

CMS - Working with Execution Units

Working with Execution Units

Execution Units are an expert level feature of the CAST Management Studio that represent how auto-generated Analysis Units (i.e source code and configuration details) are analyzed. Depending on a variety of factors, multiple Analysis Units may be grouped together into one Execution Unit, or one Execution Unit may be created for each Analysis Unit.

Their objectives are as follows:

- to improve analysis **performance** - i.e. by grouping multiple Analysis Units together in one "analysis execution," analysis time can be improved
- to reduce the risk of an **analysis crashing** - an Execution Unit has an upper limit on the amount of source code that can be analyzed in one "analysis execution" (this differs per Technology) and this limit is designed to prevent an analysis crashing due to an out of memory error
- to ensure links between Analysis Units are **not missed**, particularly links determined by the CAST Inference Engine (see individual [Technology editors](#) for more information about the Inference Engine).

How do they work?

The contents of an Execution Unit (i.e. which Analysis Units are grouped together) are determined automatically by the CAST Management Studio using internal algorithms specific to each technology type and information (source code size, dependencies between projects) identified during the source code "discovery" process in the CAST Delivery Manager Tool. For the most part, you should let the CAST Management Studio manage Execution Units - however, it is possible to configure them yourself if you find that the automatic management strategy provided by the CAST Management Studio is not providing an optimum result. This is done by:

- appending an **"annotation"** to an Analysis Unit - all Analysis Units with the same annotation are then analyzed together (for the manual management strategy)
- by modifying the default **upper limit** on the amount of source code that can be analyzed in one "analysis execution" - if the total amount of source code for all Analysis Units exceeds this limit then an additional Execution Unit is created by CAST (for the auto-management strategy).

Auto-management of Execution Units

CAST will automatically manage and define Execution Units on an **Application** basis for each supported Technology as listed in the table below:

VB	Multiple Analysis Units are grouped according to the information discovered by the CAST Delivery Manager Tool in the following order: <ul style="list-style-type: none">• Analysis Units are first grouped in an Execution Unit according to the size of the source code in each Analysis Unit. The total amount of source code in an Execution Unit is determined by the Execution Unit max size value (see below). The maximum possible number of Analysis Units up to the Execution Unit max size value (see below) are placed in one single Execution Unit.• If the total code size for all Analysis Units exceeds the Execution Unit max size (see below), then additional Execution Units will be created. These additional Execution Units are created based on the dependencies between the Analysis Units. In other words, all Analysis Units that depend on one another will be placed in one single Execution Unit where possible.• If there are no dependencies between Analysis Units, Analysis Units will still be grouped in one single Execution Unit. The maximum possible number of Analysis Units up to the Execution Unit max size value (see below) are placed in one single Execution Unit.
Oracle PL/SQL and MS /Sybase T-SQL extraction (offline)	A source code package corresponds to one RDBMS extraction (that could contain multiple schemas and therefore multiple Analysis Units), as such all Analysis Units belonging to the same source code package are analyzed together in one Execution Unit, with no upper source code limit.
ASP	An Analysis Unit corresponds to one ASP website, therefore one Execution Unit is created per Analysis Unit.
C/C++	One Execution Unit is created per Analysis Unit.
Oracle Forms	One Execution Unit is created per Analysis Unit.
Mainframe	One Execution Unit is created per Analysis Unit.
J2EE	One Execution Unit is created for ALL Analysis Units.
.NET	One Execution Unit is created for ALL Analysis Units.
SAP	One Execution Unit is created per Analysis Unit.
Universal	One Execution Unit is created per Analysis Unit.

Note that as soon you add an **Execution Unit annotation**, the auto-management strategy will be deactivated.

Manually creating an Execution Unit

Manually managing and creating Execution Units entails overriding the default rules that the CAST Management Studio uses to group Analysis Units into Execution Units. This is a highly technical feature and should only be used when you know exactly what you want to achieve.

Currently, manually creating an Execution Unit is only possible for the following technologies:

- **VB**

It is also **ONLY** possible to re-group auto-generated Analysis Units - Analysis Units created manually **cannot be grouped** in Execution Units. In this case, one Execution Unit will be created per manually defined Analysis Unit.

CAST has placed various GUI options at your disposal to achieve this:

Analysis Unit level

In each [Analysis Unit editor](#), there are the following options that enable you to create your own Execution Units from auto-generated Analysis Units.

Execution Unit annotation	<p>This option enables you to set an annotation on the Analysis Unit - all Analysis Units with the same Annotation will then be placed in the same Execution Unit and analyzed together.</p> <p>Setting an Annotation will force the Execution Unit max size (MB) to be deactivated. See Execution Unit max size (MB) below for more information.</p> <p>In addition the auto-management strategy will be deactivated for Analysis Units that have an annotation set.</p> <p>Note that this option is available for .NET technologies and for J2EE technologies, however, CAST recommends that you leave the management of Execution Units to the analyzer.</p>
Code Size (bytes)	<p>For auto-generated Analysis Units, this value is determined by the CAST Delivery Manager Tool. It displays the size of the source code in the Analysis Unit in bytes. Use this option to determine how many and which Analysis Units can be grouped in one Execution Unit (i.e. up to the Execution Unit max size).</p> <p>CAST recommends leaving this value as determined by the CAST Delivery Manager Tool - use the Reset button to return the value to value determined by the CAST Delivery Manager Tool.</p> <p>For manually created Analysis Units, this field will be blank and any value entered will be ignored.</p> <p>Note that this option is available for .NET technologies, however, CAST recommends that you leave the management of Execution Units to the .NET analyzer.</p>
Is Application entry point	<p>This option effectively enables you to force the CAST Management Studio to create one Execution Unit for the current Analysis Unit. Entry points should correspond to source code items such as EXE or entire websites (ASP) which need to be analyzed on their own and therefore should not be grouped with other Analysis Units.</p>

Technology level

In supported [Technology editors](#), the following option is available:

Execution Unit max size (MB)	<p>This option is only available for the following Technologies:</p> <ul style="list-style-type: none"> • VB <p>It enables you to set an upper code size threshold for a particular Technology. This value determines how many and which Analysis Units can be grouped in one Execution Unit.</p> <p>For example, you may have an Execution Unit max size of 100MB. If you have three Analysis Units, two of 50MB and one of 20MB, then all three cannot be grouped in one single Execution Unit.</p> <p>The default size is a recommended value that aims to prevent the analyzer crashing during the analysis and is used by CAST to automatically determine the Execution Units. You can increase or decrease this value, but doing so will have an impact as follows:</p> <ul style="list-style-type: none"> • Increasing the value will allow more Analysis Units to be automatically grouped in one Execution Unit, but does carry the risk that the analyzer may crash more frequently. • Decreasing the value will allow less Analysis Units to be automatically grouped in one Execution Unit, but can prevent the analyzer from crashing during the analysis (if this has been occurring). In addition: <ul style="list-style-type: none"> • decreasing the value when you have already run a snapshot can lead to Analysis Units being removed from a given Execution Unit and placed in a different one. If this occurs, then objects in the Analysis Unit(s) that have been moved will appear as removed and newly added in the CAST Engineering Dashboard. • decreasing the value can lead to a loss of links between objects during an analysis. <p>This option is only active when "automatic" grouping of Analysis Units in an Execution Unit is in use (i.e. the default behaviour) for supported technologies. As soon as a group of Analysis Units has an Annotation applied to it, the Execution Unit max size (MB) threshold is no longer active and all Analysis Units with the same Annotation will be grouped together in an Execution Unit regardless of their code size. There is therefore a risk that an analysis may crash if all Analysis Units with the same annotation (and thus in the same Execution Unit) have a large combined total code size. In other words, there is flexibility in that the Annotation deactivates the Execution Unit max size (MB) option allowing any number of Analysis Units to be grouped together in an Execution Unit, but you should take care not to group too many Analysis Units together if the combined code size is large as this does carry the risk that the analyzer may crash more frequently.</p>
-------------------------------------	--

Things to note

Unmerge

Analysis Units that are grouped manually in one Execution Unit should always, from that point onwards, be analyzed together in the same Execution Unit to retain the integrity of analysis results. If, however, the total source code size of the Analysis Units in the Execution Unit exceeds the **Execution Unit max size** threshold in Version N+1, AND the analyzer starts to crash (because the threshold has been exceeded) then there is no alternative but to "unmerge" an Analysis Unit from the Execution Unit so as the **Execution Unit max size** threshold is not exceeded.

You should be aware that this could change the results of your analyses and should be avoided if possible.

Impacts of source code size increase between versions

Take the following example:

- In V1 only one Execution Unit (A) was created for all Analysis Units in the Application
- In V2, the source code size has increased and exceeds the **Execution Unit max size** and CAST now decides to create additional Execution Units (A and B) to accommodate the new source code size

If an Analysis Unit was part of Execution Unit A in V1 and is now part of Execution Unit B in V2, then objects in the Analysis Unit will appear as removed and newly added in the CAST Engineering Dashboard.

Impacts of source code size decrease between versions

Take the following example:

- In V1 two Execution Units (A and B) were created for the Analysis Units in the Application because the source code size of all Analysis Units exceeded the **Execution Unit max size**
- In V2, the source code size has decreased and is now within the **Execution Unit max size** - CAST now decides to create only one Execution Unit (A)

If an Analysis Unit was part of Execution Unit B in V1 and is now part of Execution Unit A in V2, then objects in the Analysis Unit will appear as removed and newly added in the CAST Engineering Dashboard.

