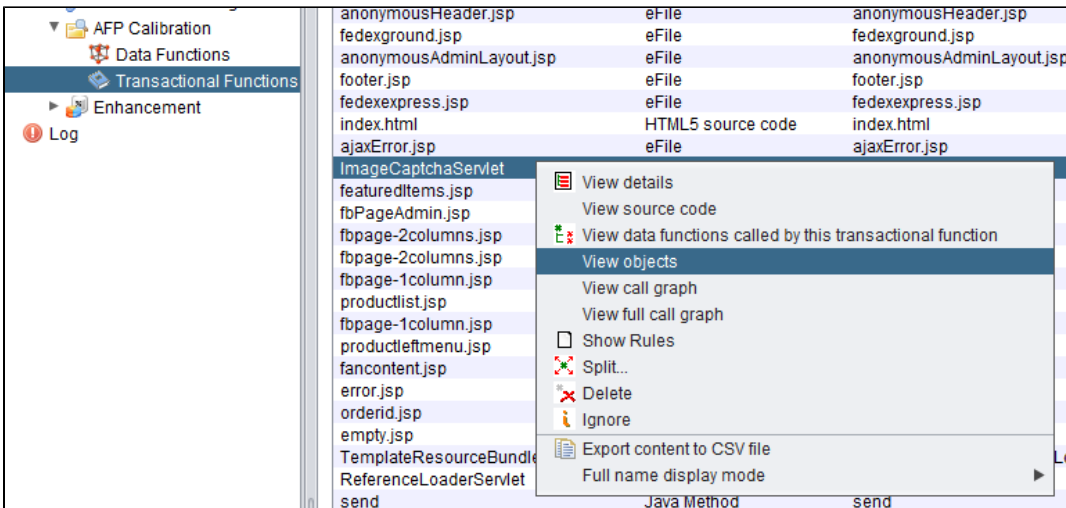
[Empty Transactions] FP is assessed based on the average of nodes per DET of non empty transactions

Assessed Empty Functions **137 (51%)**

[Empty Transactions] Transactions with FP assessed based on the average of nodes per DET of non empty transactions

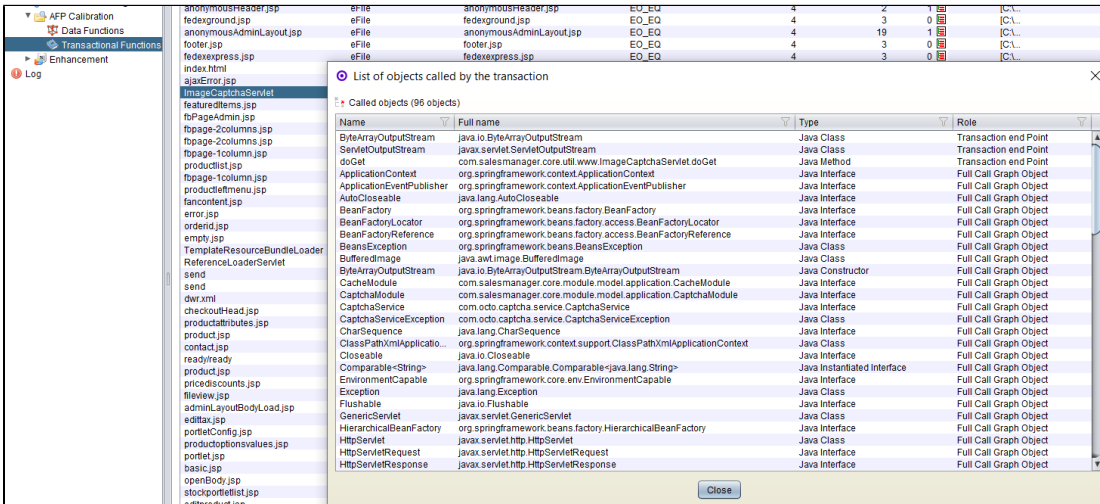
Transactional functions

The [Transactional functions panel](#) will distinguish transactional functions with an "assessed" Function Point from the others by the icon displayed in the **Default FP** column:



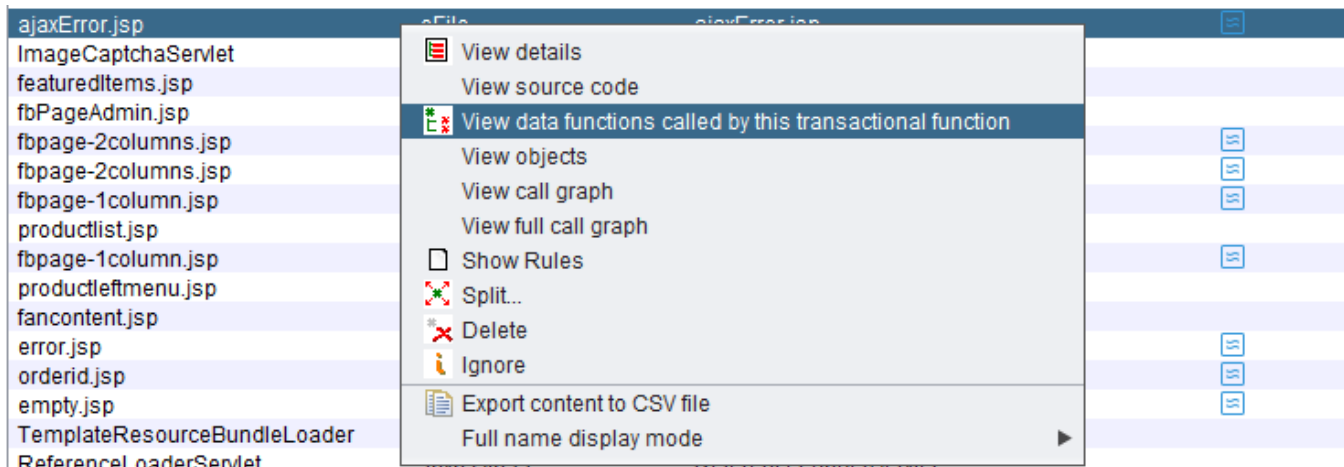
This will open a dialog box with the list of objects called by the associated transaction. This list contains all objects called by the transaction with the name, fullname, object type, and the role in the transaction graph:

Click to enlarge

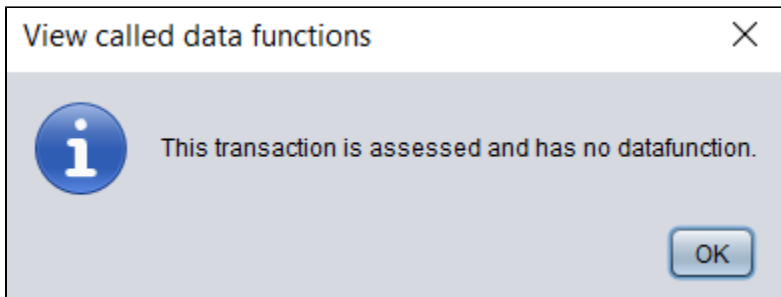


View the list of Data Functions of an assessed transactional function

An assessed transactional function is originally an empty transactional function, which has no data function. When using the **View datafunctions called by this transactional function** right click menu option in **AFP Calibration > Transactional functions**:



a message is displayed to show that the **transaction is assessed and has no data function**:



Items that are considered datafunctions of a transaction

Previous releases of CAST Transaction Configuration Center did not recognize certain object types to be datafunctions of a given transaction. This has now been updated and the following objects are correctly recognized as datafunctions of a given transaction:

- COBOL file
- COBOL database
- COBOL database segment
- Data functions defined by type inheritance of category "APM IFPUG Data Function"