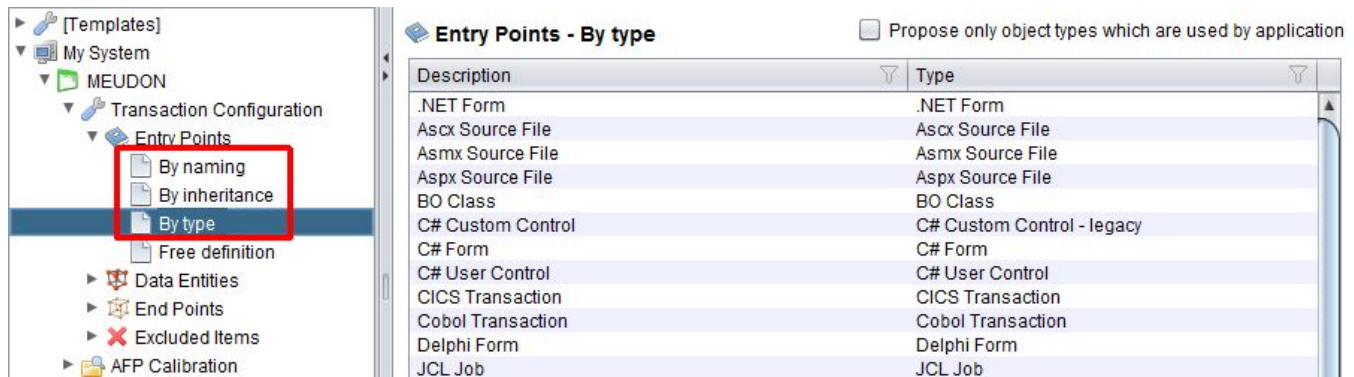


# TCC - By naming, By inheritance, By type nodes - Right hand panel

## By naming, By inheritance, By type nodes - Right hand panel

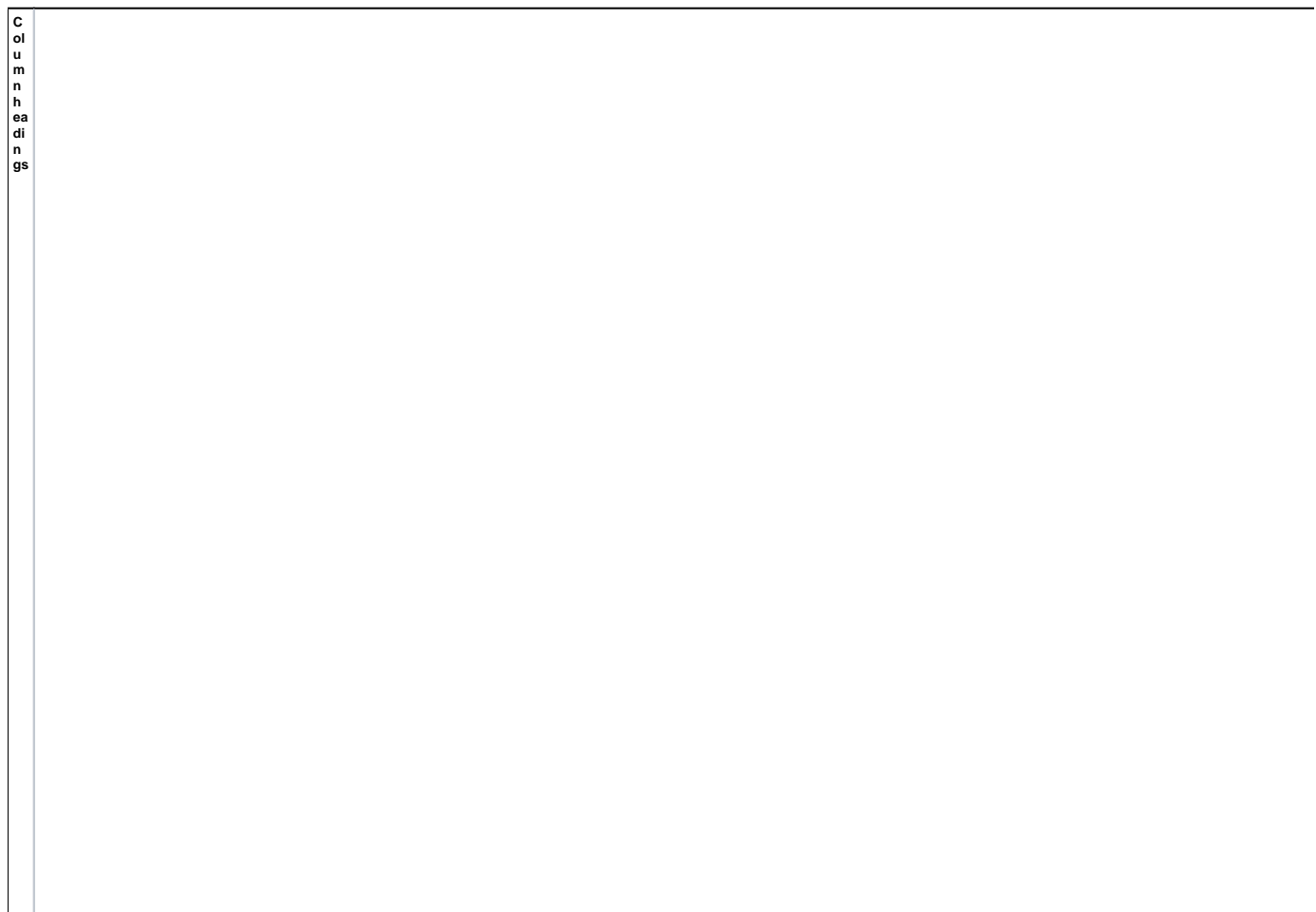
The display in the main window for these nodes lists all the sets that have been defined whether via [standard configuration files \(.TCCSetup\)](#) or specifically for the parent Application. You can **create new**, **edit existing** or **delete existing** sets from within this window (see [Set creation](#) for more information). The example below shows the **Entry Points - By type** node:




Each line in the right hand window represents one set that has been defined. In the example above there are two sets defined:

- The set entitled **SQL View** detects objects in the parent Application that are SQL views (detection **By Type**). The default DET and RET values have been given for these object types
- The set entitled **SQL Index** detects objects in the parent Application that are SQL indexes (detection **By Type**). The default DET and RET values have been given for these object types

Other options are available:



<b>Description</b>	Displays a description for the set - double click to activate a cell and make changes to your set's description.																											
<b>Type</b>	Displays a drop down list that lists either all objects that can be saved in the Analysis Service or only those used by the parent Application - depending on the position of the option <b>Show only object types which are used by application</b> - see below. This list is used when creating a set - see <a href="#">Set creation</a> for more information.																											
<b>Name</b>	Enables you to enter an object naming convention used when creating a set (see <a href="#">Set creation</a> for more information). Note that when entering search strings here, a simple <b>LIKE</b> comparison is executed to find the matching objects. As such, you can use the following characters as <b>wildcards</b> : <ul style="list-style-type: none"> <li><b>% (percent)</b> - this wildcard will match any string of characters, even empty characters</li> <li><b>_ (underscore)</b> - this wildcard will match any single character</li> </ul> <p>The wildcard characters % and _ cannot be escaped, therefore you cannot search explicitly for objects that contain these characters in their name. As such, you may find that when using the above wildcards, objects are matched which you do not require. If this is the case, you can use the <b>MATCH</b> option with a <a href="#">Regular Expression</a> to achieve what you need.</p> <p>Not available in the <b>By type</b> node.</p>																											
<b>Activation</b>	The <b>Activation</b> heading indicated whether the rule/set is "Active" or "Inactive". Special considerations: <ul style="list-style-type: none"> <li>When the Function Point computation is executed (either in the CAST Transaction Configuration Center using the <a href="#">Compute</a> option or during snapshot computation), only active configuration rules are taken into account (the inactive ones will be ignored).</li> <li>Exception: <b>Data Entities &gt; Built-in parameters</b> and <b>Free Definitions &gt; Generic Sets</b> will always be considered as active (they do not have the "Activation" attribute).</li> <li>The default value when a configuration rule is added automatically from a configuration file will be "Active"</li> <li>The default value when a configuration rule is manually created by the user will be "Active"</li> <li>The user will be able to change the Activation status of a configuration rule by selecting the status in the Activation column and toggling between Active and Inactive.</li> <li>"Generate sets" and "Generate all sets" actions will ignore the inactive configuration rules.</li> </ul> <p>See <a href="#">Working with standard configuration files (TCCSetup)</a>.</p>																											
<b>Updated</b>	When a new package version is available, it will be automatically loaded. When this process is finished, the CAST Transaction Configuration Center will flag those configuration rules that are new or updated, regarding the previous version, using the "Updated" colour. NOTE: The "Updated" column will only be relevant for standard configuration rules. The user-defined rules (those with "custom" package) will never be flagged as updated. For example, let us consider a package P with an initial version 1.0, which defines 3 rules: A, B, C. <pre>package="P"; version="1.0"Rule ARule BRule C</pre> The first time this package is loaded, all rules will be considered as "new", and therefore they will all be flagged as UPDATED: <table border="1"> <thead> <tr> <th>Rule</th> <th>Updated</th> <th>Package</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Yes</td> <td>P</td> </tr> <tr> <td>B</td> <td>Yes</td> <td>P</td> </tr> <tr> <td>C</td> <td>Yes</td> <td>P</td> </tr> </tbody> </table> This flag will remain until the next version package P is delivered. Let us consider that, after some time, a new version of package P (1.1) is delivered, containing 4 rules: A, B, C, D: <pre>package="P"; version="1.1"Rule ARule BRule CRule D</pre> Thus it contains one modified rule (C) and one new rule (D). The next time the CAST Transaction Configuration Center is opened, it will automatically refresh the rules of package P, installing the new version 1.1. When this is done, the rules C and D will be flagged as updated: <table border="1"> <thead> <tr> <th>Rule</th> <th>Updated</th> <th>Package</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>No</td> <td>P</td> </tr> <tr> <td>B</td> <td>No</td> <td>P</td> </tr> <tr> <td>C</td> <td>Yes</td> <td>P</td> </tr> <tr> <td>D</td> <td>Yes</td> <td>P</td> </tr> </tbody> </table> See <a href="#">Working with standard configuration files (TCCSetup)</a> .	Rule	Updated	Package	A	Yes	P	B	Yes	P	C	Yes	P	Rule	Updated	Package	A	No	P	B	No	P	C	Yes	P	D	Yes	P
Rule	Updated	Package																										
A	Yes	P																										
B	Yes	P																										
C	Yes	P																										
Rule	Updated	Package																										
A	No	P																										
B	No	P																										
C	Yes	P																										
D	Yes	P																										
<b>Package</b>	The Package heading identifies the origin of a configuration rule/set, with the special value "custom" for those configuration rules that are user-defined. Special considerations regarding package column: <ul style="list-style-type: none"> <li>Package attribute cannot be defined by the user, the configuration rules will take either the value defined in the standard configuration files or the special value "custom" for the user-defined rules</li> <li>The standard rules (i.e. those coming from the standard configuration files, and therefore with package value different from "custom") cannot be edited, removed or cut; they can only be activated/deactivated.</li> <li>The standard rules (i.e. those coming from the standard configuration files, and therefore with package value different from "custom") can be copied; but when pasted, the pasted item will get package "custom".</li> <li>The custom rules can be fully edited (except for the "package" value, which must always be "custom").</li> <li>All Configuration rules will have the "package" attribute, except for the elements inside <b>Data Entities &gt; Built-in parameters</b> (these items are special and are handled differently from the rest)</li> </ul> <p>See <a href="#">Working with standard configuration files (TCCSetup)</a>.</p>																											
<b>Technical</b>	Only available in the <b>free definition, by naming, by inheritance and by type</b> sub-nodes of the <b>Excluded Items</b> node. This option enables you to designate any objects in the excluded set as "technical objects" in the next snapshot. To designate the objects in the set as "technical" from the next snapshot, set the option to YES (by default it is set to NO):   <table border="1"> <thead> <tr> <th>Description</th> <th>Type</th> <th>Activat...</th> <th>Upd...</th> <th>Package</th> <th>Technical</th> <th>Number of objects</th> </tr> </thead> <tbody> <tr> <td>TEST</td> <td>Java Class</td> <td>ACTIVE</td> <td></td> <td>custom</td> <td>NO</td> <td>148</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li><b>NO:</b> When this column is set to NO (default), objects which are Entry Points, Data Entities or End points are excluded i.e. the complete Data Function or Transactional Function that has these objects as entry points, data entities or end points will not consider calculations. But if these objects are not entry points, data entities or end points but are part of Transactional Function, then they will have an impact on the status of the Transactional Function when the Automated Enhancement Point (AEP) measure is being used (the default). If the objects are entry points, data entities or end points the complete Data Function or Transactional Function which have these objects as entry points, data entities or end points will not be considered at all for Function Point calculation.</li> <li><b>YES:</b> When this column is set to YES, objects which may be (but not necessarily) entry points, data entities and end points are considered as "technical objects" and they do not impact the status of Transactional Functions when the Automated Enhancement Point (AEP) measure is being used (the default). If the objects are entry points, data entities or end points the complete Data Function or Transactional Function which have these objects as entry points, data entities or end points will not be considered at all for Function Point calculation.</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>it is not necessary to use the <a href="#">Recompute checksums and snapshot statuses</a> option when setting the Technical column to YES.</li> <li>values for previous snapshots will not be impacted by this option, which is in contrast to excluding objects in the <a href="#">Enhancement node</a>, which is global to all snapshots.</li> <li>see also <a href="#">Exclude objects</a>.</li> </ul>	Description	Type	Activat...	Upd...	Package	Technical	Number of objects	TEST	Java Class	ACTIVE		custom	NO	148													
Description	Type	Activat...	Upd...	Package	Technical	Number of objects																						
TEST	Java Class	ACTIVE		custom	NO	148																						
<b>DET, RET, Contribute</b>	Please see <a href="#">DET, RET, Contribute values</a> for more information about these columns.																											
<b>Number of objects</b>	Displays how many objects have been identified by the rule. This column will only update after using the <a href="#">Compute</a> option or generating a snapshot.																											

You can also use the Filter icon  in each column heading to sort and filter according to content:

- Choose the filter type in the drop down list
- Choose whether the criteria will be applied using case sensitive text or not.
- Enter the filter text.

Only content that matches the chosen criteria will be displayed.

Shortcut menu

The shortcut menu can be displayed by right clicking anywhere in the main window:

<b>New line</b>	Use this option to create a new set (see <a href="#">Set creation</a> ). A blank set will be displayed.
<b>Cut</b>	Use this option to cut an existing set to the clipboard.
<b>Copy</b>	Use this option to copy an existing set to the clipboard.
<b>Paste</b>	Use this option to paste a set from the clipboard - for example when copying sets from one node to another.
<b>Delete</b>	Use this option to delete an existing set.
<b>Generate set</b>	This option will calculate the selected set - see <a href="#">Generate sets</a> for more information.
<b>Generate all sets</b>	This option will calculate all the sets in the main window - see <a href="#">Generate sets</a> for more information.

Propose only object types which are used by application

When selected, this option will hide all objects that are not used by the current Application from the drop down list in the **type** column (where available). In the example below only Java related objects can be expanded:

When not selected, all possible objects that can be saved in the Analysis Service are listed in the drop down list in the **type** column.

This option is particularly useful as it narrows down the choice of object types when you are creating a set. The list of objects that can be stored in the Analysis Service is very long, so restricting the list to only the objects current application can save time.

## By Type node - Entry points node

The **By Type** node that is visible under the **Entry point node** (in the [Templates node](#)) will always contain a default list of sets that are defined by CAST:

**Entry Points - By type**  Propose only object types which are used by application

Description	Type
.NET Form	.NET Form
Ascx Source File	Ascx Source File
Asmx Source File	Asmx Source File
Aspx Source File	Aspx Source File
BO Class	BO Class
C# Custom Control	C# Custom Control - legacy
C# Form	C# Form
C# User Control	C# User Control
CICS Transaction	CICS Transaction
Cobol Transaction	Cobol Transaction
Delphi Form	Delphi Form
JCL Job	JCL Job

The sets define object types that are generally ALWAYS considered as **Entry points** - therefore CAST has created the sets for you. In contrast to the [Built-in parameters node](#), you can freely **create new**, **edit existing** and **delete existing sets**, whether in the [Templates node](#) or the [System node](#).

Please note, however, that if you delete ALL sets in the **By Type** node when working in the [Templates node](#), and then close and re-launch the CAST Transaction Configuration Center, all deleted sets will be re-displayed. Deleting only some of the sets, and then closing and re-launching the CAST Transaction Configuration Center will NOT cause all sets to be redisplayed.

Remember that modifying a set in the [Templates node](#) will not cause corresponding sets located in existing or new Systems/Applications to be modified.

See Also

[Set creation](#)

