# CMS - Dependencies - Rules tab

**Dependencies - Rules tab**



This tab enables you to manage all the dependency rules for the Technologies in the Application.

When a dependency exists and during the analysis process, source code corresponding to the Technologies/Analysis Units in the **Source** column is scanned for references to objects in source code corresponding to the Technologies/Analysis Units in the **Target** column. If any matches are found, then a link between the two objects will be created (this will be visible in CAST Enlighten or in the CAST Discovery Portal).

**References** are traced using **search strings** which is less selective than parser based technology used for other links traced by the analyzer. This technology detects a reference to an object wherever its name is mentioned, regardless of the context in which this reference occurs. As a result, **incorrect links** may be traced if a string happens to match a given name even if it is not logically related to the corresponding object. As a result you may have to **int ervene** to filter incorrect references. Refer to the Dynamic Link Manager for more details on how to ignore these ambiguous links.

## Table columns

| Source | Lists the Technology or Analysis Unit that forms the **Source** for link creation (calling objects). |
|---|---|
| Target | Lists the Technology or Analysis Unit that forms the **Target** for link creation (called objects). |
| Refere nce Pattern | Indicates whether a **Reference Pattern** is also being used to determine links between Source and Target code. These can be added manually. Since **Reference Patterns** are based on **Regular Expression searches**, this is a way of adding **more accuracy** to the search for links between the Source and Target by targeting specific Regular Expressions. **Notes** <ul><li>If you add a Reference Pattern, only that pattern will be taken into account during the analysis - therefore if, for example, the Reference Pattern has been added to a default rule between two technologies, then you will need to create an additional custom rule without a Reference Pattern between the same technologies to make sure you still run the standard "search string" based searches between the technologies (if you require them of course).</li><li>You can add Reference Patterns regardless of the origin or the dependency rule (see below). Please see Dependencies - add a reference pattern for more information.</li><li>If you add a Reference Pattern and want to ensure that the Reference Pattern is applied to ALL Analysis Units of a given technology in the current Application, you need to ensure that you select the Technology type (for which the Reference Pattern has been created) as the Source for the Dependency.</li></ul> |

| Origin | Default | Default dependency Rules are created **automatically** by the CAST Management Studio between Technologies, as follows: |
|---|---|---|

<table>
<tr><td rowspan="3">Origin</td><td>Default</td><td>

Default dependency Rules are created **automatically** by the CAST Management Studio between Technologies, as follows:

If no dependency Rules are "discovered" (during the package configuration in the CAST Discovery Manager Tool) between Analysis Units in the Application, then CAST will use a "default" rule defined at Technology level (you can modify the default Dependencies yourself - please see the individual Technology for more information).

Note that generally speaking **database technologies** do not have any technologies set as **Target**. The exception to this is MS SQL Server which has a default Target .NET technology.

In the example screen shot below, no dependencies were "discovered" between any .NET Analysis Units and MS SQL Server Analysis Units - as such a default Technology level rule was used to ensure links were created where references exist:



A more complex example of this feature is as follows. Take a situation where we have the following rules defined at Technology level:

- J2EE > J2EE
- J2EE > SQL
- J2EE > Mainframe

If the following Analysis Units exist in the Application:

- AU1 (J2EE)
- AU2 (J2EE)
- AU3 (J2EE)
- AU4 (SQL)

And CAST has discovered the following dependencies automatically:

- AU1 > AU2
- AU3 > AU2

Then the following dependencies will exist in this tab:

- AU1 > AU2
- AU3 > AU2
- J2EE > SQL

In other words, the J2EE > J2EE rule has been removed because CAST has discovered that certain Analysis Units depend on others and has created specific Discovered rules for them. No dependencies were discovered between any J2EE Analysis Units and SQL Analysis Units, therefore the default rule J2EE > SQL will be applied.

The J2EE > Mainframe rule will not appear because there are no Analysis Units of this Technology type in the Application.

</td></tr>
<tr><td>Discovered</td><td>

Discovered dependency Rules are also created **automatically** by the CAST Management Studio - these Rules are, however, directly "discovered" from the source code, i.e., CAST has detected that one Analysis Unit depends on another.

Dependency Rules that have been "discovered" are **between Analysis Units**:



</td></tr>
<tr><td>Custom</td><td>

Custom dependency Rules are those that have been created manually. Typically these are created to cover a situation which the automatic Default and Discovered dependency Rules do not take into account.
For example, this may be **cross technology situations** where code in Analysis Unit Technology A relies on code in Analysis Unit Technology B:



Custom rules can be created between Analysis Units - and between Analysis Units and Refined Targets - see below for more information about creating Custom rules.

</td></tr>
</table>

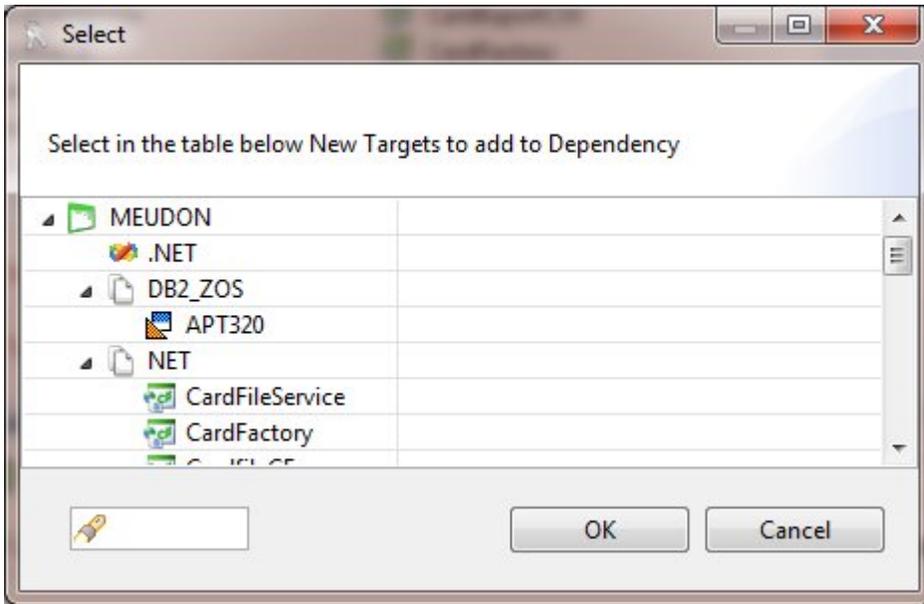Working with dependency rules via the available buttons

Use this button to add a **new Custom dependency Rule**. A blank line will appear in the list of dependencies, which when selected will enable you to define the Custom rule:



**Creating a rule**

- First choose the **Source** - click the "Select" link and the following dialog box will be displayed:



- Choose the Analysis Unit or Technology (within the Application) that will form the **Source**
- Repeat with the **Target** - when choosing a Target, you are not limited to the Analysis Units/Technologies within the current Application, you can also choose an existing Refined Target (i.e. if the selection of an Analysis Unit or a Technology is too broad), as shown below:
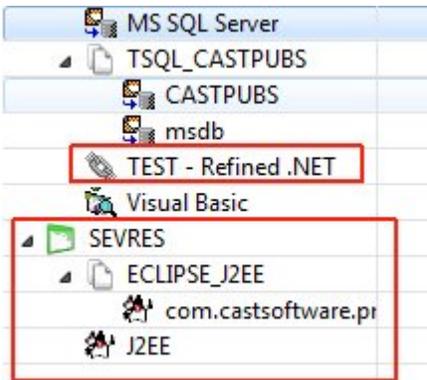


- Once the **Source** and **Target** are selected, the simple rule is complete and will be displayed in the list of rules as a "Custom" rule.
- During the next analysis or snapshot CAST will scan the Source for references to items in the Target and create links where appropriate.

📝 **Notes**

- Please see Dependencies - refine a target for more information about refining a target.
- Please also see Dependencies - using Reference Patterns for more information about adding Reference Pattern.

Use this option to **remove** a rule - you can remove any rule (Default, Discovered or Custom).

Use this option to **reset** the configuration: remove any Custom rules, and re-instate all removed Default and Discovered rules.

| | |
|---|---|
| | Use this option to **reset** the configuration: re-instate all removed Default and Discovered rules. |
| | Use this option to **clone** an existing rule |
| | This option will **transform** an existing Default or Discovered dependency rule into a Custom rule. |
| | This option is only available if you have added a **Reference Pattern** to an existing dependency. Clicking the option will **run the** Reference Pattern between the Source and Target and **save** the results to the CAST Analysis Service. |

See Also

Dependencies - refine a target | Dependencies - using Reference Patterns

C A S T