

TCC - By naming, by inheritance, by type, free definition and Built-in parameters - Browser

By naming, by inheritance, by type, free definition and Built-in parameters - Browser

Each of the sub-nodes in the [Templates] and System nodes contains further sub-nodes. Each is explained below:

-  By naming
-  By inheritance
-  By type
-  Free definition
-  Built-in parameters

- [By naming](#)
- [By inheritance](#)
- [By type](#)
- [Free definition](#)
- [Built-in parameters](#)
- [Right hand panel](#)

By naming

Use this node to set up sets that detect objects according to their **type** and **name**:

- object types are specified using the drop down list
- object names are specified using text
 - if objects are named with a specific prefix, enter the **prefix** (for example L_ would capture objects called "L_test", "L_new" etc.).
 - if the objects you are targetting are named with a specific **suffix**, enter the suffix preceded by the % sign (for example %_I would capture objects called "test_I", "new_I" etc.).
 - to look for a substring in the name you can use **%Substring%** which will look for all names containing the word **Substring**
 - You can also enter specific object names, although this will only capture one object.

By inheritance

Use this node to set up sets that detect objects that inherit directly or indirectly from another object defined according to its **type** and **name**. Inheritance is where one object re-uses another object - typically classes (for example, in Java the code "extends" signifies object inheritance).

The same configuration is used as for the **By naming** node described above.

See [By naming](#), [By inheritance](#), [By type nodes - Right hand panel](#) for more information.

By type

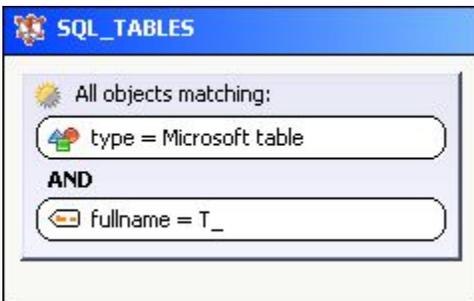
Use this node to set up sets that detect objects purely according to their type. Object types are specified using the drop down list.

Note that CAST uses a predefined list of Transaction entry points (defined in the [Templates node](#)). Object types that appear in this list are always considered **Entry Points** (e.g.: C# forms / VB.NET forms / VB forms etc.).

See [By naming](#), [By inheritance](#), [By type nodes - Right hand panel](#) for more information.

Free definition

Use this node to create sets that detect objects using filters that can be defined graphically. You can create complex filters to detect very specific object types - they are much more flexible than the naming, inheritance and type nodes and can even combine multiple criteria:



See [Free definition - Right hand panel](#) for more information.

Built-in parameters

The **Built-in parameters** node is only available in the **Data Entities** node. This section provides:

- **Built-in type for data entities** > a set of object types that are ALWAYS considered as Data Entities - this list is predefined by CAST:

Data Entities - Built-in parameters

Built-in types for data entities

Type	Description
CAST_Oracle_RelationalTable	Oracle table
CAST_MSTSQL_RelationalTable	Microsoft table
CAST_Oracle_MaterializedView	Oracle materialized view
CAST_ASETSQL_RelationalTable	Sybase table
CAST_TSQL_UDT_TABLE	SQL Server UDT TABLE
SQL_TABLE	SQL Table
CAST_ABAP_SAPUnresolvedTable	SAP unresolved Table
SAPTable	SAP Table
CAST_Oracle_ObjectTable	Oracle object table
CAST_DB2ZOS_RelationalTable	DB2 z/OS relational table
CAST_CICS_DatasetPrototype	CICS DataSet
CICS_DATASET	CICS DataSet - legacy
COBOL_DBD	Cobol DB Definition
CAST_COBOL_SavedFileDescription	Cobol File Link - legacy
CAST_COBOL_ExternalFilePrototype	Cobol File Link
CAST_IMS_SavedDatabase	IMS DB Definition - legacy
CAST_IMS_DatabasePrototype	IMS DB Definition
CAST_IMS_AnalyzedSegment	IMS DB Segment
CAST_IMS_SegmentPrototype	IMS Segment

- **FK/PK analysis: table prefixes to ignore** > this section allows you to define a list of prefixes that are used to name your SQL tables. Any prefix on the list will be ignored when the CAST Transaction Configuration Center algorithm attempts to detect data functions from SQL tables using Foreign key (FK) and Primary Key (PK) relationships.
- **Names of tables to ignore (case insensitive regular expressions)** > this section allows you to define criteria based on Regular Expressions that are used to match specific table names. All matching objects will not be considered Data Functions at all.

See [Built-in parameters node - Right hand panel](#) for more information.

Right hand panel

The right hand panel lists any sets that have been created and also enables you to create new, edit existing and delete existing sets:

Data Entities - By type

Description	Type
SQL_VIEW	... soft view

	New line	Ctrl+N
	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Delete	Delete
	Generate set	
	Generate all sets	

Please see the [Right hand panel](#) section for more information about using this window.