


How to configure a language for the CAST Engineering Dashboard

- [Introduction](#)
- [Category for the technology and the module](#)
- [Categories for the objects](#)
- [Subset](#)
- [Assessment Model](#)
- [CAST Engineering Dashboard Cost Control](#)
- [CAST Enlighten Cost Estimation](#)
- [Quality status thresholds](#)

Introduction

The procedure below explains how to configure a Universal Language in the CAST Engineering Dashboard. This procedure will be illustrated with the PHP language.

Category for the technology and the module

 This section describes changes that must be made to the [xxxMetaModel.xml file](#).

First of all, you must select a category for an existing APM technology or create a new one. This category achieves two goals:

- identifies the technology name (in the *description* tag of the category)
- associates the modules with the technology

To create a new category, please insert it before the GrepTag called "GrepTag:APM_next_language". For example, for PHP, a new category "PHP" technology has been created:


```
<category name="APM PHP Module" rid="16">
<description>PHP</description>
<inheritedCategory name="APM Client Modules"/>
</category>
```

Next you need to associate the category to the type of project for the language:

```
<type name="phpProject" rid="6">
<description>PHP Project</description>
<inheritedCategory name="UAPProject"/>
<inheritedCategory name="PHP"/>
<inheritedCategory name="APM PHP Module"/>
</type>
```

Please note that the category description (here PHP) must be the same as the language name.

Categories for the objects

 This section describes changes that must be made to the [xxxMetaModel.xml file](#).

The relevant objects of the language must now be associated to the APM primitive categories. These categories are of two main types:

- **Inventory categories:** counts statistics on applications (number of classes, forms, etc.)
- **Process categories:** identifies objects for APM calculations (source objects, data objects, etc.)

To make the metamodel file more readable, it may be useful to create an applicative category. For example in PHP:

```

<category name="APM PHP Artifacts" rid="7">
<description>PHP Artifacts</description>
<inheritedCategory name="APM Client Language Artifacts"/>
</category>


```

Below is an overview of the APM categories described in the document above. Moreover, inventory categories determine the types of objects that must appear in "inventory" metrics.

Process categories	
APM Client Language Artifacts	Basic programming element (contains source code)
APM Sources	Highest level objects containing source code (used for line counting)
APM Forms	Forms
APM Events	Form event
APM Controls	Form control
APM Interfaces	Interface (object oriented languages)
APM Classes	Classes (object oriented languages)
APM Methods	Methods of a class / interface (object oriented languages)
APM Data Members	Members of a class (object oriented languages)

Inventory categories	
APM Inventory Files	For number of files
APM Inventory Methods	Methods
APM Inventory Functions	Functions
APM Inventory Functions and Procedures	SQL functions and Procedures
APM Inventory Triggers	Triggers
APM Inventory Events	Events
APM Inventory Macros	Macros
APM Inventory Programs	Programs

Subset

 This section describes changes that must be made to the [xxxMetaModel.xml](#) file.

You also need to define a subset type for your language – this is obligatory. The subset type allows you to create "User-defined modules" for your language type in the CAST Management Studio.

For example, for PHP, the subset has already been defined:

```

<type name="PHP_SUBSET" rid="9">
<description>PHP Subset</description>
<inheritedCategory name="PROJECT_SUBSET" />
<inheritedCategory name="PHP" />
<inheritedCategory name="APM PHP Module" />
</type>

```

The `<inheritedCategory name="PROJECT_SUBSET"/>` element is obligatory, the `<inheritedCategory name="PHP"/>` identifies the language category and the `<inheritedCategory name="APM PHP Module"/>` is identical to the category defined in the "project" type. Make sure you define a unique "rid" for the subset.

Assessment Model

After this, you must modify the parameter values for some Quality Rules, Distributions, Measures, and Sizing Indicators. CAST recommends modifying the following:

Metric Name	Description	Parameter	ID
+General Summary +Functional Weight +Backfired OMG-compliant Automated Function Points	BackFired OMG-compliant Automated Function Points	BackFired FP ratio	10201
+Technical Criteria +Volume - Number of LOC +Size Distribution	Distribution of artifacts regarding their size (Lines Of Code)	<ul style="list-style-type: none"> • Small Size Threshold • Average Size Threshold • Large Size Threshold • Very Large Size Threshold 	65105
+Technical Criteria +Architecture - Object-level Dependencies +Coupling Distribution	Distribution of artifacts regarding their coupling	<ul style="list-style-type: none"> • Low Coupling Artifacts • Average Coupling Artifacts Threshold • High Coupling Artifacts Threshold • Very High Coupling Artifacts Threshold 	65350
+Technical Criteria +Complexity - Technical Complexity +Cyclomatic Complexity Distribution	Distribution of artifacts regarding their cyclomatic complexity	<ul style="list-style-type: none"> • Low Complexity Artifacts • Average Complexity Artifacts Threshold • High Complexity Artifacts Threshold • Very High Complexity Artifacts Threshold 	65501
+Technical Criteria +Complexity - Algorithmic and Control Structure Complexity +Complexity Volume (% of LoC)	The rate of high and very high complexity is evaluated as a % of the number of code lines of the application details	Complexity Volume (% of LoC)	7122
+Technical Criteria +Architecture - Reuse +Reuse by Call Distribution	Distribution of artifacts regarding their reuse	<ul style="list-style-type: none"> • Low Reuse Threshold • Average Reuse Threshold • High Reuse Threshold • Very High Reuse Threshold 	66010
+Technical Criteria +Complexity - Algorithmic and Control Structure Complexity + Class Complexity Distribution (WMC)	Class Complexity Distribution (WMC)	<ul style="list-style-type: none"> • Low Complexity classes • Moderate Complexity classes • High Complexity classes • Very High Complexity classes 	66015
+Technical Criteria +Architecture - Object-level Dependencies + Class Fan-Out Distribution	Class Fan-Out Distribution	<ul style="list-style-type: none"> • Low Fan-Out classes • Moderate Fan-Out classes • High Fan-Out classes • Very High Fan-Out classes 	66020
+Technical Criteria + Architecture - Object-level Dependencies + Class Fan-In Distribution	Class Fan-In Distribution	<ul style="list-style-type: none"> • Low Fan-In classes • Moderate Fan-In classes • High Fan-In classes • Very High Fan-In classes 	66021

To do this, you must use the CAST Management Studio and the Assessment Model editor - import the Assessment Model from the CAST Dashboard Service. For example, for the Quality Distribution "Class Fan-Out Distribution" you need to drill down as follows:

Represents a model to assess the quality and the quantity of Applications.

Name: Assessment model imported from V704_CENTRAL

Quality Model | Sizing Model | Contextual Parameters | Background Facts | Services | Notes

Quality part of the Assessment Model.

Business Criteria | Technical Criteria | Quality Rules | Quality Distributions | Quality Measures | Consolidation Settings

List of Quality Distributions (tactical quality indicators, designed to assess a component based on the balance of the distribution of an attribute value among objects of the comp)

Name	Active	Orphaned	External ID
4GL Complexity Distribution	true	false	65601
Class Complexity Distribution (WMC)	true	false	66015
Class Fan-In Distribution	true	false	66021
Class Fan-Out Distribution	true	false	66020
Coupling Distribution	true	false	65350
Cyclomatic Complexity Distribution	true	false	65501
OO Complexity Distribution	true	false	65701
Reuse by Call Distribution	true	false	66010
Size Distribution	true	false	65105
SQL Complexity Distribution	true	false	65801

Name: Class Fan-Out Distribution

Active: Orphaned: Attach

Represents a tactical quality indicator, assessing a component based the balance of the distribution of a feature value among objects of the component.

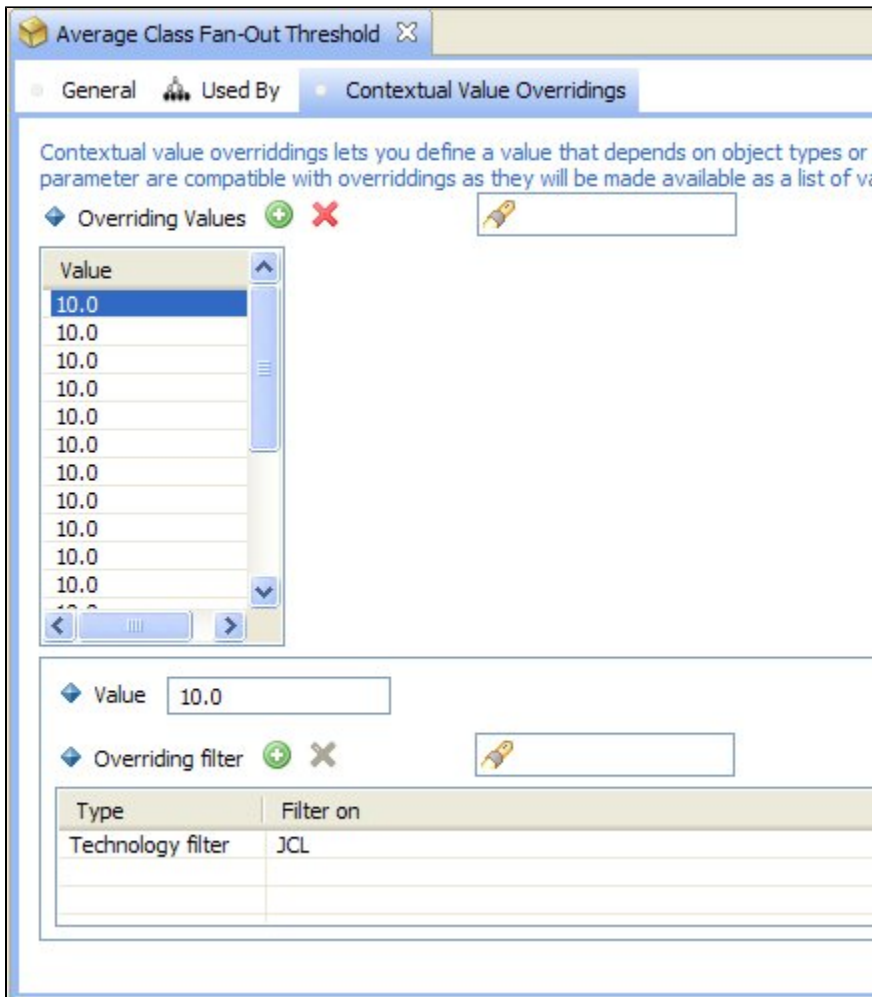
Documentation | Parameters | Grade Impacts | Settings | General | Notes

Configuration of input parameters that drives the distribution of objects in Categories.

Contextual Parameters

Type	Name	Parameter Index
Decimal number	Average Class Fan-Out Threshold	1
Decimal number	High Class Fan-Out Threshold	2
Decimal number	Very High Class Fan-Out Threshold	3

- Assessment Model
- Quality Model
- Quality Distributions
- Select the parent distribution ("Class Fan-Out Distribution" in this case)
- Parameters
- Select the appropriate Contextual Parameter
- Click the edit button
- A new tab will be opened as shown below:



- First check that the parameter value you require for your language is present in the list, if not, add it in.
- Then select the value in the list and add in a Technology filter for your new language - the language should be present if you have followed the instructions in the rest of this document.
- Changes will be taken into account during the next Snapshot generation.

CAST Engineering Dashboard Cost Control

To use the CAST Engineering Dashboard Cost Control functionality, you must enter a value for:

- Cost per man-day
- Change effort rates for artifacts of low, average, high and very high complexity.
- New development effort rates for artifacts of low, average, high and very high complexity.

This can be done in two ways:

- With the CAST Engineering Dashboard GUI:
 - From Quick Access, in Administrator pages, follow link : **Access To Cost Model Administration Page**
 - Enter **Edit technologies cost per man-day**
 - Enter **Edit technologies change effort rates**
 - Enter **Edit technologies new development effort bookmark rates**
- With the CAST Engineering Dashboard configuration file (you may need to create this file yourself):
 - Edit InstallScripts \ COST \ COST_ConfigData.xml file.
 - Enter new nodes of this type COST_CONFIG:

```
<COST_CONFIG>
<TECHNO_ID>-16</TECHNO_ID>
<PROFILE_NAME>Default</PROFILE_NAME>
<COST_NUM_VALUE>0.05</COST_NUM_VALUE>
<COST_VALUE_INDEX>0</COST_VALUE_INDEX>
</COST_CONFIG>
```

TECHNO_ID: ID of the technology created in the metamodel file. See section 9.2.

PROFILE_NAME: only 1 configuration profile for now. Must be set to 'Default'.

COST_VALUE_INDEX: Indicate the domain and the type of artifact:

- 0 : Cost per man-day
- 1 : Change effort rate for Low complexity artifact
- -1 : New development effort rate for Low complexity artifact
- 2 : Change effort rate for Average complexity artifact
- -2 : New development effort rate for Average complexity artifact
- 3 : Change effort rate for High complexity artifact
- -3 : New development effort rate for High complexity artifact
- 4 : Change effort rate for Very High complexity artifact
- -4 : New development effort rate for Very High complexity artifact

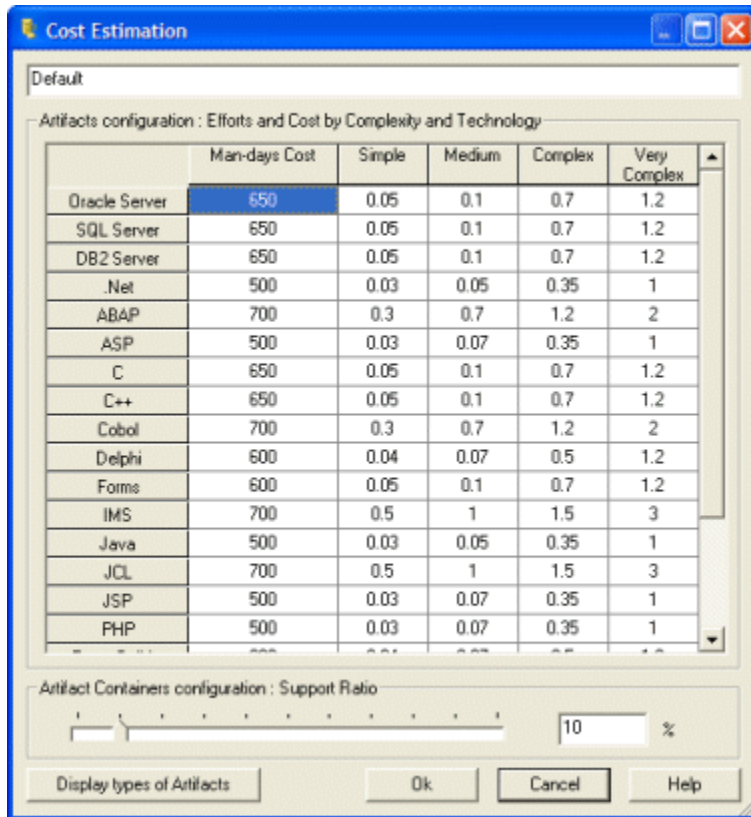
COST_NUM_VALUE: Value for the cost

In all, you must add 9 nodes (1 for Cost per man-day, 4 for new development effort rates and 4 for Change effort rates). Finally, you need to reinstall the **CAST Dashboard Service** using CAST Server Manager in order that this configuration file is taken into account.

CAST Enlighten Cost Estimation

In same way as described above, you must configure the cost of object modifications per technology. The Cost Estimation feature is based on APM Cost handling, thus it is mandatory to support these technologies in the CAST Engineering Dashboard.

To setup Cost, please launch CAST Enlighten and select **Tools > Cost Estimation > Cost Setup**:



Quality status thresholds

When adding a new diagnostic-based Metric, default status thresholds will be applied.

The default values are:

- From **0 up to 10%** of compliant objects: the grade is '1' and the status is 'Very High Risk' with a **red color-code**
- From **10 up to 70%** of compliant objects: the grade grows linearly from '1' to '2' and the status is still 'Very High Risk' with **red color-code**
- From **70 up to 90%** of compliant objects: the grade grows linearly from '2' to '3' and the status is 'High Risk' with **orange color-code**
- From **90 up to 99%** of compliant objects: the grade grows linearly from '3' to '4' and the status is 'Moderate' with **yellow color-code**
- From **99 up to 100%** of compliant objects: the grade is '4' and the status is 'Low Risk' with **green color-code**.

The thresholds that drive these statuses can be updated. This can be done in two ways:

- With the CAST Management Studio via the Assessment Model. Drill down as follows:
 - Quality Model
 - Technical Criteria
 - Select the appropriate Technical Criteria
 - Grade Impacts
 - Select the parent Business Criterion whose weight you need to modify
 - Modify the weight
 - Changes will be taken into account during the next Snapshot generation

Represents a model to assess the quality and the quantity of Applications.

Name: Assessment model imported from V704_CENTRAL

Quality Model | Sizing Model | Contextual Parameters | Background Facts | Services | Notes

Quality part of the Assessment Model:

Business Criteria | **Technical Criteria** | Quality Rules | Quality Distributions | Quality Measures | Consolidation Settings

List of Technical Criteria (operational quality indicators, designed to assess a technical area).

Name	Active	Orphaned	External ID
Architecture - Architecture Models Automated Checks	true	false	66070
Architecture - Multi-Layers and Data Access	true	false	61001
Architecture - Object-level Dependencies	true	false	61013
Architecture - OS and Platform Independence	true	false	61004
Architecture - Reuse	true	false	66009
Complexity - Algorithmic and Control Structure Com...	true	false	61009
Complexity - Dynamic Instantiation	true	false	61029
Complexity - Empty Code	true	false	61031
Complexity - Functional Evolvability	true	false	66008
Complexity - OO Inheritance and Polymorphism	true	false	61010
Complexity - SQL Queries	true	false	61011
Complexity - Technical Complexity	true	false	61026
Dead code (static)	true	false	61027

Name: Architecture - Architecture Models Automated Checks

Active: Orphaned: Attach

Represents an operational quality indicator, designed to assess a technical area.

Grade Contributors | **Grade Impacts** | Documentation | General | Notes

Business Criteria aggregating the current Technical Criterion.

Impacted Business Criteria

Weight	Critical	Business Criterion
1	false	Architectural Design
22	false	Technical Quality Index
6	false	Robustness
8	false	Changeability
8	false	Security

Weight: 1 Critical: Business Criterion: Assessment model imported from V704_CENTRAL / Architectural Design

- With the CAST Engineering Dashboard configuration file:
 - Edit InstallScripts \ ADG \ ADG_ConfigData.xml file (you may need to create this file yourself).
 - Enter new nodes of this type COST_CONFIG:

```
<DSS_METRIC_STATUS_THRESHOLDS>
<STATUS>0</STATUS>
<METRIC_ID>2000000</METRIC_ID>
<THRESHOLD_1>50.00000000000000</THRESHOLD_1>
<THRESHOLD_2>90.00000000000000</THRESHOLD_2>
<THRESHOLD_3>95.00000000000000</THRESHOLD_3>
<THRESHOLD_4>99.00000000000000</THRESHOLD_4>
</DSS_METRIC_STATUS_THRESHOLDS>
```

Where the following is true:

- **STATUS:** '0' for Diagnostic-based Metrics
- **METRIC_ID:** ID of the Diagnostic-based Metric created in the Metric Tree.
- **THRESHOLD_1:** percentage thresholds to switch from the '1' plateau to the '1' to '2' ramp; instead of 10%.
- **THRESHOLD_2:** percentage thresholds to switch from the '1' to '2' ramp to the '2' to '3' ramp; instead of 70%.
- **THRESHOLD_3:** percentage thresholds to switch from the '2' to '3' ramp to the '3' to '4' ramp; instead of 90%.
- **THRESHOLD_4:**percentage thresholds to switch from the '3' to '4' ramp to the '4' plateau; instead of 99%.



Note: Thresholds must be strictly ascending yet they can be outside of the 0-100 % interval if one status must never be reached.