

Outline page

- [Quality and Quantity section](#)
- [Top 10 Violations section](#)
- [Productivity section](#)
- [Content section](#)
- [Interpreting aggregate totals in the Outline page](#)
 - [Symptoms](#)
 - [Rationale](#)

Page frame name:

FRAME_PORTAL_TOP_LEVEL_DETAILS



Please note that this page is now deprecated and should no longer be used. It will be removed in a future version. In addition, screenshots have not been updated.

For each node i.e., Application, Module - and of the optional Organization Tree - i.e., Organizations, Teams, Developers -, the CAST Engineering Dashboard offers an **Outline page**. It is identical for all nodes.

Quality and Quantity section

This section contains four subsections:

- **Health Factor grades** - the grade of the node according to the five Application Health Factors (Transferability, Changeability, Robustness, Performance, and Security), the SEI Maintainability and the Technical Quality
- **Rule Compliance grades** - the grade of the node according to the three Rule Compliance (Programming Practices, Documentation, and Architectural Design)

Focus on: Application FRONT_OFFICE - part of LONDON - for Computed on 200907102123 [N/A] snapshot

Quality and Quantity

Assessment of the quality and quantity of the selected component.

Health Factors for Application FRONT_OFFICE in Computed on 200907102123 [N/A] snapshot (07-10-2009)

Assessment of the quality of the selected component. Click on the hyperlinks below to see the application risk factors for the current context - component and snapshot.

Transferability	3.15
Changeability	3.21
Robustness	2.94
Performance	3.24
Security	2.36

SEI Maintainability	3.4	3.4
Technical Quality Index	3.03	3.03

Technical Quality and SEI Maintainability assess the cost and difficulty/ease to maintain an application in the future. Technical Quality is based on 100s of metrics about the source code. SEI Maintainability is based on statistics about 100s of development projects.

Rule Compliance for Application FRONT_OFFICE in Computed on 200907102123 [N/A] snapshot (07-10-2009)

Assessment of the rule compliance for the selected component. Click on the hyperlinks below to drilldown through rule compliance information for the current context - component and snapshot.

Architectural Design	3.01
Documentation	2.68
Programming Practices	3.18

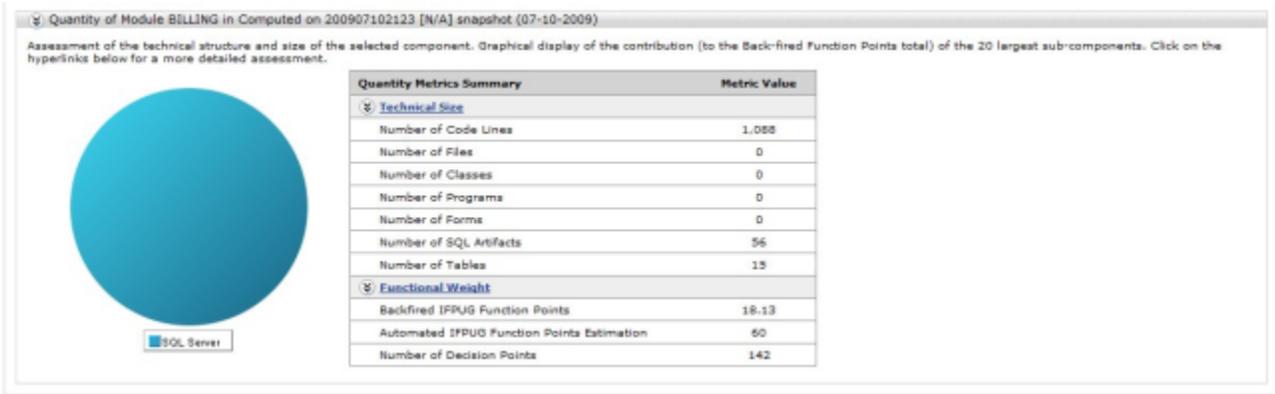
- **Violated Rules** - i.e. the Quality Rule-based metrics checked against the selected node

Violated Rules

This section lists all violated rules in the current context, along with the number of non-compliant objects.

	Diagnostic-based Metric Name	Number of Defective Objects	Metric Description
SQL Server	Diagnostic-based Metric relative to SQL Server language	139	139 SQL Server Objects are considered as defects
	T-SQL: Stored Procedure naming convention - prefix control	34	Names of Procedures should start with X. The prefix value is a parameter that can be changed at will
	T-SQL: Avoid using GOTO statement	16	Avoid using GOTO statement
	T-SQL: Avoid Functions and Procedures doing an Insert, Update, Delete, Create Table or Select without including error management	15	Avoid stored procedures or functions doing Insert, Update, Delete, Create Table or Select and not including error management (check @@error variable).
	T-SQL: Table naming convention - prefix control	15	Names of Tables should start with X. The prefix value is a parameter that can be changed at will
	T-SQL: Avoid tables without a clustered index	14	List all tables that do not have a clustered index
	T-SQL: Avoid Stored Procedures not returning a status value	10	Avoid Stored Procedures not returning a status value (RETURN)
	T-SQL: Avoid Functions and Procedures doing an Insert, Update or Delete without managing a transaction	7	Avoid Stored Procedures doing an Insert, Update or Delete and not managing a transaction (execute "begin tran" if @@trancount=0)
	T-SQL: Avoid Artifacts with lines longer than 80 characters	7	Avoid Artifacts with lines longer than 80 characters
	T-SQL: Avoid undocumented functions and procedures	5	Functions and procedures should have comments
	T-SQL: Avoid functions and procedures with a very low comment/code ratio	5	Functions and Procedures should have at least a ratio comment / code >= X. The threshold is a parameter and can be changed at will.

- **Quantity** - Main Technical Size and Functional Weight metrics for the selected node



Each subsection allows further drill-down into the Quality and Quantity model.

Top 10 Violations section

The "Top 10 Violations" section displays the 10 objects that violate the largest number of rules:

Top 10 Violations

List of objects that violate the largest number of programming rules. Click on the hyperlinks below for more details about the object.

Object Name	Object Full Name	Violation Count	New in Top 10?	Changed count?
CallCreateSale	WESLEY\WESLEY.CASTPUBS..CallCreateSale	7	No	No
CreateSale	WESLEY\WESLEY.CASTPUBS..CreateSale	7	No	No
CreateNewInvoice	WESLEY\WESLEY.CASTPUBS..CreateNewInvoice	6	No	No
CreateNewOrderLine	WESLEY\WESLEY.CASTPUBS..CreateNewOrderLine	6	No	No
DeliverOrder	WESLEY\WESLEY.CASTPUBS..DeliverOrder	6	No	No
DeliverOrderLine	WESLEY\WESLEY.CASTPUBS..DeliverOrderLine	6	No	No
history_proc	WESLEY\WESLEY.CASTPUBS..history_proc	6	No	No
CreateNewOrders	WESLEY\WESLEY.CASTPUBS..CreateNewOrders	5	No	No
GetAuthors	WESLEY\WESLEY.CASTPUBS..GetAuthors	5	No	No
storename_proc	WESLEY\WESLEY.CASTPUBS..storename_proc	5	No	No

Important note: the number and the list of violations of a shared object depend on the context and on the configuration. Some rules may not be checked for all contexts. Therefore one object may violate different number of rules depending on the context. The results displayed here concern the component whose name is indicated in the title bar.

i Important note: the number and the list of violations of a shared object depend on the context and on the configuration. Some rules may not be checked for all contexts. Therefore one object may violate a different number of rules depending on the context. The results displayed here concern the component whose name is indicated in the title bar.

Productivity section

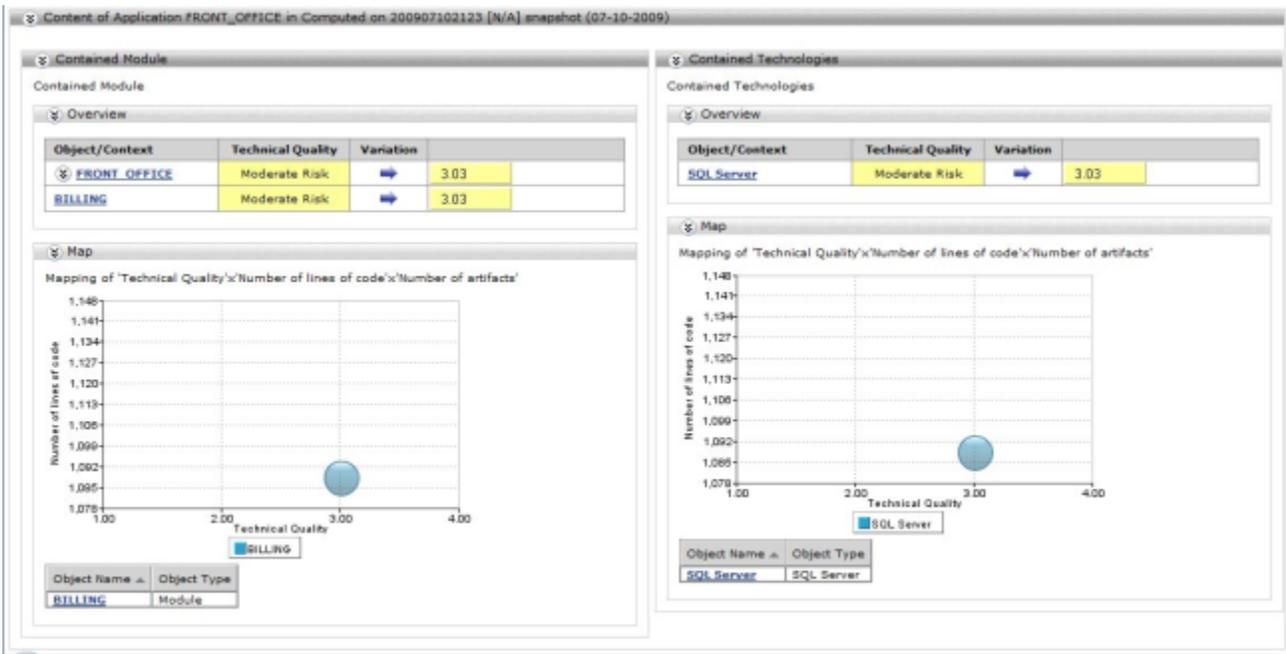
This section lists - and links to - the Organization Tree nodes (e.g., the list of Teams where Developers are assigned to the Modules of the selected Application) along with the consolidated Productivity metrics.

i Note that this section is optional - i.e. it will only be displayed if your CAST Administrator has uploaded Productivity information (known as Background Facts) using the CAST Management Studio - see [Background Facts and Business Value Metric upload](#) for more information.

Content section

This section displays - and links to - the nodes contained in the selected node, that is:

- Maps of Modules and Technologies in the case of an Application portrait
- A map of Modules in the case of a Technology portrait
- List of subordinate nodes, supervisor node, and impacted System/Application/Module nodes in the case of an Organization/Team/Developer portrait
- A hyperlink to the comparison of quality and quantity information from all available snapshots for the selected node



- A link to an **object browser** in the case of a **Module** portrait:

Object Browser for CASTPUBS - part of CASTPUBS-T-SQL Analyzer - in JAN snapshot

Module Contents - Objects

- CASTPUBS
 - SQL Trigger
 - SQL View
 - AuthorIdCodes
 - StoreIdView
 - titleview
 - SQL Index
 - audind
 - audmind
 - discounts_1481955782
 - PK_Invoice
 - PK_Invoice_Line
 - PK_Order_Line
 - PK_Orders
 - PK_publishers
 - PK_Sales
 - PK_Stock
 - PK_Stock_Line
 - PK_stores
 - roytoidind
 - salesdetailind
 - saltdind
 - tauidind

Object Overview

Object Name: CASTPUBS
Number of checked rules: 57

Object Details

Object Name: CASTPUBS
Object description: null
[Technical Details](#)

List of rules checked against the selected Module Subset with display of the Module metric grade and number of violation in the Module Subset.

Metric Name	Metric Grade	Violation Count
Avoid "SELECT *" queries	3.53	2
Avoid Artifact with high Commented-out Code Lines/Code Lines ratio	4	0
Avoid artifacts having recursive calls	4	0
Avoid having multiple artifacts deleting data on the same SQL table	4	0
Avoid having multiple artifacts inserting data on the same SQL table	4	0
Avoid having multiple artifacts updating data on the same SQL table	4	0
Avoid Module with high Commented-out Code Lines/Code Lines ratio	4	0
Avoid redundant indexes	3.37	1
Avoid SQL queries not using the first column of a composite index in the WHERE clause	1.83	4
Avoid SQL queries that no index can support	2.97	4
Avoid SQL queries using functions on indexed columns in the WHERE clause	3.23	3
Avoid SQL queries with implicit conversions in the WHERE clause	4	0
Avoid Too Many Copy Pasted Artifacts	1.97	4
Avoid too many indexes on one table	4	0
Avoid using SQL queries inside a loop	2.97	4
Complexity Volume [% of LoC]	4	0
Functional Evolvability	4	0
Never use SQL queries with a cartesian product	4	0
SQL: Avoid Artifacts with High Depth of Nested Subqueries	4	0
T-SQL: Avoid Artifacts with High Cyclomatic Complexity	4	0
T-SQL: Avoid Artifacts with High Depth of Code	4	0
T-SQL: Avoid Artifacts with High Essential Complexity	2.54	3

Drilling down into the Object Browser and selecting an object will display a **"Technical Details"** link in the right hand "Object Details" section. If you click this link, a second browser window will be loaded, which will actually transfer you into the **Discovery Portal**.

Interpreting aggregate totals in the Outline page

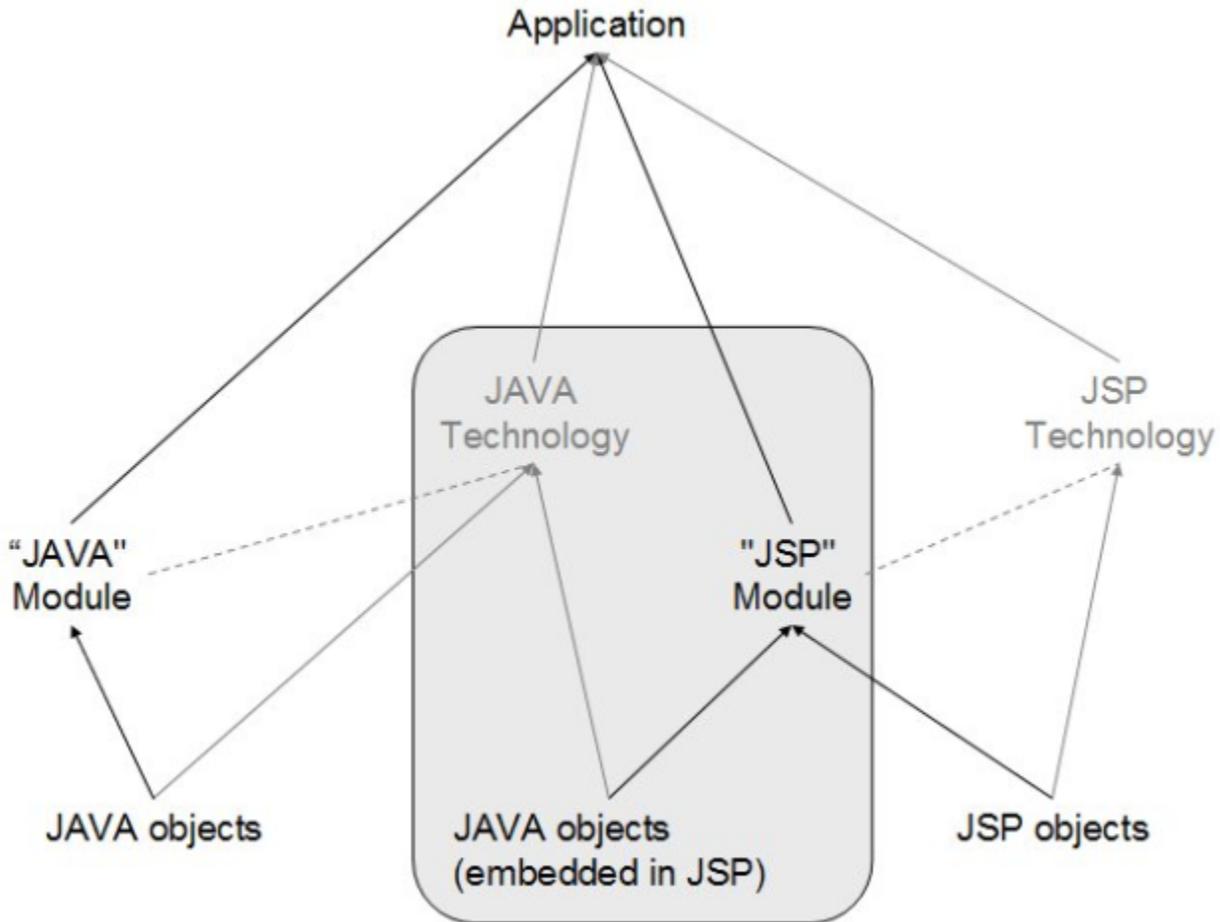
Symptoms

The Java technology aggregate for the application shows a higher number of lines of codes - or any other quantity metric value - than the sum of the lines of code for referenced modules in the "**Content of ...**" section in the Java Technology Outline page.

Rationale

As shown in the image below, the "**JSP**" **Module** also contains JAVA objects embedded in the JSP pages. These objects will form part of the number of lines of code count for JAVA Technology and for the "**JSP**" **Module**.

However, the "**Content of ...**" section in the Java Technology Outline page does not reference the "**JSP**" **Module**. Therefore, to validate the aggregated number of lines of code for the JAVA technology it is not enough to simply look at the referenced "JAVA" Module - you also need to look at the "**JSP**" **Module** to see the remaining JAVA lines of code.



This is also true for other technologies where one object type is embedded in another.