

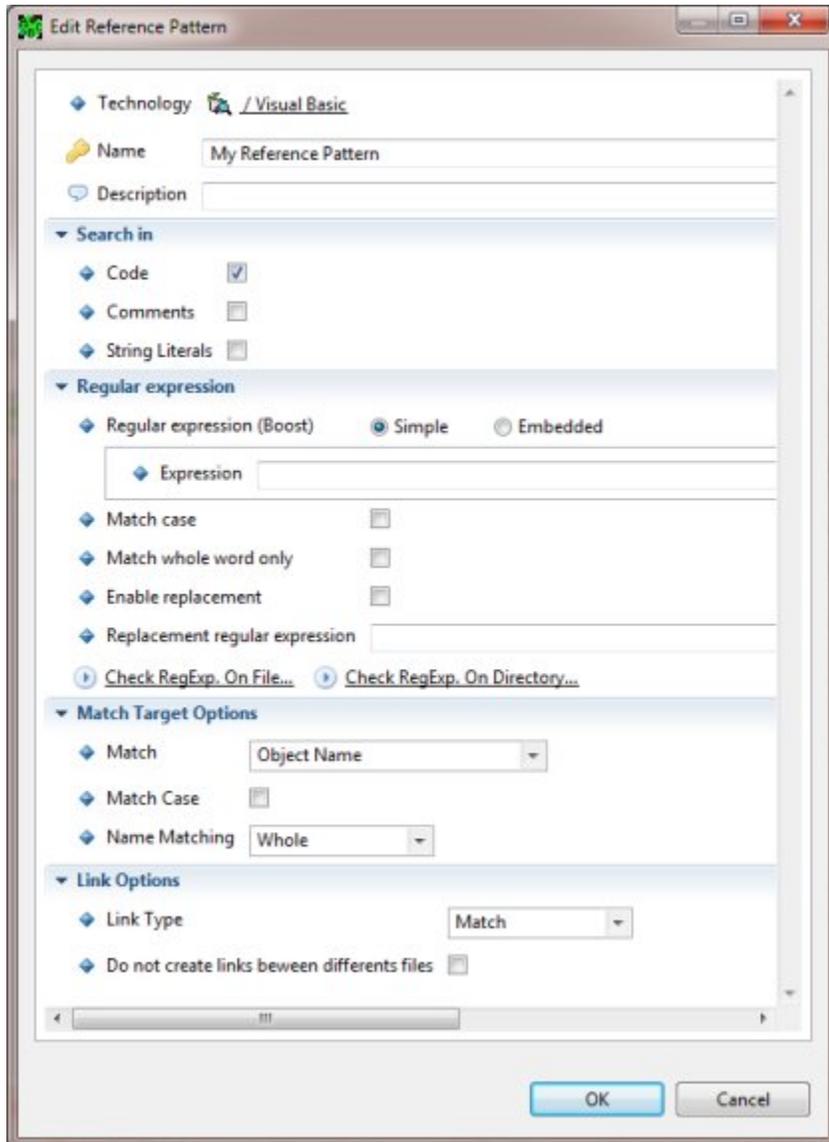
CMS - Dependencies - using Reference Patterns

Dependencies - using Reference Patterns

The CAST Management Studio includes a built in feature called **Reference Patterns** - their usage is explained in more detail in [Reference Pattern tab](#) (part of the [Technology editors](#)). This page explains how to create a new Reference Pattern and how to test and use it.

Add a new Reference Pattern

- You must be working in [Expert mode](#)
- Open the required [Technology editor](#) - choose the Technology type that will form the **Source** Technology (see [Dependencies - Rules tab](#) for more information)
- Select the [Reference Pattern tab](#)
- Click the  button to **add** a new Reference Pattern - the **Reference Pattern dialog box** will then be displayed - in the example below, we have chosen Visual Basic as the Technology type:



- Now fill in the relevant field and options as required. Each is explained below:

Technology	Displays the Technology type that will be targeted by the Reference Pattern. This is not editable.
Name	Choose a name for the Reference Pattern. CAST recommends choosing a name that will help you recognise the Reference Pattern more easily.

Description	Add an optional description of the Reference Pattern to help distinguish it.						
Search in	<table border="1"> <tr> <td data-bbox="162 205 284 233">Code</td> <td data-bbox="284 205 1495 233">This section enables you to choose where you want to search for the string entered in the Regular Expression text box (see below). By default, the Code option is selected.</td> </tr> <tr> <td data-bbox="162 233 284 260">Comments</td> <td data-bbox="284 233 1495 260">Select the options you require by clicking in the check box next to the item.</td> </tr> <tr> <td data-bbox="162 260 284 289">String Literals</td> <td data-bbox="284 260 1495 289"></td> </tr> </table>	Code	This section enables you to choose where you want to search for the string entered in the Regular Expression text box (see below). By default, the Code option is selected.	Comments	Select the options you require by clicking in the check box next to the item.	String Literals	
Code	This section enables you to choose where you want to search for the string entered in the Regular Expression text box (see below). By default, the Code option is selected.						
Comments	Select the options you require by clicking in the check box next to the item.						
String Literals							
Select the language to which it applies	<p>This section is only visible for Technologies that have multiple component languages, for example:</p> <ul style="list-style-type: none"> • J2EE (HTML, JSP, Java, JavaScript VBScript) • ASP (HTML, ASP, JavaScript, VBScript) • .NET (ASP, C#, HTML, JavaScript, VBScript, VB.NET) • Mainframe (CICS, Cobol, IMS, JCL) <p>The purpose is to enable you to configure specific search parameters for a specific component language if required. When the Reference Pattern is run (see below), the search parameters will be applied to each selected component language independently of the other component languages.</p>						
Languages filter	<p>Don't filter on languages > this option is active by default - the languages filter is disabled.</p> <p>Filter on languages (choose below) > choosing this option will enable the table below, allowing you to choose the component languages that you want the search to target:</p> <div data-bbox="324 514 698 766" style="border: 1px solid #ccc; padding: 5px;"> <p>Type</p> <p><input checked="" type="checkbox"/> HTML Language</p> <p><input checked="" type="checkbox"/> JSP Language</p> <p><input type="checkbox"/> Java Language</p> <p><input type="checkbox"/> Java Script Language</p> <p><input type="checkbox"/> VB Script Language</p> </div> <p>Note that if you want to target different component languages with different search parameters (for example one search for HTML and one for JSP), then you must create multiple Reference Patterns and assign them each to specific Dependency Rules.</p>						
Regular Expression (boost)	<p>This section enables you to choose what to search for and whether to enable "replacement".</p> <ul style="list-style-type: none"> • Simple (default) > Enter the Regular Expression, word or phrase in the Expression text box that you want the Reference Pattern to target. Please use the Regular Expressions hints and tips to formulate a Regular Expression • Embedded > Select this option if you want to restrict the search to a specific zone. You can specify the start and end of the zone through Regular Expressions. Then specify the regular expression you want to search for within this zone. <p>For example, in a Cobol program, you want to search all code for all instances of PERFORM that can be found between IF and END-IF, and all instances of TEST that can be found between PERFORM and END-PERFORM. Configure the following Embedded Regular Expressions:</p> <ol style="list-style-type: none"> 1. Begin = IF Expression = PERFORM End = END-IF 2. Begin = PERFORM Expression = TEST End = END-PERFORM <p>Use the buttons to add, edit or remove Embedded expressions. Using the  button will display a popup box enabling you to enter the expressions:</p> <div data-bbox="324 1102 1079 1585" style="border: 1px solid #ccc; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 5px;"> <p>New Rule</p> <p>◆ Begin <input type="text" value="IF"/></p> <p>◆ Expression <input type="text" value="PERFORM"/></p> <p>◆ End <input type="text" value="END-IF"/></p> <p style="text-align: center;">OK Cancel</p> </div> </div>						
Notes	<p> Notes</p> <ul style="list-style-type: none"> • Please note that the limits of the zones you are searching (i.e.: the Begin and End Expressions) MUST NOT be located in comments. • Zones can overlap one another and a zone can be included within another zone. • Each zone will be searched independently for its own regular expression. 						
Match case	<p>If you select this option, the search will only return results that match the case sensitivity of the string you entered. In other words, if you enter Select, then SELECT or select will not be returned.</p> <p>By default this option is left blank, meaning case sensitivity will be ignored.</p>						

Match whole word only

If you select this option, the results that are returned will correspond **exactly** to the string you enter in the **Expression** text box. For the following text examples:

1. 'select * from TITLES
2. 'select * from _TITLES
3. 'select * from vTITLES
4. 'select * from TITLEAUTHOR
5. 'select * from TITLEAUTHOR, TITLES
6. 'select * from TITLES_AUTHOR
7. 'select * from TITLEA_AUTHOR
8. 'select * from TITLES_____AUTHOR
9. 'select * from TITLESAUTHOR
10. 'select * from TITLE
11. 'select * from TITLEdAUTHOR

...the search string "**TITLE[A-Z]+**" will identify the following hypothetical results:

- **TITLES** for examples 1 and 5
- **TITLEAUTHOR** for examples 4 and 5
- **TITLES_AUTHOR** for example 6
- **TITLEA_AUTHOR** for example 7
- **TITLES_____AUTHOR** for example 8
- **TITLESAUTHOR** for example 9
- **TITLEdAUTHOR** for example 11

If you leave this **option blank**, then the default setting (**sub-match**) will be applied. In effect this means that the results that are returned will correspond **in part** to the string you enter in the Expression text box.

The search string "**TITLE[A-Z]+**" will identify the following hypothetical results:

- **TITLES, _TITLES, TITLES** for examples 1, 2 and 5
- **TITLEAUTHOR** for examples 4 and 5
- **TITLES_AUTHOR** for example 6
- **TITLEA_AUTHOR** for example 7
- **TITLES_____AUTHOR** for example 8
- **TITLESAUTHOR** for example 9
- **TITLEdAUTHOR** for example 11

 **Notes**

- Please note, however, that if you select **Match whole word only** and NOT the **Match case** option, only whole words will be identified. For example, using

ATTRIBUT[a-z]+

the following results will NOT be identified:

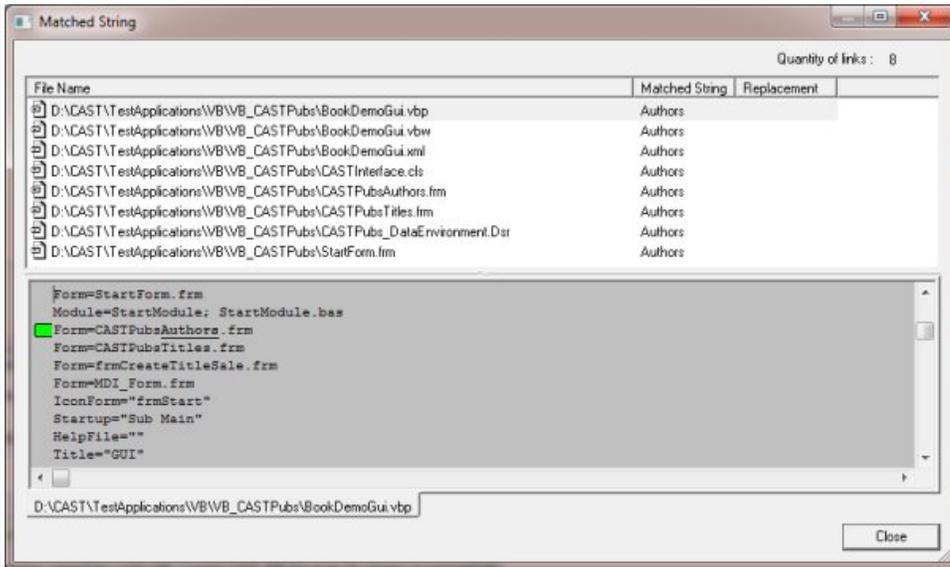
A AT ATT ATTRI ATTRIB ATTRIBU ATTRIBUT

but will go directly to ATTRIBUT instead.

A Regular Expression will thus match the longest string.

<p>Enable Replacement</p>	<p>Activating this option enables you to apply a replacement process to the results of the Regular Expression search prior the results being saved to the Analysis Service</p> <p>How does it work?</p> <p>Each time the Regular Expression is matched in the source, the chosen replacement string is produced and is used to match the name (with the same sub/over/whole match options).</p> <p>Replacement is based on Regular Expression grouping. For example:</p> <ul style="list-style-type: none"> Using the Regular Expression R: <code>(([a-zA-Z])([a-zA-Z_0-9]+))</code> each parentheses pair generates a grouping and that grouping can be referenced using the notation <code>\1</code>, <code>\2</code>, etc., <code>\n</code>. If R matches the text "The_Cat" then <code>\1</code> is the character "T" and <code>\2</code> is the string "he_Cat". Using the Regular Expression S: <code>(HisFunc)MyFunc([^\s]*)</code>. If S matches the text "HisFunc(his_parameter)", then <code>\1</code> is "HisFunc" and <code>\2</code> is "his_parameter". <p>Examples</p> <p>These examples illustrate how this feature could be used:</p> <p>1)</p> <ul style="list-style-type: none"> Regular Expression entered: <code>LoadLibrary("([^\r\n]"")\d\d\d")</code> Replacement entered: <code>\1</code> This combination will match the name of the DLLs (without the extension) called in C/C++ source code <p>2)</p> <ul style="list-style-type: none"> Regular Expression entered: <code>com\my_package\.([a-zA-Z_][a-zA-Z_0-9]*)</code> Replacement entered: <code>\2</code> This combination matches the last part of qualified names beginning with <code>com.my_package</code> <p>3)</p> <ul style="list-style-type: none"> Regular Expression entered: <code>ld(d x)_Object_.([a-zA-Z_][a-zA-Z_0-9]*)</code> Replacement entered: <code>ld\1_\2</code> This combination matches the the names that have the form "ldd_Object_Something" and then eliminates the "Object" in the middle E.g.: <ul style="list-style-type: none"> <code>ldd_Object_Frame</code> -> <code>ldd_Frame</code> <code>ldd_Object_Window</code> -> <code>ldd_Window</code> <code>ldx_Object_Button</code> -> <code>ldx_Button</code> <p>4)</p> <ul style="list-style-type: none"> Regular Expression entered: <code>System.loadLibrary([\s]*"([^\s]*)"[\s]*)</code> Replacement entered: <code>\1.dll</code> This combination matches the names of the C libraries used to call functions via Java native methods <p>Order of events during execution</p> <p>A source code analysis has already been carried out and the Analysis Service contains the objects resulting from this analysis. A Reference Pattern is then created and the Enable Replacement option is activated and a replacement text entered in the field. When the Reference Pattern is then run, the following occurs:</p> <ul style="list-style-type: none"> The string chosen as the replacement string is checked for validity, then one of the following occurs: <ul style="list-style-type: none"> If the chosen replacement string is not valid, the Reference Pattern cannot be executed. The replacement string is considered invalid if it is empty (i.e. nothing entered in the field) or if it contains a reference to a non-existent grouping (i.e. <code>\4</code> whereas the Regular Expression contains only two groups). If the chosen replacement string is valid, the Reference Pattern will then be run. During the process, each time a match with the Regular Expression is located in the selected objects, it is transformed using the chosen replacement string. The result of this transformation is then compared to the names/full names/paths of the Target objects. For each object "A" whose name matches the result of the transformation, a link will be created between the object containing the Regular Expression match and object "A". <p>Limitations</p> <ul style="list-style-type: none"> The maximum number of groupings that you can reference is 9, which means that if you specify <code>\10</code> as a replacement it will be interpreted as <code>\1</code> followed by the character "0" (zero). <p> Notes</p> <ul style="list-style-type: none"> It is worth remembering that if the grouping is the object of a repetition then only the last match in the grouping will be retained. Take for example: <ul style="list-style-type: none"> Regular Expression entered: <code>[a-zA-Z_][a-zA-Z_0-9-]*</code> Match string: <code>James_Brown</code> - Replacement entered: <code>\1</code> <p>In this case, <code>\1</code> corresponds to "N" and not to "ames_Brown"</p> <p>Thus there is a difference between the following two Regular Expressions using the same match string and replacement:</p> <ul style="list-style-type: none"> <code>[a-zA-Z_][a-zA-Z_0-9-]*</code> = "N" <code>[a-zA-Z_][a-zA-Z_0-9-]*</code> = "ames_Brown"
<p>Replacement Regular Expression</p>	<p>Enter the replacement Regular Expression text.</p>
<p>Check RegExp. on file</p>	<p>These options enable you to test the search string on a specific file or on a specific folder of files - a standard Windows dialog box will be displayed enabling you to choose the file or folder you want to run the search on.</p> <p>Results are displayed as follows:</p>

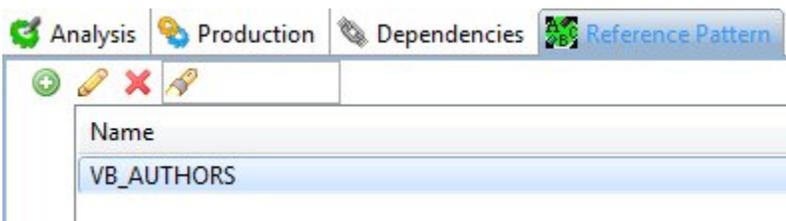
Check
RegExp. on
directory



The upper section lists the file or files that contain a matched string for the Regular Expression, together with the actual matched string. Clicking on a file will show the file's code and a bookmark (green icon) will be placed on the matched string.

Match Target options	Match	This section enables you to define what the results of the Source search will be matched to in the Target . Choose from: <ul style="list-style-type: none"> • Object Name > The object's short name • Object Full Name > The object's full name as stored in the Analysis Service • Object Path > The path of the object as stored in the Analysis Service
	Match case	Select this option if you want the results of the Source search to match the Target using case sensitivity. For example, a Source search for author[a-z] may return strings in the Source such as UTHORS or authors . Links would only be created to objects called authors in the Target if this Match case option is selected.
	Name Matching	<ul style="list-style-type: none"> • whole > exact result must correspond to the names of the objects in the Target Selecting this option will force the Reference Pattern engine to look for objects in the Target whose names correspond exactly to the match found in the Source. For example, if the Regular Expression is t_[a-zA-Z0-9_]+ then t_a is a match, so if the Target contains an object named t_a the a link will be created between the match found in the Source and this object, but there will be no link to an object named zt_a, nor to an object named t_. <p>You could also use this option to find all references to a database table from the analyzed code. For example, if the table name in the Target is MyTable: use MyTable as the Regular Expression to search for in the Source and select the option Whole. When the Reference Pattern is run you should see links between the following lines <i>int MyTable</i>; // link only if the code option in the Search in section has been checked <i>char * str = "SELECT COUNT(*) FROM MyTable"; // link only if the String Literals option in the Search in section has been checked but there will also be a link to the following line:</i></p> <pre>char * str2 = "SELECT COUNT(*) FROM MyTable2";</pre> <ul style="list-style-type: none"> • over > exact result must correspond to part of the names in the Target Selecting this option will force the Reference Pattern engine to look for objects in the Target whose names contain the match found in the Source. For example, if you used the Regular Expression t_[a-zA-Z0-9_]+ then t_a is a match, so if the Target contains an object named t_a, t_atable or t_aMyTable there will be a link between the match found in the Source and these objects. <p>You could also use this option to find links between the analyzed code and database tables whose names contain a certain word. You could also use this option to find links to a file whose name contains a certain word.</p> <p>Here's another example: Say you have a file named my_file.cob and you want to find all calls made to it without the extension ".cob". First you enter my_file as the Regular Expression, then you select the option Over, and you ensure that the file "my_file.cob" is in the Target. • sub > part of the result must correspond to the names of the objects in the Target Selecting this option will force the Reference Pattern engine to look for objects whose names are part of the match found in the Source. For example, if you used the Regular Expression t_[a-zA-Z0-9_]+ then t_a is a match, so if the Target contains an object named t_a, or t_a there will be a link between the match found in the Source and these objects, but there will be no link to a selected object named zt_a. </p>
Link Options	<p>Link Type Use this option to select the link that will be created between the Source and the Target. By default the link type is set to Match.</p> <p>Do not create links between different files Use this option if you want to avoid links being created between different source code files or between files and database objects. In other words, this restricts the links to inter-file links.</p>	

- Click **OK** to confirm the settings.
- The new Reference Pattern will now be displayed in the [Reference Pattern tab](#):



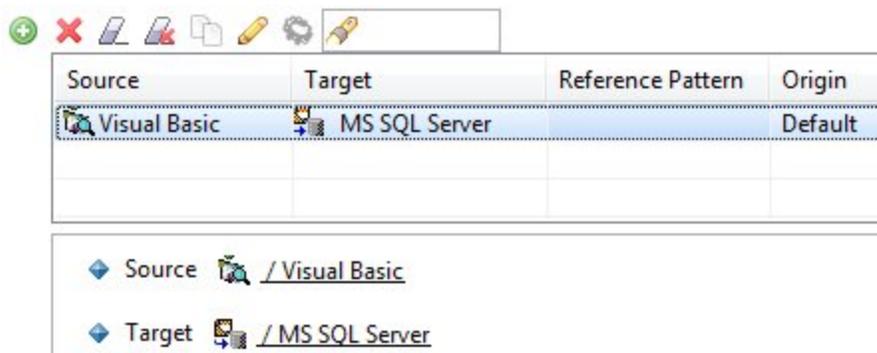
- See below for more information about using the Reference Pattern.

Using a Reference Pattern you have added

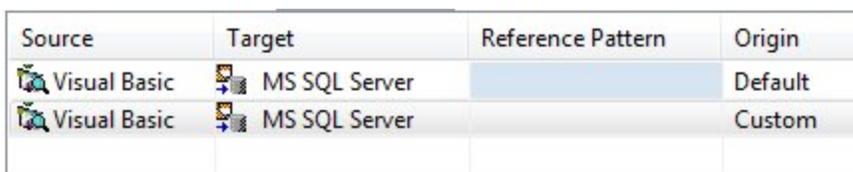
To use a Reference Pattern that you have added, you need to associate it to an existing Dependency Rule. Dependency Rules are managed in the [Dependencies tab](#) in the [Application editor](#):

- You must be working in [Expert mode](#)
- Open the [Application editor](#)

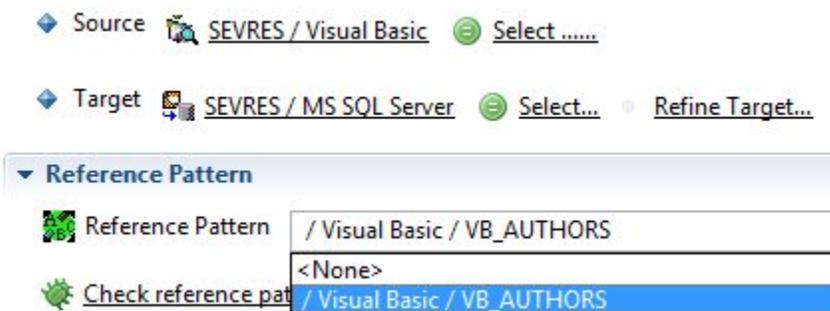
- Select the [Dependencies - Rules tab](#)
- Select the existing Default rule between the two Technologies - the Source technology must be the same as the Technology for which you created the Reference Pattern::



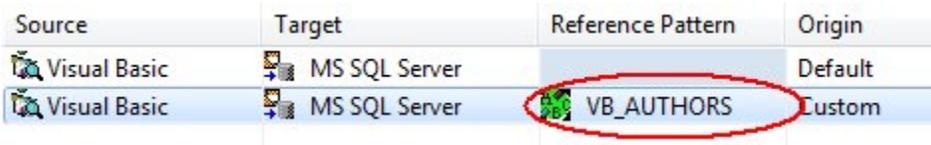
- You can then either:
 - **Copy** the rule to leave the Default rule in place (we will do this)
 - **Edit** the existing rule to transform it into a Custom rule
- Click the button to copy the existing rule
- Select the new custom rule:



- Expand the **Reference Pattern** section and select the Reference Pattern you created above - only Reference Patterns that match the Source Technology for the current Dependency Rule will be displayed:



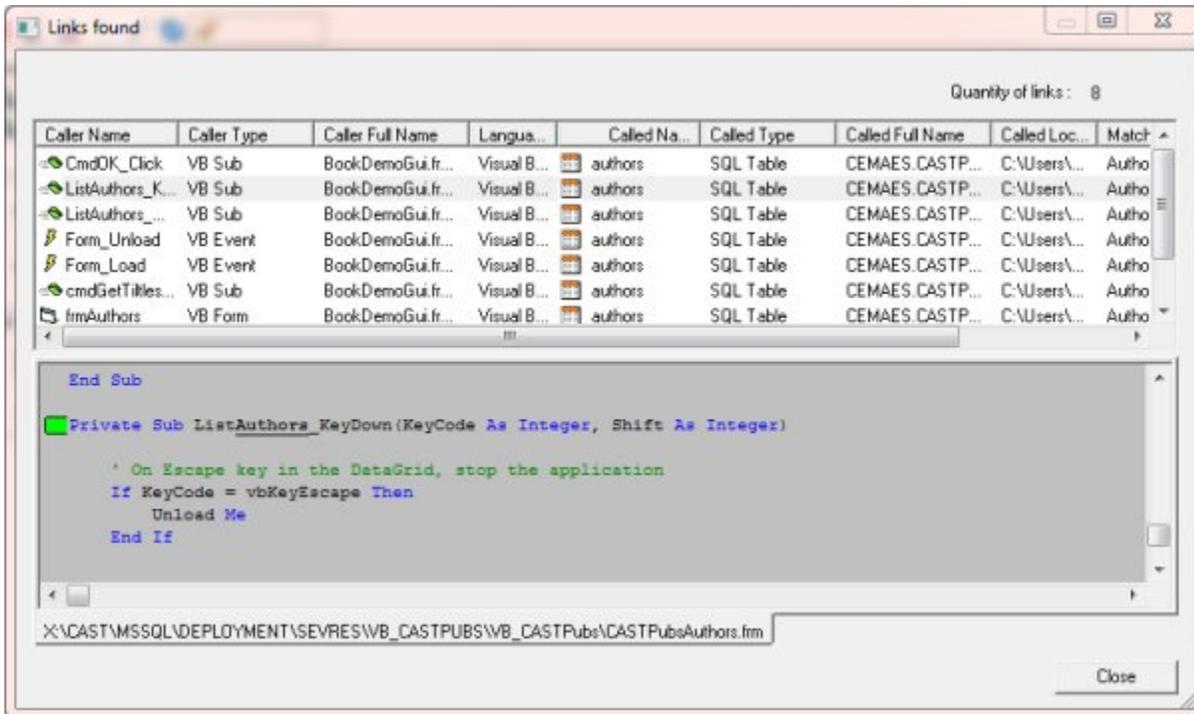
- The Reference Pattern will now be associated with the custom Dependency Rule:



Testing a Reference Pattern

Before running a Reference Pattern and saving the results in the Analysis Service, you have several opportunities to test the outcome of the process:

- You can test the Regular Expression search string you have defined on an individual file or folder of files (see the **Check RegExp. on file** and **Check RegExp. on directory** options listed above for more information)
- You can test the Reference Pattern itself on a Dependency Rule using the **Check Reference Pattern Dependency Results** option in the [Dependencies tab](#) (an analysis of the Application must have already been complete). The upper section lists the file or files that contain a matched string for the Regular Expression, together with the actual matched string and the object in the Target which will be linked. Clicking on a line in the upper section will show the file's code and a bookmark (green icon) will be placed on the matched string which will create the link.



Running a Reference Pattern

There are several ways to run a Reference Pattern:

- Use the **Run Reference Pattern on Dependency** option in the [Dependencies tab](#) - this runs the Reference Pattern against a specific Dependency Rule and results will be saved to the Analysis Service
- Create a **Reference Pattern tool** - see the [Content Enrichment tab](#) in the [Application editor](#).
- **Run an analysis** - during the analysis the Reference Pattern (if configured in a Dependency Rule) will be run
- **Generate a Snapshot** -during the analysis part of the snapshot generation process, the Reference Pattern (if configured in a Dependency Rule) will be run

Results

Results (i.e. links between source code objects) can be displayed in **CAST Enlighten**.

See Also

[Dependencies tab](#) | [Application editor](#) | [Reference Pattern tab](#)

