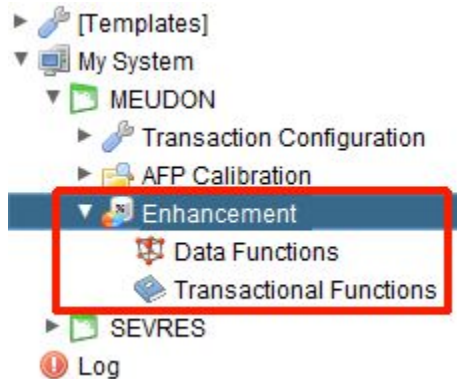


TCC - Enhancement node - Right hand panel

Enhancement node - Right hand panel

The **Enhancement node** has been designed in response to a specific issue. Please read below to find out more information.



- [Background](#)
- [Enhancement feature](#)
- [Measures used to calculate enhancement Function Points](#)
- [Enhancement right hand panel](#)
- [Application Functional Size](#)
- [Ratios](#)
- [Data Functions/Transactional Functions sub nodes](#)

Background

In previous versions of CAST AIP, when examining Function Point data between successive snapshots it became clear that a large quantity of Function Points were being flagged as "modified" with a discrepancy compared to the actual number of artifacts that were flagged as modified. As such, this was preventing users from gathering reliable information about what had really changed between two snapshots. The Enhancement feature is a direct response to this issue.

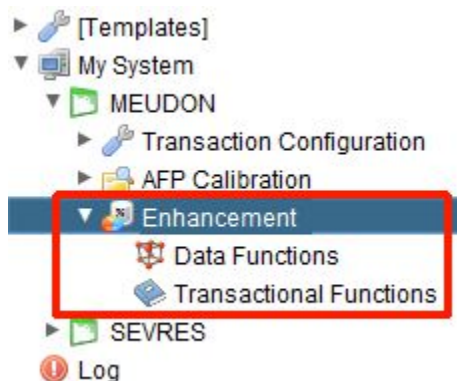
The concept of items (Data Functions/Transactional Functions) that have been added and deleted between snapshots is relatively easy to comprehend, however, an item that is flagged as modified is more complex: in basic terms an item is flagged as modified when its checksum value is deemed to have changed since the previous snapshot. However, an item has a checksum value that is the sum of the checksum values for ALL of the objects that form the original item:

- the initial object (the Data Function/Transactional Function)
- the data function record objects that are used by the entry-point object (tables and persistence objects described as data entities or end-points)
- all objects that are located on the [call graph](#) between the entry and end-point objects.

If any of these objects change between successive snapshots, then the overall item checksum value will change and the item will be flagged as modified. For example, an added or removed comma or white space would cause a change in the checksum value. Modified Function Points are the sum of the Function Point values for each modified item - as such it was not uncommon to find that changing something very trivial in one object that was shared between multiple items caused a large number of items to be flagged as modified between snapshots, thus making the Modified Function Point value also very high.

Enhancement feature

In essence the Enhancement feature provides a detailed presentation of both Data Functions / Transactional Functions in separate nodes:



Each node contains a list of snapshots (**Upper left panel**) and a list of all objects present in the selected snapshot (**Upper right panel**) allowing you to understand why certain items have been marked as modified/added/deleted between snapshots. The Enhancement feature will identify all the objects that form each item (displayed in the **Lower panel**) and between how many items each object is shared (only available for Transactional Functions: a Transactional Function can belong to more than one "item" - see the **Shared Counter** option below). Once the object identification is complete you can then decide whether an object or objects should be **included** or **excluded** in the checksum value calculation for all items (Data Functions/Transactional Functions). When you have included or excluded an object from the checksum value calculation, you can then re-calculate (using the **Recompute checksums and snapshot statuses** option) all checksums for all items and all snapshots. The metrics (in particular the modified function points metric) are then also re-calculated based on the new checksum values and will be redisplayed in the **Upper left panel** allowing you to check whether the value is more in line with expectations.

Measures used to calculate enhancement Function Points

CAST AIP uses two distinct measures to calculate enhancement Functions Points - these can be chosen when you generate a snapshot with the CAST Management Studio/AIP Console:

Automated Enhancement Points (AEP)

Note that this measure is used **by default** from **CAST AIP 8.2.x** onwards.

CAST integrates the estimation of Automated Enhancement Points (AEP). Like EFP, the feature helps to evaluate overtime the size of evolutions implemented by teams producing applications. It computes and displays data functions and transactional functions that have been modified, added, or deleted, and information on the technical components of applications that do not belong to any transaction. The unadjusted function points that are calculated are weighted with Complexity Factors based on the objects complexity.

The AEP measure considers both the functional and the technical sides of the application enhancement. Therefore, the computation of Automated Enhancement Points (AEP) is divided in two sections: Automated Enhancement Function Points (**AEFP**) and Automated Enhancement Technical Points (**AETP**). The total AEP will be the addition of these two: **AEP = AEFP + AETP**.

AEFP Computation

The total value of Automated Enhancement Function Points (AEFP) is calculated by adding the AEFP value of all Transactional Functions and Data Functions in the application. The AEFP of each individual Transactional Function / Data Function is calculated by multiplying its FP value by its Complexity Factor ($AEFP = FP \times CF$). The Complexity Factor of a certain Transactional Function / Data Function is an adjustment factor which is calculated based on its status (added / modified / deleted) and the complexity of the objects inside the Transactional Function / Data Function.

NOTE: the value reported for AEFP includes only added, modified and deleted Functions. All unchanged Functions are automatically excluded from this value (their Complexity Factor is considered as 0, and thus their AEFP value will also be 0).

AETP Computation

Automated Enhancement Technical Points calculation considers all technical objects in the application, i.e. those computational elements which are not the Automated Enhancement Function Point scope.

Enhancement Function Points (EFP)

In addition to AEP, CAST also integrates the estimation of Enhancement Function Points (EFP) also called Project Function Points, so that CAST provides both baselining, e.g sizing an entire application using Automated Function Points and Enhancement Function Points. With the Enhancement Function Points feature, CAST AIP is in a position to display detailed information about the changes between two versions of an application and between two snapshots of the same application.

This feature helps to evaluate overtime the efficiency of teams producing applications. It is particularly useful in outsourcing contexts as many outsourcing contracts are based on FP when it comes to determine the size of the deliverables. This feature computes and displays data functions & transactional functions that have been modified, added or deleted. For each application, CAST AIP will provide 3 different pieces of information:

1. **Added FPs:** FPs identified from the functions identified as created in the Vn+1 (i.e. that is added by the enhancement project)
2. **Modified FPs:** FPs counted because existing function is changed during the enhancement project
3. **Deleted FPs:** function points counted for function deleted during the enhancement project.

Differences between the two measures

The main differences between the legacy Enhancement Function Points (EFP) mode and the Automated Enhancement Points (AEP) mode (introduced in CAST AIP 8.2) are as follows:

- EFP does not consider the technical side of the application evolution; AEP does (calculating Automated Enhancement Technical Points, AETP)
- EFP uses a transaction **reduced call graph** (from Entry Points to End Points / Data Entities); whereas AEP considers the **transaction full call graph** (it covers also the paths that do not reach an End Point / Data Entity). This means that the number of objects in the transaction call graph for AEP is higher.
- The adjustment factor used to calculate Enhancement Function Points (EFP) is called "**Impact Factor**", and it is based on fixed user-defined formulas; on the contrary, the adjustment factor to calculate Automated Enhancement Function Points is called "**Complexity Factor**", and it is based on dynamic formulas.

The table below summarizes these differences:

Mode	Transaction Call Graph	Adjustment Factor	Evolution
EFP	Reduced	Impact Factor (fixed user-defined formula)	Only Functional
AEP	Full	Complexity Factor (dynamic formula)	Functional and Technical

Note about switching between measurement modes

When changing the measurement mode it is important to take into consideration the impact this will have on the analysis/snapshot results and that it will introduce a disruption into the measurement trending. There are important differences between the two measures (AEP and EFP) that must be understood:

- The EFP mode does not manage the technical part of application whereas AEP does.
- When using EFP mode, Transactional Functions are based on the reduced call graph whereas for AEP, Transactional Functions are based on the full call graph.
- The AEP specification defines Complexity Factors based on artifacts' complexity or changes made to data entities.

When moving from EFP to AEP to get more precise values, results will show objects that do not belong to the reduced transaction call graph as "added" and this will impact the measure, even if these objects have not really been added. Points related to the technical part of the Application will also appear in the results. Moreover, taking in to account object complexity to weight the FP values will impact the results as well.

Moving in the opposite direction from AEP to EFP will result in having objects that do not belong to the full transaction call graph to be considered as "deleted". This will also impact the results, even if those objects have not been really removed. In addition, Complexity Factors will be replaced by Impact Factors that are set to 1 by default and points related to the technical part of the application will disappear.

If you plan to change the measurement mode, it is necessary to recreate a new baseline snapshot. To do that, after you change the measure you must generate a new snapshot for the same Application version but it must not be considered as consistent for the new measure. The next snapshot you generate for the Application will be consistent and can be used as the first snapshot using the new enhancement measure.


Enhancement right hand panel

The data in this panel is automatically fetched from the database server and displayed when selecting the node. The data presented here represents the "evolution" of the Function and Technical Points between successive snapshots.

By default the data for the most recent snapshot will be displayed when the node is selected. If there is only one snapshot present for the selected Application, then data will show Added Function Points for both Transactional Functions and Data Functions. When more than one snapshot is present, the data show is a comparison between the current and previous snapshot. In this case, you may see Modified and Deleted Function Point data as well as Added.

Display with EFP measure

Application Enhancement

 Snapshot information

< Prev.

Next >

Enhancement Measure: EFP

Name: Computed on 201702061651

Date: 2017-02-06 16:52:44.251

Functional date: 2017-02-06 01:00:00.0

Enhancement Function Points: 70

Transactional Functions

Total EFP:	70
Added EFP:	0 (0%)
Modified EFP:	70 (100%)
Deleted EFP:	0 (0%)

Data Functions

Total EFP:	0
Added EFP:	0
Modified EFP:	0
Deleted EFP:	0

Display with AEP measure

Application Enhancement

Snapshot information

< Prev.

Next >

Enhancement Measure: AEP

Name: Computed on 201711131513

Date: 2017-11-13 15:16:35.394

Functional date: 2017-11-13 00:00:00.0

Automated Enhancement Points: 253

Display AEP Detail

Automated Enhancement Function Points: 72

 Transactional Functions

Total AAFP: 5
Added AAFP: 0 (0%)
Modified AAFP: 5 (100%)
Deleted AAFP: 0 (0%)

 Data Functions

Total AAFP: 67
Added AAFP: 67 (100%)
Modified AAFP: 0 (0%)
Deleted AAFP: 0 (0%)

Automated Enhancement Technical Points: 181

Added AETP: 181 (100%)
Modified AETP: 0 (0%)
Deleted AETP: 0 (0%)

Export List of Objects

Note about Automated Enhancement Technical Points (AETP)

AETP is the count of added / modified / deleted technical points, which are calculated based on the added / modified / deleted technical objects (i.e. objects which are not part of any transaction). In other words, AETP summarize the evolution performed in the application, but outside the functional scope (these objects are not taken into account in the functional points, so the new AEP measure introduces the concept of "technical points" to assess the enhancement done in these types of objects).

AETP data is not available when the EFP measure has been used to calculate the snapshot.

From CAST AIP 8.3.26, in order to provide greater accuracy, the calculation of AETP values has been modified. Previously, all added/deleted/updated AETP detail values between 0 and 1 were calculated with no decimal places, effectively giving the impression in some circumstances (when all added /deleted/updated values were below 1) that total AETP = 0. This behaviour has been changed and AETP detail values are now considered to two decimal places for added/deleted/updated. In addition AETP total values will now be **rounded up** as follows:

Real value	New rounded up value
0	0
0.8	1
0.5	1
0.2	1
1	1
1.8	2
1.5	2
1.2	2

Prev. / Next buttons

It is possible to see data from a previous/next snapshot by clicking the **Previous** or **Next** buttons - data in the panel will refresh automatically.

Display AEP Detail

The formula used to generate AEP information are complex and use a variety of intermediary metrics. To help understand how AEP is generated, clicking the **Display AEP Detail** button will display these intermediary metrics and formula in a text file. You can find out more about this text file in [Enhancement node - AEP Details](#).

```
report AEP My System - MEUDON.txt X
1 Detail of AEP metrics for application : "MEUDON" - snapshot : Computed on 201711131513
2
3
4 =====
5 I. FUNCTIONAL PART
6 =====
7
8 I.a TRANSACTIONAL FUNCTIONS
9 =====
10 ----- Definitions and Formulas -----
11 | AEPF      Transactional Functions = sum(round(CF x AFP)) (*)
12 | Added     Transactional Functions : CF = 1
13 | Deleted   Transactional Functions : CF = 0.4
14 | Modified  Transactional Functions : CF = see below
15
16 | CF is defined by the following matrix ,
17 |   if less than 50% of the modified Artifacts are shared components :
18 | -----
19 | | Evolved EC | EC Variation | | | | |
20 | |-----|-----| |-----|-----|
21 | |           | <= 1/3 (x 100%) | <= 2/3 (x 100%) | <= 100% | > 100% |
22 | |-----|-----| |-----|-----|
23 | | <= 1/3 x 100% | 0.25 | 0.50 | 0.75 | 1.00 |
24 | |-----|-----| |-----|-----|
25 | | <= 2/3 x 100% | 0.50 | 0.75 | 1.00 | 1.25 |
26 | |-----|-----| |-----|-----|
27 | | <= 100% | 0.75 | 1.00 | 1.25 | 1.50 |
28 | |-----|-----| |-----|-----|
29 | | > 100% | 1.00 | 1.25 | 1.50 | 1.75 |
30 | -----
31 | CF values for transactions that did contain shared Artifacts that were
32 | changed are determined as follows:
33 | a. If 100% of the modified Artifacts are shared , CF = 0.25
34 | b. If > 75% of the modified Artifacts are shared , CF is capped at 0.50
35 | c. If > 50% of the modified Artifacts are shared , CF is capped at 0.75
36 | -----
37
```

Export List of Objects

This button can be used to export the list of technical objects that are involved in the Automated Enhancement Technical Points (AETP) calculation for a given snapshot (note that this feature will be available only for the snapshots computed using AEP measure). The output CSV file will contain the following values:

- **OBJECT_ID**: Object ID
- **OBJECT_NAME**: Name
- **OBJECT_FULLNAME**: Full name
- **OBJECT_TYPE_STR**: Type of object
- **STATUS**: Status in the current snapshot
- **EFFORT_COMPLEXITY**: Effort Complexity
- **ORIGIN**:
 - **RULE**: Objects that have been captured through a set
 - **MANUAL**: Objects that have been manually excluded

For example:

OBJECT_ID	OBJECT_NAME	OBJECT_FULLNAME	OBJECT_TYPE_STR	STATUS	EFFORT_COMPLEXITY	ORIGIN
252	BookManager	com.castsoftware.bookdemo.javabean.BookM	Java Method	UNCHANGED	0.05	-
256	checked	com.castsoftware.util.gui.GuiHtml.checked	Java Method	UNCHANGED	0.05	-
257	OracleConnection	com.castsoftware.dbconnection.OracleConnec	Java Method	UNCHANGED	0.05	-
262	I_Order_Line	default.CASTPUBS.dbo.I_Order_Line	Microsoft procedure	UNCHANGED	0.2	-
275	on_delete_authors	default.CASTPUBS.dbo.on_delete_authors	Microsoft DML trigger	UNCHANGED	0.1	MANUAL
284	GuiHtml	com.castsoftware.util.gui.GuiHtml.GuiHtml	Java Constructor	UNCHANGED	0.05	-

Application Functional Size

Application Functional Size

Automated Function Points: 131



Transactional Functions

Total AFP: 51



Data Functions

Total AFP: 80

This section shows the Automated Function Points values (for Transactional Functions, for Data Functions and total) of the application when the displayed snapshot was taken.

Warning: these values are not necessarily the same as the ones displayed in the [AFP Calibration node](#). This node displays the AFP values at the present time, and they can suffer variations if you perform calibration actions and click on the **Compute Function Points** button. On the contrary, the values displayed in the Enhancement node are the definite AFP values that were computed for the application at the moment when a snapshot was executed. These values cannot be changed.

Ratios

Ratios

Number of Artifacts:	63	Percentage of Artifacts Involved in TF:	33%
Number of classes/programs:	0	Average AFP per class/program:	0.00
Number of Objects:	146	Percentage of Objects Involved in TF:	18%
TFP / DFP :	0.64		

This section provides information about the elements that comprise the current Applications and snapshot:

Number of Artifacts	Total number of artifacts in the Application.
Number of classes/programs	The number of main objects (classes and/or programs) in the Application.
Number of Objects	The number of all objects in the Application.
TFP/DFP	This is the ratio between Transactional Function points and Data-Function points.
Percentage of artifacts involved in TF	The ratio of the distinct number of artifacts in all Transactional Functions in the application compared to the total number of artifacts in the Application.
Average AFP per class/program	This is to see if the size of the functional part (AFP) of the Application is relevant or not compared to the number of main objects (classes and/or programs).
Percentage of objects involved in TF	The percentage of the objects in the Application involved in Transactional Functions.

Data Functions/Transactional Functions sub nodes

See [Enhancement - Data Functions/Transactional Functions sub-nodes - Right hand panel](#) for more information.

