

# Task Factory Advanced Lookup Cache

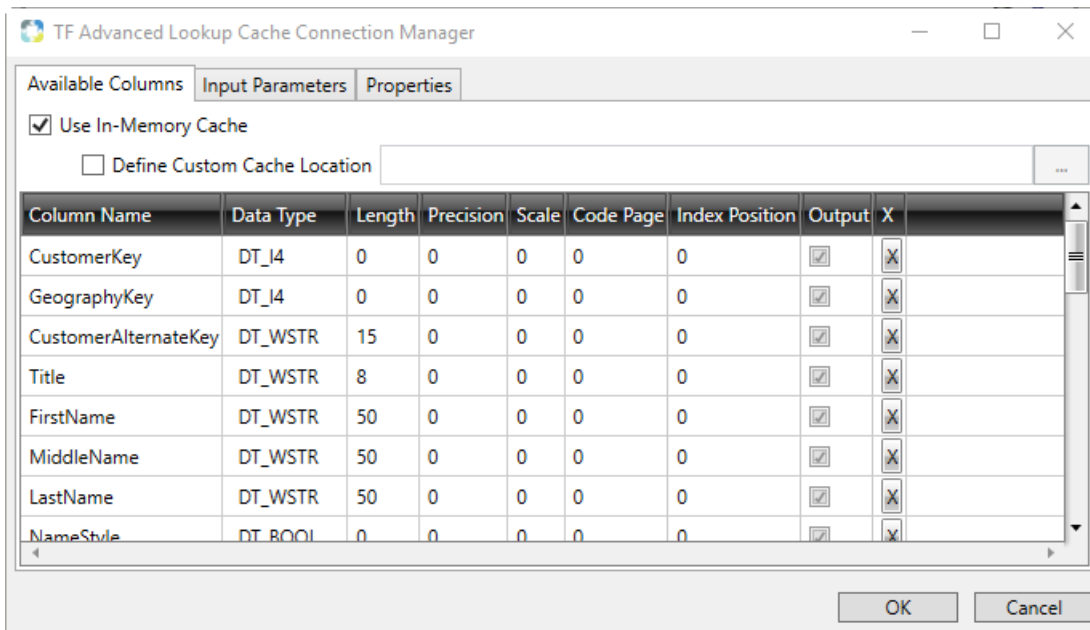
Last Modified on 28 September 2020

## Advanced Lookup Cache Connection Manager

The **Advanced Lookup Cache Connection Manager** is used to setup the cache database and configure how data is retrieved from it.

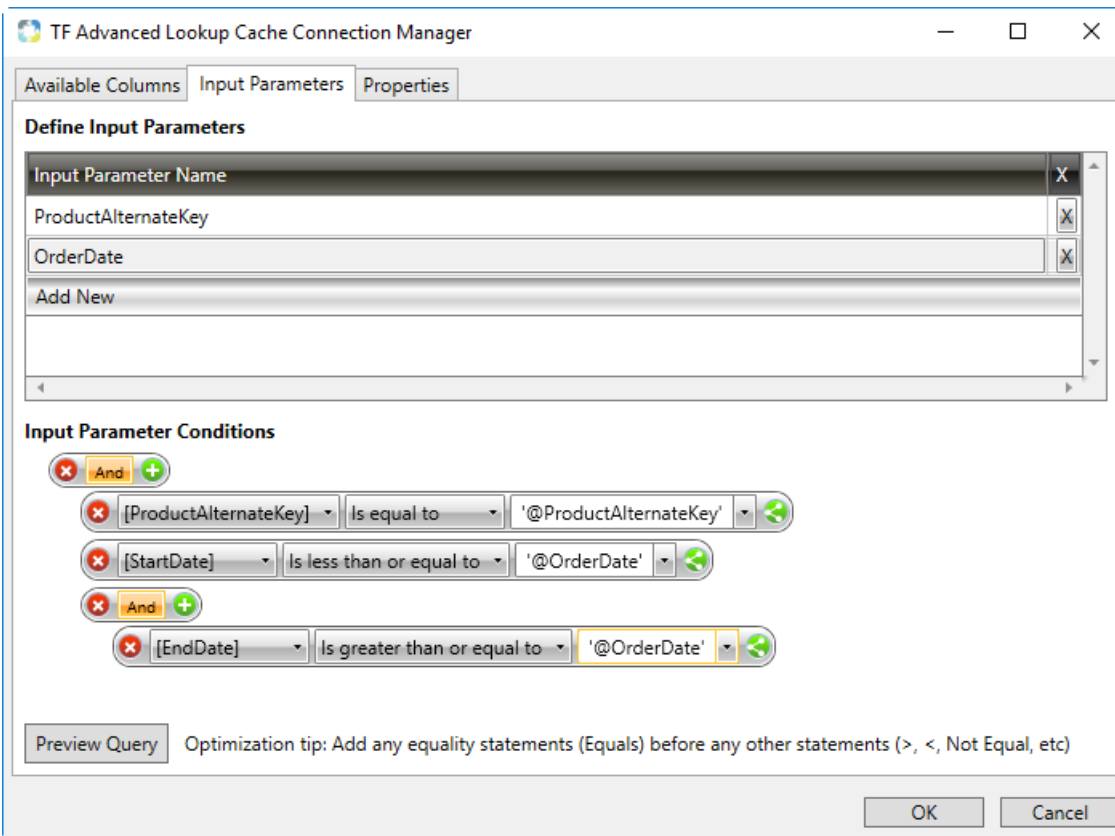
### Available Columns Tab

The available columns tab allows you to define columns that are part of the cache, whether or not the column is output when a lookup is performed, and what type of cache database to use.

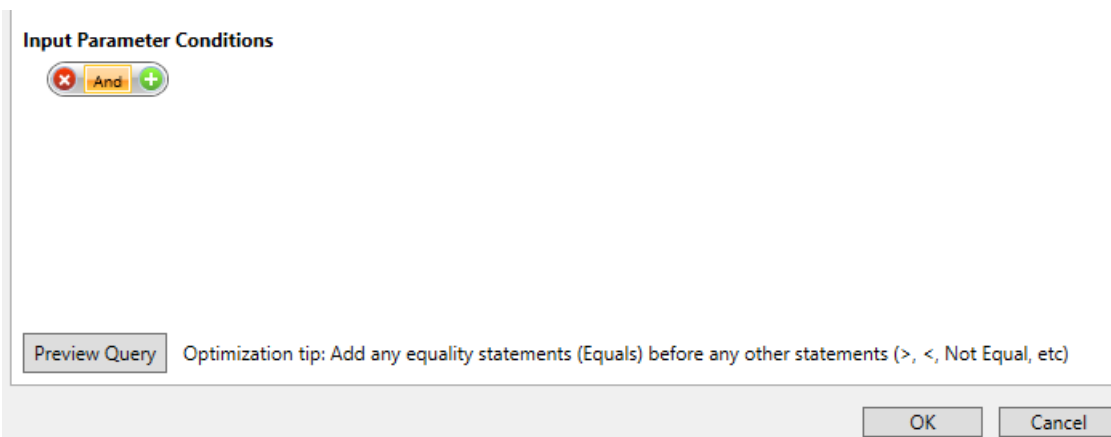


Option	Description
<b>Use In-Memory Cache</b>	This tells the <u>connection manager</u> whether to use in-memory cache or on-disk cache when <u>creating the cache database</u> .
<b>Define Customer Cache Location</b>	By default, the cache connection manager is created in a temporary file location defined by SSIS. You can choose to store the database in a custom location by checking this box and entering/selecting a path to the file. <b>Note:</b> The file should be defined with an .s3db extension.
<b>Output Columns</b>	The <b>Output column</b> tells the connection manager to return this value when a lookup is performed. The output columns can be seen on the <b>Advanced Lookup Transform UI</b> after a cache connection manager is selected.

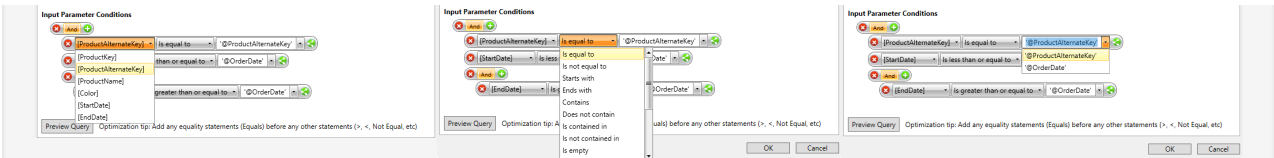
### Input Parameters Tab



Option	Description
<b>Input Parameter</b>	Used by the <b>Advanced Lookup Transform</b> to define the lookup values to find rows in the cache database. They are mapped from <b>Input columns</b> in the <b>Advanced Lookup Transform</b> .
<b>Input Parameter Condition</b>	This is where you can easily create conditions for queries. By selecting the word <b>And</b> , you can select between an <b>And</b> and <b>Or</b> conditions. Selecting the green <b>+</b> adds more conditions and selecting the red <b>x</b> removes them.

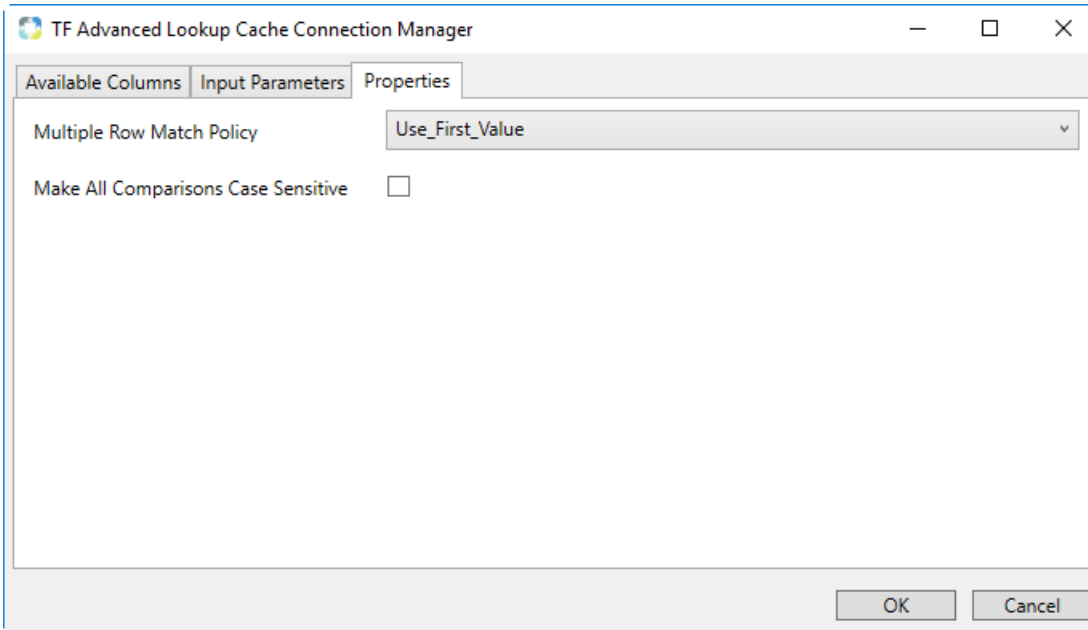


**Note:** The first condition requirement is the table key is equal to the input parameter.




Select the green **New Group Icon** at the end of the condition to create a new **And/Or** group.

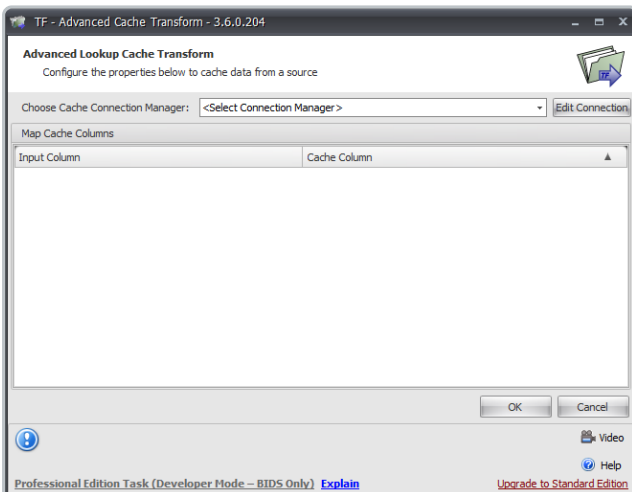
## Properties Tab



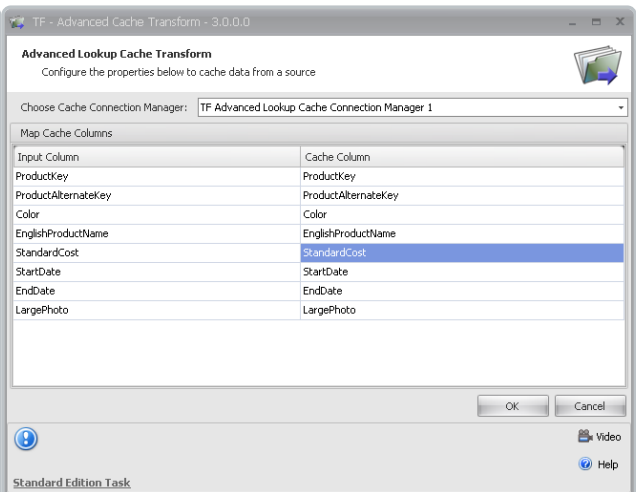
Option	Description
<b>Multiple Row Match Policy</b>	This option lets you define what should happen if multiple rows are found during a lookup.
<b>Make All Comparisons Case Sensitive</b>	Using this option makes all string comparisons performed case sensitive (e.g. SentryOne does not equal sentryone).

## Advanced Lookup Cache Transform

Transform Icon	Transform Description
	<p>The <b>Advanced Lookup Cache Transform</b> is used to store rows from a source into an in-memory or on-disk cache. See <a href="#">Advanced Lookup Cache Connection Manager</a> to learn more about setting up the connection manager.</p>



**Advanced Lookup Cache Transform Unpopulated**




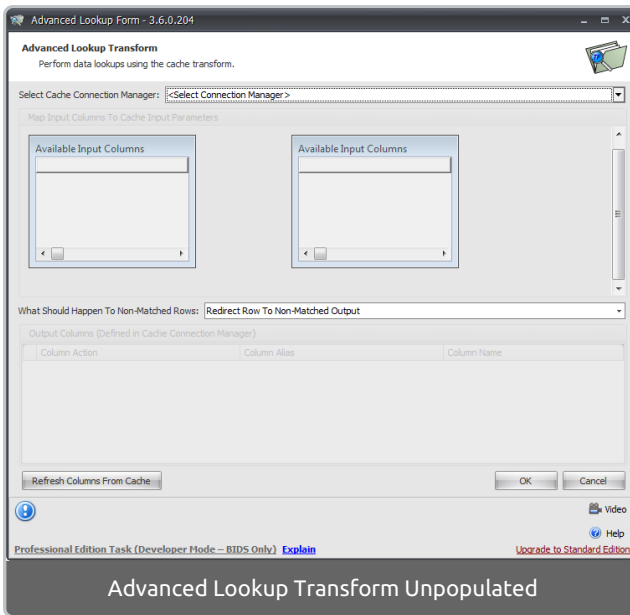
**Advanced Lookup Cache Transform Populated**

Option	Description
<p><b>Choose Cache Connection Manager</b></p>	<p>You can select an existing <b>Advanced Lookup Cache Connection Manager</b> or choose to create a new one.</p>
<p><b>Map Cache Columns</b></p>	<p>After a <b>Cache Connection Manager</b> has been selected, the input columns and columns from the cache are automatically mapped if the names match. Otherwise you need to manually map the</p>

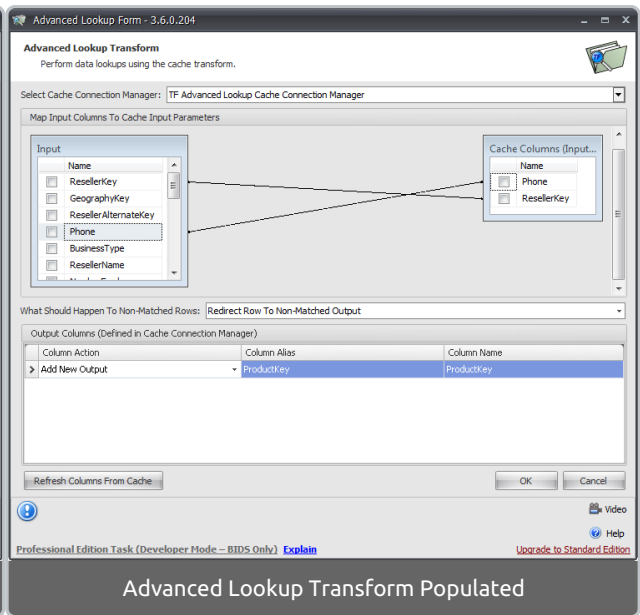
Option	columns. Description
--------	-------------------------

## Advanced Lookup Transform

Transform Icon	Transform Description
	<p>The <b>Advanced Lookup Transform</b> allows you to store a temporary cache of rows and perform lookups on that cache to pull that data into a data flow. There are two modes for the <b>Advanced Lookup Transform</b> :</p> <ul style="list-style-type: none"><li>• Connected</li><li>• Disconnected</li></ul> <p>See the Advanced Lookup Cache Connection Manager to learn more about setting up the connection manager.</p>



Advanced Lookup Transform Unpopulated



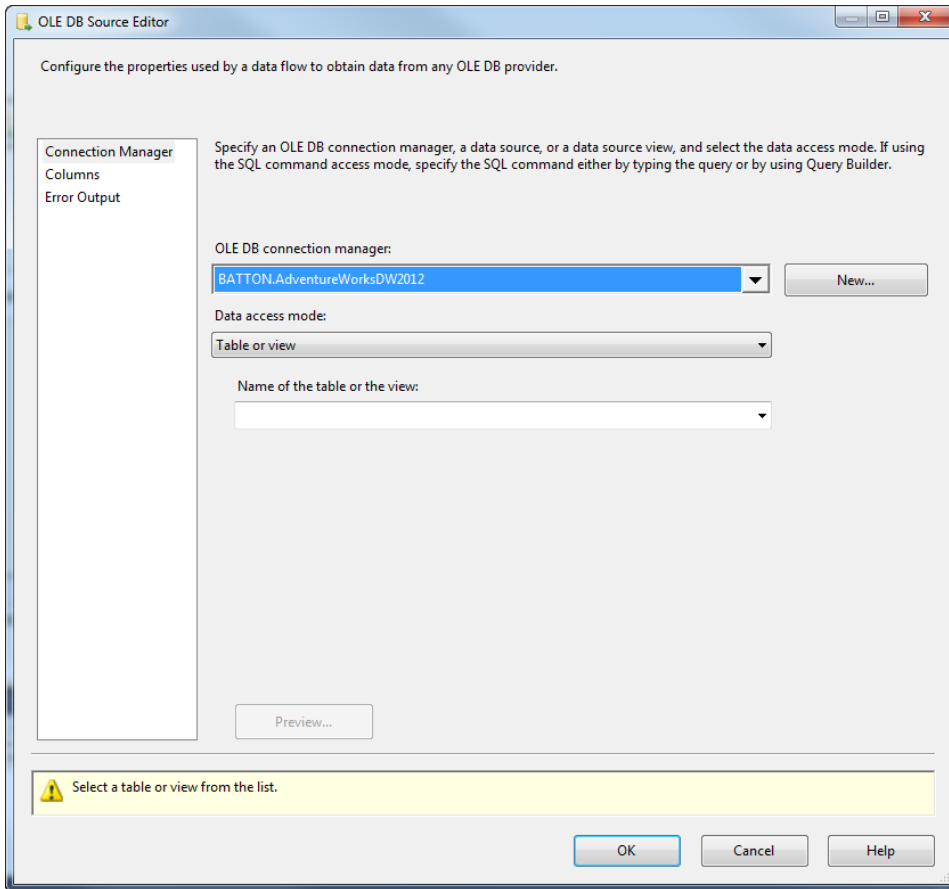
Advanced Lookup Transform Populated

Option	Description
<b>Select Cache Connection Manager</b>	Select an <b>Advanced Lookup Cache Connection Manager</b> . If you have not created one yet, you will need to create one. Once you have one selected, it automatically populates the available columns.
<b>Map Input Columns To Cache Input Parameters</b>	This is where you map input columns to the input parameters defined in the <b>Advanced Lookup Cache Connection Manager</b> .
<b>What Should Happen To Non-Matched Rows</b>	This tells the component what should happen when a non-matched row is found. <ul style="list-style-type: none"> <li>• <b>Fail Component</b> - This option fails the component when a non-matched row is found.</li> <li>• <b>Redirect Rows To Non-Matched Output</b> - This option redirects the rows to the non-matched output.</li> <li>• <b>Output Null Values To Output</b> - This option outputs NULL values for each of the output columns defined in the advanced lookup.</li> </ul>
<b>Output Columns (Defined in Cache Connection Manager)</b>	This section lists all of the output columns defined in the <b>Advanced Lookup Cache Connection Manager</b> . You have a few choices on how to output the values of the output columns based on the column action. <ul style="list-style-type: none"> <li>• <b>Add New Output</b> - This option creates a new output column based on the column alias</li> <li>• <b>Replace Input Column</b> - This option allows you to overwrite the current value of the selected input column with the value of the output column from the lookup cache.</li> </ul>
<b>Refresh Columns From Cache Button</b>	This button refreshes the columns from the cache connection manager.

## Setting up an Advanced Lookup Transform

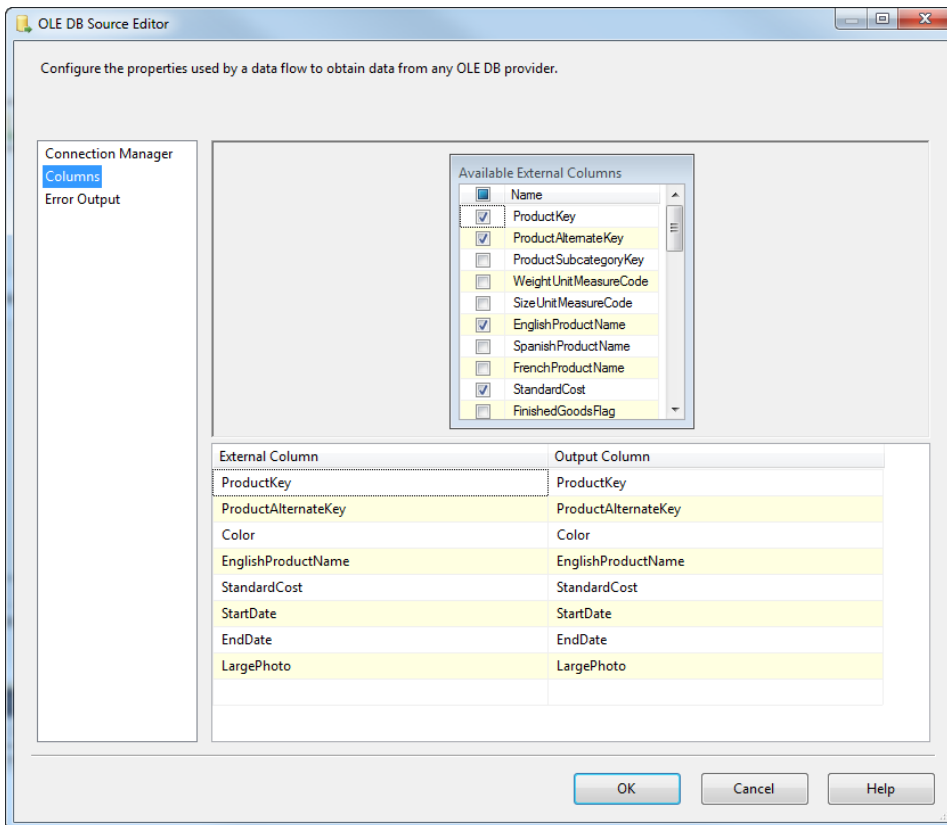
Complete the following steps to setup an **Advanced Lookup Transform**:

1. Create a data flow task, and then add an **OLE DB Source**. In this example we are using an OLE DB source connected to AdventureWorksDW. ([download here](#)).

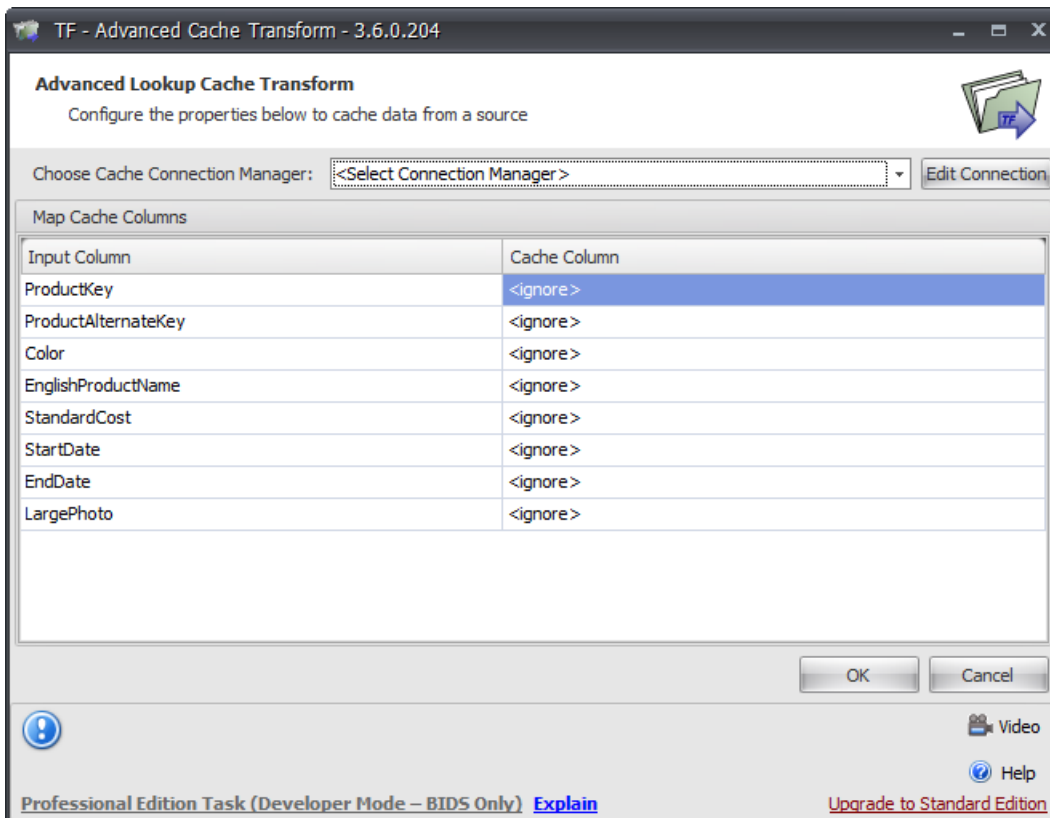


2. For this example select only the following columns from the columns tab.

- ProductKey
- Color
- StandardCost
- EndDate
- ProductAlternateKey
- EnglishProductName
- StartDate
- LargePhoto



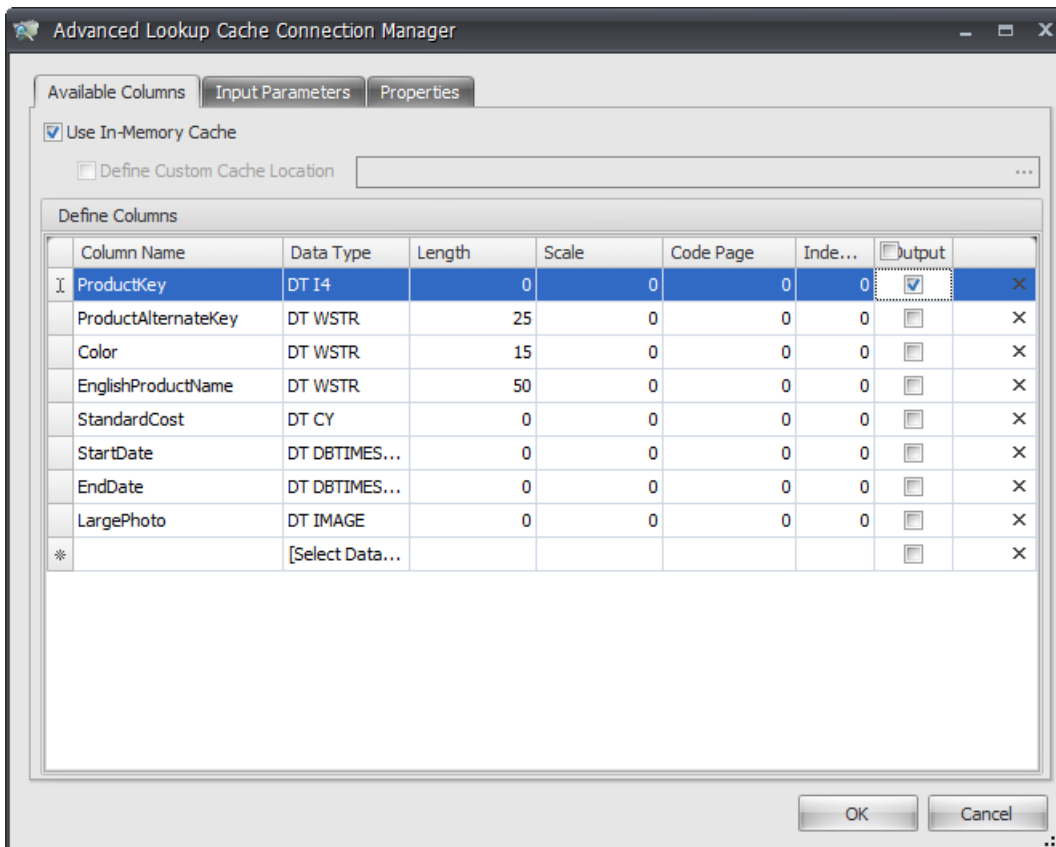
3. Now, create a lookup cache using the **Advanced Lookup Cache Transform**. Drag an instance of **TF Advanced Lookup Cache Transform** to the designer window. Attach the OLE DB Source Output to the **TF Advanced Lookup Cache Transform**. Open the transform by double clicking it, which opens the edit window.



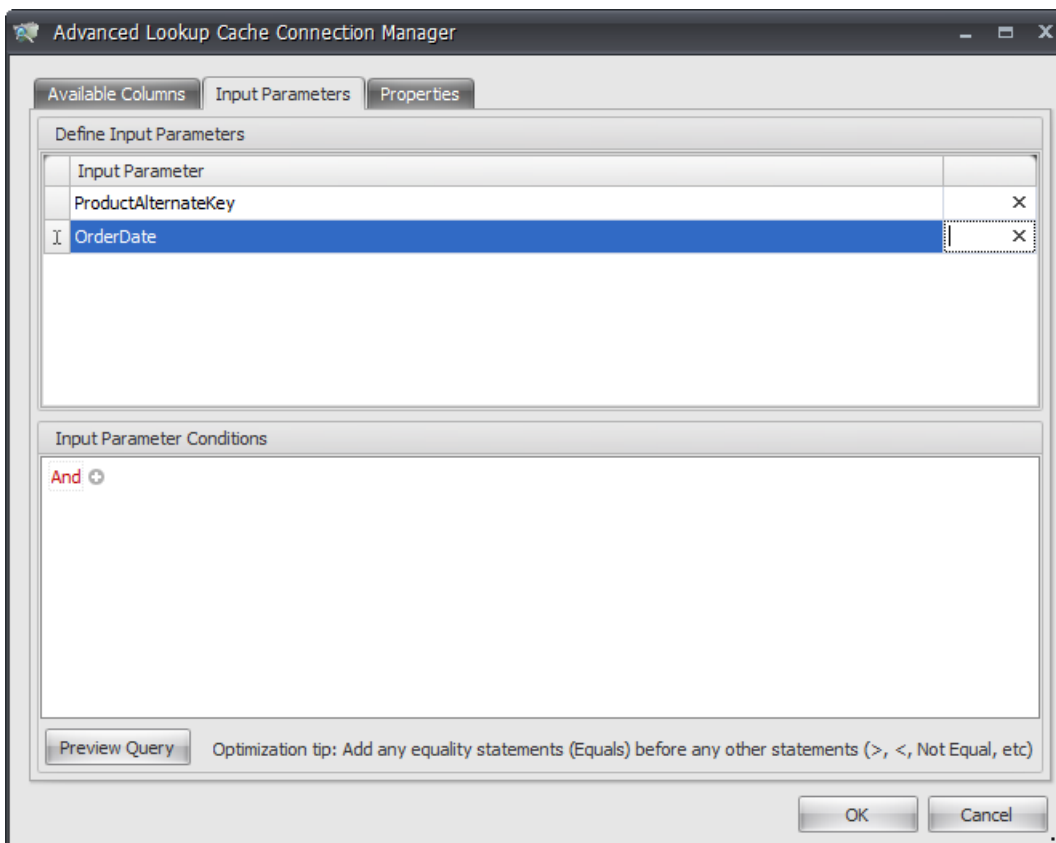
4. Select the **Choose Cache Connection Manager** drop down list and then choose **Create New Lookup Cache Connection...**



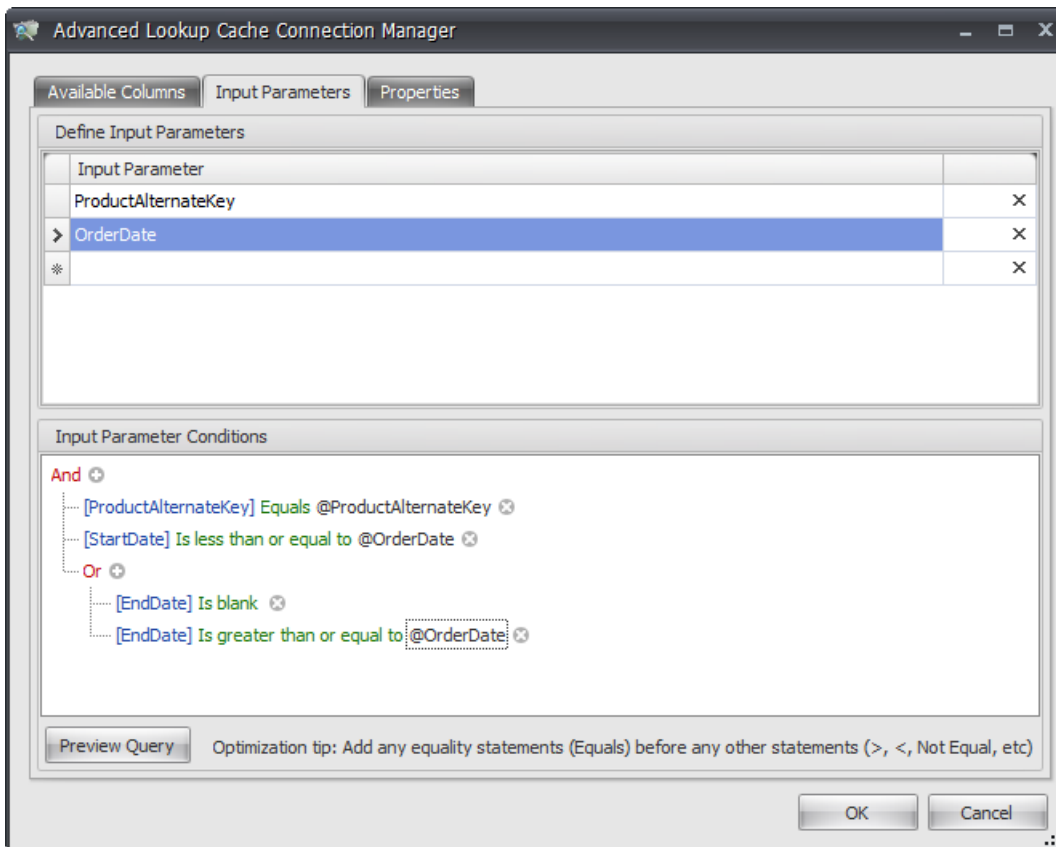
5. Setup the **Advanced Lookup Cache Connection Manager**. Then deselect all output except for **Product Key**.



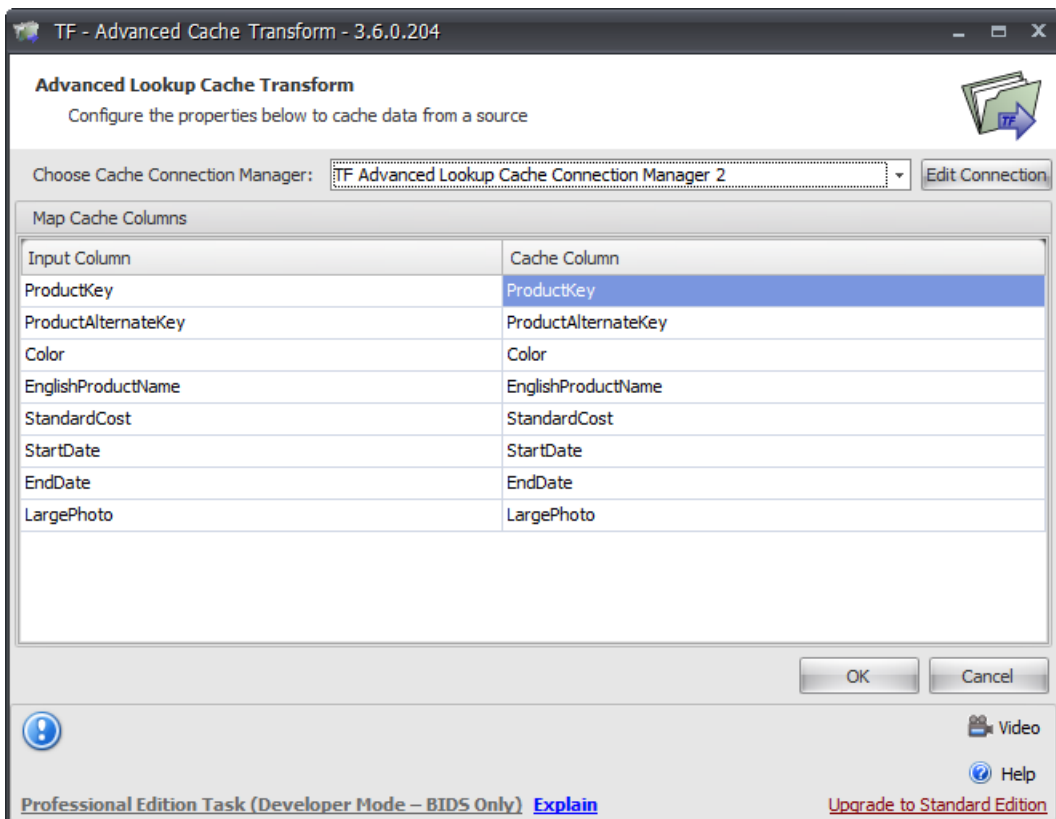
6. Next, select the **Input Parameters** tab, and then create two input parameters named **ProductAlternateKey** and **OrderDate**.



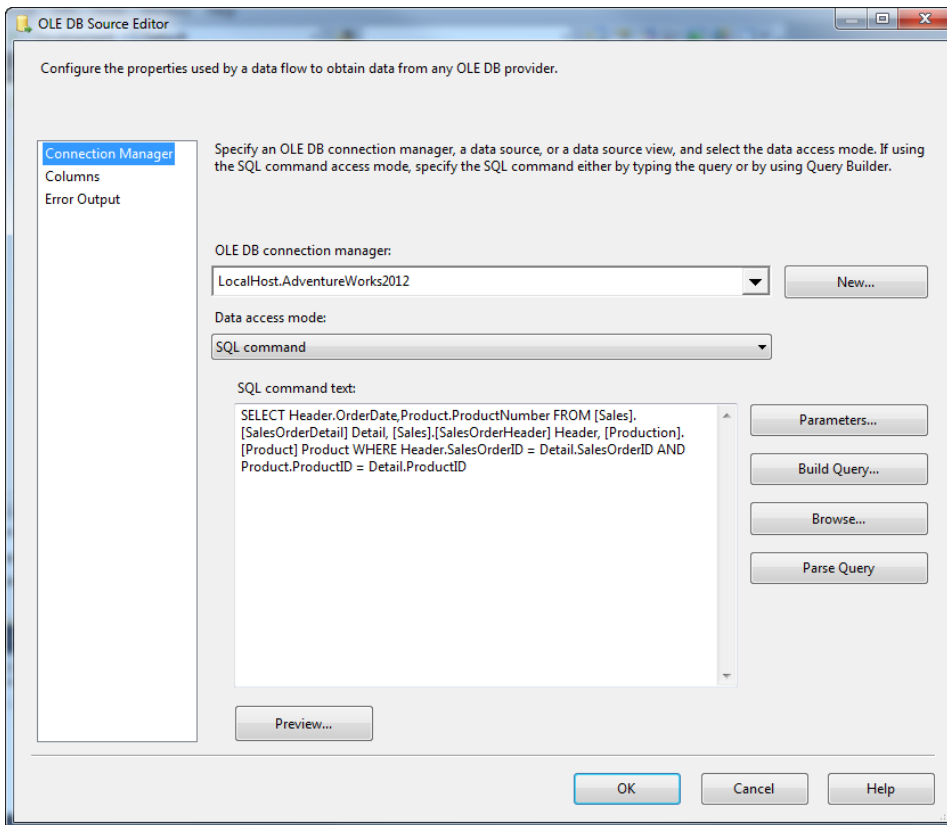
7. Then, create the following conditions: Note that **Is Blank = Is Null**.



8. Select **OK** and then your window should look like this:

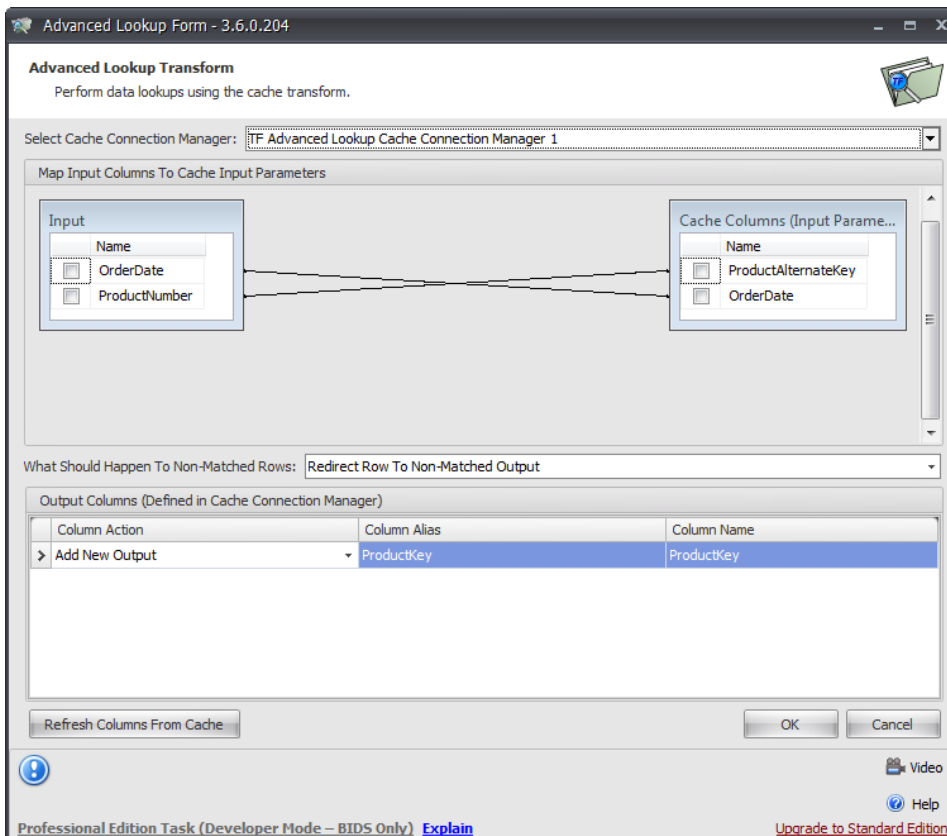


9. Add a new data flow task and a new OLE DB source, for this example we are connecting to AdventureWorks and creating a custom SQL query.

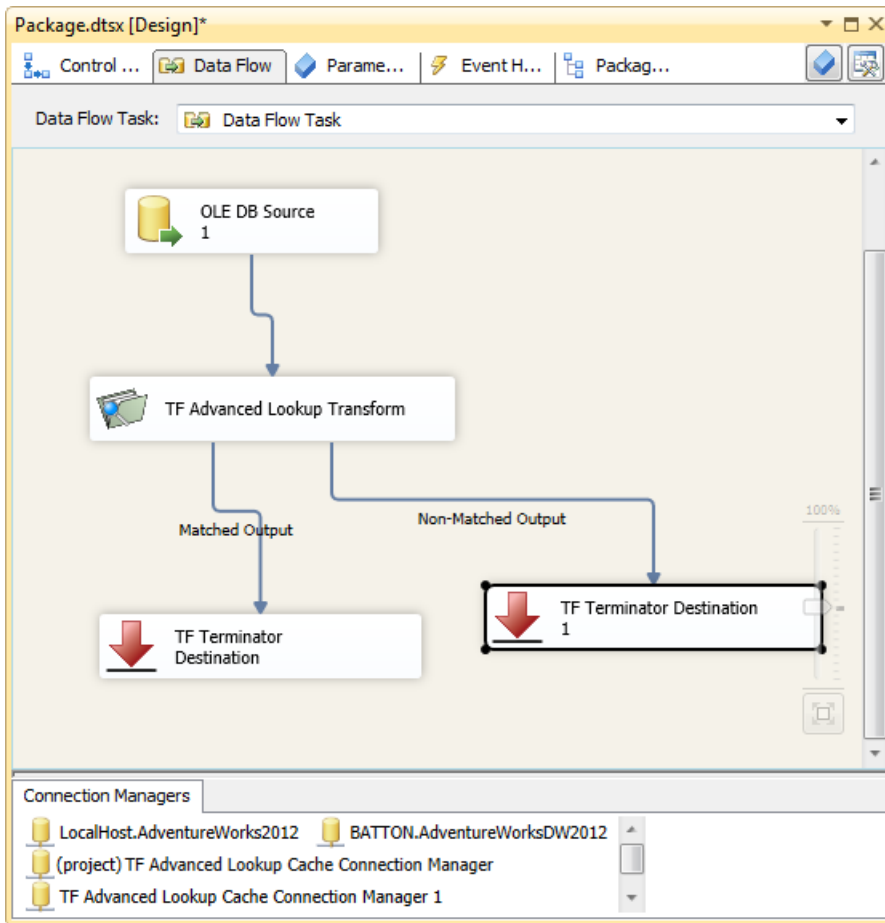


10. Add a new **TF Advanced Lookup Transform** to the designer, drag the output from the **OLE DB source** to the **TF Advanced Lookup Transform**. Double click the transform to open it. Once the **editor** window is open, choose the previously created Cache Connection Manager.

11. In the **Map Input Columns to Cache Input Parameters** window, connect **OrderDate** from the **Input column** to the **OrderDate** from the **Cache Column** by dragging one to the other. Do the same for the remaining field, and then select **OK**.



12. Drag your Matched and Non-Matched outputs to your desired destination. For this example we used TF Terminator Destinations for both.



## Disconnected Lookups

**Disconnected Lookups** allow you to use a cache to perform lookups without an **Advanced Lookup Cache Transform** in your data flow.

### Functions

There are two functions that can be used to execute disconnected lookups:

Function	Description
<p><b>LookupData(connectionManager As AdvancedLookupCacheConnectionManager, ParamArray args As Object)</b></p>	<p>Used to retrieve the FIRST column setup as an output column in the <b>Advanced Lookup Cache Manager</b> defined in the connectionManager parameter.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>connectionManager</b> - The name of the Advanced Lookup Cache Connection Manager.</li> <li>• <b>args</b> - Parameter array of input parameters defined in the <b>Advanced Lookup Cache Connection Manager</b>. Args matches the number of input parameters setup in the Advanced Lookup Cache Connection</li> </ul>

Function	Description manager defined in the connectionManager argument.
<p><b>LookupDataByColumn(connectionManager As AdvancedLookupCacheConnectionManager, outputColumnName As String, ParamArray args As Object)</b></p>	<p>Used to retrieve the value of the column defined in the <b>outputColumnName</b> parameter from the Advanced Lookup Cache Connection manager defined in the connectionManager parameter.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>connectionManager</b> - The name of the Advanced Lookup Cache Connection Manager</li> <li>• <b>outputColumnName</b> - The name of the column to retrieve from the output. The column defined here must be setup as an output column in the Advanced Lookup Cache Connection Manager.</li> <li>• <b>args</b> - Parameter array of input parameters defined in the <b>Advanced Lookup Cache Connection Manager</b>. Args matches the number of input parameters setup in the Advanced Lookup Cache Manager defined in the connectionManager argument.</li> </ul>