

# CMS Snapshot Analysis - Run Analyzer - Wrong objects

## Purpose (problem description)

This page handles the issue of wrong object creation after analyzing an application. For information on objects created by the CAST AIP analyzers, refer to the relevant pages for each technology in the official documentation:

- For CAST AIP 8.3: [Covered Technologies](#)
- For CAST AIP 8.2: [Supported Technologies](#)
- For CAST AIP 8.1: [Supported Technologies](#)
- For CAST AIP Extensions: [CAST AIP Extensions documentation](#)

Error Free Logs - [CMS Snapshot Analysis - Run Analysis for Technology - Error Free Logs - CAST AIP 8](#)

## Observed in CAST AIP

Release	Yes/No
8.3.x	✓
8.2.x	✓
8.1.x	✓
8.0.x	✓
7.3.x	✓

## Observed on RDBMS

RDBMS	Yes/No
Oracle Server	✓
Microsoft SQL Server	✓
CSS3	✓
CSS2	✓

## Step by Step scenario

1. Analyze a project
2. Open Enlighten
3. Validate the created objects in Enlighten Graphical view

## Action Plan

Proceed with the below steps

1. Open Enlighten and identify the wrong object as described in [Enlighten - Information - How to identify and open an object in Enlighten Graphical View](#)
2. Validate if object is **EXPECTED** or **NOT EXPECTED**, according to the expected objects created by the CAST AIP analyzers as described in [CMS Snapshot Analysis - Run Analyzer - Information - How to validate expected objects and links in CAST AIP documentation](#)
  - a. If you have confirmed that the object should be created, this is an expected product behavior.
  - b. If the object should not be created or if you cannot confirm the expectation, proceed to next step
3. Validate object expectation from the source code. You can locate the object source code :
  - Either from [ENL - Launching the Code Viewer](#)
  - Either from the project source file that you can retrieve from [object properties](#)
  - For certain technologies (like Mainframe) an Unknown Program object is created when the source code contains a call to a program that is unresolved (not present in the source code). In this case the source code file that should be checked is the one where the call is declared. You can retrieve this file by opening the [linked object](#) to the Unknown Program and looking at the source code of the link
  - a. If you find an exact reference to the object name, then validate the object type in regard to the syntax and reserved keywords of the used technology
    - i. The object is **EXPECTED** when it corresponds to the expected object type
    - ii. The object is **NOT EXPECTED** when it does not correspond to the expected object type
  - b. If you do not find an exact reference to the object name then the object is **NOT EXPECTED** except in some special cases where object names are created as a combination of references in the source code. This is the case for some instances of AngularJS Extension objects or JQuery Extension objects.
4. Check if wrong objects are a result of the existence of ghost projects or ghost objects.
  - a. Run the queries and remediation that can be found in [SQL Queries - CAST Knowledge Base - Queries on objects - Identify and clean up all the ghost objects and ghost projects in your knowledge base](#).
  - b. Check if wrong objects refer to source code that is not present in the current version. These objects are ghost objects that are not detected by the queries in previous step, because they are still attached to projects which themselves are attached to existing analysis units. These projects should be detected and removed. Check in your [deployment folder](#) if the object's source code file exists (you can retrieve it from [ENL - Viewing properties](#))
    - i. If it exists proceed to next step
    - ii. If it does not exist then locate and delete the project from where the objects belongs:
      1. Run the following query in local database

```
select o.idobj, o.FullName from ObjFulNam o
where o.FullName = '%<Object_source_code_fullpath>%'
```

Above query may return several rows, you can use the **idobj** for anyone of them

2. Use any of the above retrieved idobj and run query:

```
select idpro from ObjPro op where op.IdObj = <idobj>
```

3. Take a backup of the local database
4. Delete project by running query, using the retrieved idpro

```
select PROJECT_DELETE(<idpro>)
```

5. Rerun analysis/snapshot

5. Check if wrong objects are the result of the previous snapshot and in the new delivery the source code has been removed via DMT. But the analysis unit is still there and new snapshot has been taken and we could still see the objects in KB from last delivered source code. This issue has been seen in UA analyzer but it can come for another analyzer as well.
  - a. Run the query in the cdt\_objects table to identify the wrong objects coming from the source code path inside the deploy folder.
  - b. Keep an empty file in the root of that path if it is SQL file keep empty SQL file etc.
  - c. Run the analysis
  - d. The objects will be removed from the KB and hence no more objects are seen on the dashboard post snapshot. This issue is under investigation and has to be fixed in a future version, we will update the doc once this issue is fixed.
6. Check if your case matches one of the known cases with solution, described in:
  - [CMS Snapshot Analysis - Run Analyzer - Wrong objects - DOTNet with Core CAST AIP](#)
  - [CMS Snapshot Analysis - Run analyzer - Wrong Objects - DOTNet with extensions](#)
  - [CMS Snapshot Analysis - Run Analyzer - Wrong objects - Mainframe with Core CAST AIP](#)
  - [CMS Snapshot Analysis - Run Analyzer - Wrong objects - Web with extension - HTML5 and JavaScript](#)
7. If you do not find the information you are looking for or solution for your problem, on this page, contact [CAST Technical Support](#), with the [Relevant input](#)

## Relevant input

- [Sherlock](#) export with the following options: **Export Cast Database(Management and Local databases), Export Logs, Export Configuration files, Source code**

## Notes/comments

## Related Pages