

CAST Engineering Dashboard - Metrics - Number of metrics - Unexpected increase or decrease

Purpose (problem description)

In this case, the idea is to retrieve and identify objects generating the difference for the **Number of** between the 2 snapshots and compare them at the module level for both snapshots.

If the comparison is at application or system level, compare the **Number of** for each module belonging to the application/system. That's why you need to get a copy of each Knowledge base involved in the two snapshot computations.

Observed in CAST AIP

Release	Yes/No
8.3.x	✓

Observed on RDBMS

RDBMS	Yes/No
Oracle Server	✓
Microsoft SQL Server	✓
CSS	✓

Action Plan

The idea here is to identify the object(s) leading to the difference and identify the reason generating these objects. to do this please proceed as follow

1. [241909854](#)
2. [Get the customer's input](#)
3. For each snapshot, [#Get the modules names](#)
4. For each snapshot, [#Get the Number of per module shown in ADG Dashboard](#)
5. Identify the module(s) having different value of "Number of" in the 2 snapshots.
6. For the module(s) identified in step 5,
 - a. [#Calculate the Number of per module in the Knowledge Base](#) using the knowledge base involved for each snapshot computation
 - b. get the list of objects used to compute the "Number of" per module in the Knowledge Base using the query in [#Calculate the Number of per module in the Knowledge Base](#).
7. Identify the difference between the 2 lists of objects used to compute the number of at module level
8. Check if the additional/missing objects between the 2 snapshots justify the difference identified at **Number of** count in the dashboard
9. Identify the reason generating the difference (additional/missing objects)
 - a. If the reason leading to the difference is related to modified code source, modified job configuration, give customer the reason leading to this difference
 - b. If you cannot identify the reason, [#Transmit the problem to the R&D team](#)

Get the Customer's Input

1. Ask the customer for his expectations and how he justifies them.
2. One of the following screenshot (depending on the problem):
 - a. The **Outline** page (FRAME_PORTAL_TOP_LEVEL_DETAILS). Note that the screenshot should show the **Focus on** section and the url address, as in the above example. This is compulsory to see the whole **Focus on** section in the screenshot, since we require additional information other than the **Number of** value. If the screenshot is not as expected, ask the customer to produce it correctly.
 - b. The **Version Comparison** page (FRAME_PORTAL_CONTENT_EVOLUTION)
 - c. The **APPLICATION view** (FRAME_PORTAL_RISK_VIEW)
3. Using the screenshot, check the level at which the metric is calculated: System, Application, Module.
 - a. In the Outline page, read the **Focus on** and **Quantity of** sections.
 - b. In the **Version Comparison** page, read the content of the context field : in this page we can see only the name of the page without knowing if its System, Application or module. Please,
 - c. The **APPLICATION view**, read the name of the application
4. For the case of the **Version Comparison** page, ask customer if the comparison context correspond to a system, application or module
5. Check the content of the identified system, application via the quick access page for example:
 - a. If the **Number of** is calculated at the System level and the system contains only one application with only one module: the central is not useful, we just need the name of the module. Otherwise, provide the dump of the central site
 - b. If the **Number of** is calculated at the Application level and if the application contains only one module: the central is not useful, we just need the name of the module. Otherwise, provide the dump of the central site
6. Knowledge base involved for each snapshot computation.

Verify the Metric (specific cases)

Number of PL/SQL artifacts for offline analyzer:

If you are migrating from 7.0.x (x<10) and 7.3.x, you may observe on the dashboard a decrease of the number of PL/SQL artifacts (in the Technical Size category) even though the source code has not changed. This is explained by the functional modification of the counting process of the number of artifacts meantime. Between CAST AIP 7.0.9 and 7.0.10, a change was introduced in the offline PL/SQL Analyzer. Previously, stored procedures and functions declared in a package were counted twice by CAST AIP:

- Once in the package body
- Once in the package header

After a fix introduced in CAST AIP 7.0.10, stored procedures and functions declared in packages are only counted once: in the package body.

Therefore when migrating from CAST AIP 7.0.x (x<10) to CAST AIP 7.3.x and after running a snapshot in CAST AIP 7.3.x, results may show a decrease in the number of PL/SQL artifacts.

Number of Code Lines:

[For number of code lines please refer to Number of Line Of Code - Troubleshooting](#)

Number of SQL Artifacts :

[For number of SQL Artifacts refer to Number of SQL Artifacts](#)

Number of Critical Violations :

[CAST Engineering Dashboard - Violations - Variation in Critical Violations value](#)

Get the Number of per module shown in the AD Dashboard

If you have central repository,

1. Connect to the dashboard and get the Number of for each module belonging to the system/application. You can get this value in the page 'Focus on Module <Module_Name> - part of <APPLICATION_NAME>- for' of the dashboard as following one
If the **Number of** is calculated at module level, read the **Number of** value in the screenshot received from the customer.
2. Identify the module(s) generating the difference:
Example If you have one application MyAppli with 2 modules: Module1 and Module2
 - The number of Files for MyAppli is N in snapshot 1 and M in the snapshot 2
 - The number of Files for Module1 is P in snapshot 1 and P in the snapshot 2
 - The number of Files for Module2 is Q in snapshot 1 and R in the snapshot 2 with P is different to QHere we can conclude that difference of Number of Files observed at application level is generated by the difference on Number of Files at level of Module2. In this case, we must identify the reason of this difference as described below.

If you have not the central repository,

Number of at module level = Number of at application level = Number of at system level
you can read the information in the provided screenshot.

Calculate the Number of per module in the Knowledge Base

PLEASE NOTE: THE QUERIES BELOW ARE ONLY VALID IN CAST AIP 8.3.36 AND BELOW. FOR CAST AIP 8.3.37 AND ABOVE YOU MUST MODIFY THE QUERIES AS FOLLOWS:

- change references:
 - PROPERTIES = 0
- to
 - PROPERTIES & 255 = 0

- change references:
 - c.PROPERTIES = 0
- to
 - c.PROPERTIES & 255 = 0

This section allows you to get the Value of th **Number of** for each module in the knowledge base the detail of this value by providing the **list of objects** used to compute this value

- If the module is the result of an analysis job, the **Number of** is the number of the application's non-external objects with a type inheriting from the APM category.
- If the module is a User Defined module, the **Number of** is the number of the User Defined module's non-external objects with a type inheriting from the APM category.

For each module, proceed as follow:

1. Check whether the module is the result of an analysis job or a user defined module:
 - a. For the module resulting from an analysis job, the module name will be followed by the Analyzer type such as 'Leben_C_Module-C++ Analyzer' and 'EAR_MAXNET_WEB_MAXNET-J2EE Analyzer'
 - b. For the user defined module, the module name will not be followed by the Analyzer type as it can be based on the results of many analyses.
2. If the module is the result of an analysis job:
 - a. Get the Application's Id associated with the analysis job (module on the central) using following query: Select * From <KB_name>.CSV_ANALYSIS_APPLICATIONS where APPLICATION_NAME like <MODULE_NAME>
 - b. Get the application's non-external objects list: Select * from <KB_name>.CTT_OBJECT_APPLICATIONS where APPLICATION_ID=< Application's Id > and PROPERTIES=0
 - c. Get vaue of the **Number Of** and **list of objects** used to compute the Number of
 - i. Query for Number of Comments:
To get the number of comment lines of an application's non-external objects, run following query: Select sum(INFVAL) From <KB>.OBJINF where IDOBJ in (select (OBJECT_ID) from <KB>.CTT_OBJECT_APPLICATIONS where APPLICATION_ID=<APPLICATION_ID> and PROPERTIES=0) and INFTYP=2 and INFSUBTYP=0
 - ii. Query for Number of Commented-out code lines:
To get the number of commented-out lines of an application's non-external objects, run following query: Select sum(INFVAL) From <KB>.OBJINF where IDOBJ in (select (OBJECT_ID) from <KB>.CTT_OBJECT_APPLICATIONS where APPLICATION_ID=<APPLICATION_ID> and PROPERTIES=0) and INFTYP=9 and INFSUBTYP=1509
 - iii. Query for Number of Comment lines - Other: Number of comment lines-Other of an application's non-external objects = Number of **comment lines** of an application's non-external objects minus the Number of **commented-out** lines of an application's non-external objects
 - iv. Query for other Number Of (Number of tables, classes, methods....)
 1. Get the CATEGORY_NAME value, used in the below queries, by searching it in the assesment model- take the category mentioned in the Detail SQL procedure of the diag(For instance, CATEGORY_NAME = 'DataBase Table' for Number of Tables).
 2. **Count the Number of the** objects with a type inheriting from the identified category: Select count(K1.IDKEY) From KEYS K1, KEYS K2 where K1.IDKEY in (select (OBJECT_ID) from CTT_OBJECT_APPLICATIONS where APPLICATION_ID==<Application_ID> and PROPERTIES=0) and K1.IDKEY=K2.IDKEY and K2.OBJTYP in (select distinct t.IdTyp from Typ t, TypCat tc where t.IdTyp = tc.IdTyp and tc.IdCatParent in (select Cat.IdCat from Cat where CatNam like '<CATEGORY_NAME>'))
 3. Get **list of the objects** with a type inheriting from the category under investigation: Select K1.* From KEYS K1, KEYS K2 where K1.IDKEY in (select (OBJECT_ID) from CTT_OBJECT_APPLICATIONS where APPLICATION_ID=<Application_ID> and PROPERTIES=0) and K1.IDKEY=K2.IDKEY and K2.OBJTYP in (select distinct t.IdTyp from Typ t, TypCat tc where t.IdTyp = tc.IdTyp and tc.IdCatParent in (select Cat.IdCat from Cat where CatNam like '<CATEGORY_NAME>'))

1. If the module is a User Defined module:

- a. Get the User Defined module's objects list: `SELECT UDM.KeyNam, (SELECT ofn.IDOBJ FROM <KB>.ObjFulNam ofn WHERE ofn.IdObj=UDMi.IdKey), (SELECT ots.ObjTypStr FROM <KB>.ObjTypStr ots WHERE ots.ObjTyp=UDMi.ObjTyp) FROM <KB>.APPSET s, <KB>.Keys UDM, <KB>.AnaJob j, <KB>.SETROOT r, <KB>.OBJSET o, <KB>.Keys UDMi WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>'`
- b. Get the User Defined module's non-external objects list: `SELECT UDM.KeyNam,(SELECT ofn.IDOBJ FROM <KB>.ObjFulNam ofn, WHERE ofn.IdObj=UDMi.IdKey),(SELECT ots.ObjTypStr FROM <KB>.ObjTypStr ots WHERE ots.ObjTyp=UDMi.ObjTyp) FROM <KB>.APPSET s, <KB>.Keys UDM, <KB>.AnaJob j, <KB>.SETROOT r, <KB>.OBJSET o, <KB>.Keys UDMi, <KB>.CTT_OBJECT_APPLICATIONS c WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>' AND UDMi.IdKey= c.OBJECT_ID and c.PROPERTIES=0`
- c. **Get Number of**
 - i. **Query for Number of Comments** `Select sum (INFPAL) From <KB>.OBJINF where IDOBJ in (SELECT (SELECT ofn.IDOBJ FROM <KB>.ObjFulNam ofn, WHERE ofn.IdObj=UDMi.IdKey) FROM <KB>.APPSET s, <KB>.Keys UDM, <KB>.AnaJob j, <KB>.SETROOT r, <KB>.OBJSET o, <KB>.Keys UDMi, <KB>.CTT_OBJECT_APPLICATIONS c WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>' AND UDMi.IdKey= c.OBJECT_ID and c.PROPERTIES=0 and INFTYP=2 and INFSUBTYP=0`
 - ii. **Query for Number of Commented-out code lines** `Select sum(INFPAL) From <KB>.OBJINF where IDOBJ in (SELECT (SELECT ofn.IDOBJ FROM <KB>.ObjFulNam ofn, WHERE ofn.IdObj=UDMi.IdKey) FROM <KB>.APPSET s, <KB>.Keys UDM, <KB>.AnaJob j, <KB>.SETROOT r, <KB>.OBJSET o, <KB>.Keys UDMi, <KB>.CTT_OBJECT_APPLICATIONS c WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>' AND UDMi.IdKey= c.OBJECT_ID and c.PROPERTIES=0) and INFTYP=9 and INFSUBTYP=1509`
 - iii. **Query for Number of Comment lines - Other Number of comment lines-Other** of an application's non-external objects = **Number of comment lines** of an application's non-external objects minus **Number of commented-out lines** of an application's non-external objects
 - iv. **Query for other Number Of:**

Get the CATEGORY_NAME value, used in the below queries, by searching it in the [metric category names and Ids](#) page (For instance, CATEGORY_NAME = 'DataBase Table' for Number of Tables).

 1. **Count the Number** of objects(non external objects of the User defined module) with a type inheriting from the identified category `SELECT count(UDM.KeyNam) FROM APPSET s, Keys UDM, AnaJob j, SETROOT r, OBJSET o, Keys UDMi WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>' and UDMi.ObjTyp in (select distinct t.IdTyp from Typ t, TypCat tc where t.IdTyp = tc.IdTyp and tc.IdCatParent in (select Cat.IdCat from Cat where CatNam like 'CATEGORY_NAME'))`

The **Number of** at the module is the result of the above query
 2. Get the **list of the objects** with a type inheriting from the identified category, using the following query: `SELECT UDM.KeyNam, (SELECT ofn.IDOBJ FROM <KB>.ObjFulNam ofn, WHERE ofn.IdObj=UDMi.IdKey), (SELECT ots.ObjTypStr FROM <KB>.ObjTypStr ots WHERE ots.ObjTyp=UDMi.ObjTyp) FROM <KB>.APPSET s, <KB>.Keys UDM, <KB>.AnaJob j, <KB_name>.SETROOT r, <KB>.OBJSET o, <KB_name>.Keys UDMi, <KB>.CTT_OBJECT_APPLICATIONS c WHERE s.IDSET = UDM.IdKey AND s.IDJOB = j.IdJob AND j.JobTyp = 3001 AND UDM.ObjTyp = 20000 AND UDM.IdKey = r.IDSET AND o.IDSET = r.IDROOT AND o.IDOBJ != r.IDROOT AND o.IDOBJ = UDMi.IdKey AND UDM.KeyNam='<UDM_NAME>' AND UDMi.IdKey= c.OBJECT_ID and c.PROPERTIES=0 and UDMi. OBJTYP in (select distinct t.IdTyp from Typ t, TypCat tc where t.IdTyp = tc.IdTyp and tc.IdCatParent in (select Cat.IdCat from Cat where CatNam like '<CATEGORY_NAME>'))`

Calculate the Number of at application/system level in the Knowledge Base

This paragraph is relevant only for the following cases:

- One application with more than one module
- One system with more than one application
- One system with one application containing more than one module

The number of at application level = sum of number of at modules level

The number of at system level = sum of number of at applications level

Transmit the problem to the R&D team

If you cannot identify the reason generating the difference (additinnal/missing objects), transmmit to the team in charge of the analyzer

1. The problem description
2. the used KBs
3. the list of objects (additinnal/missing objects) generating the difference between the 2 snapshots

