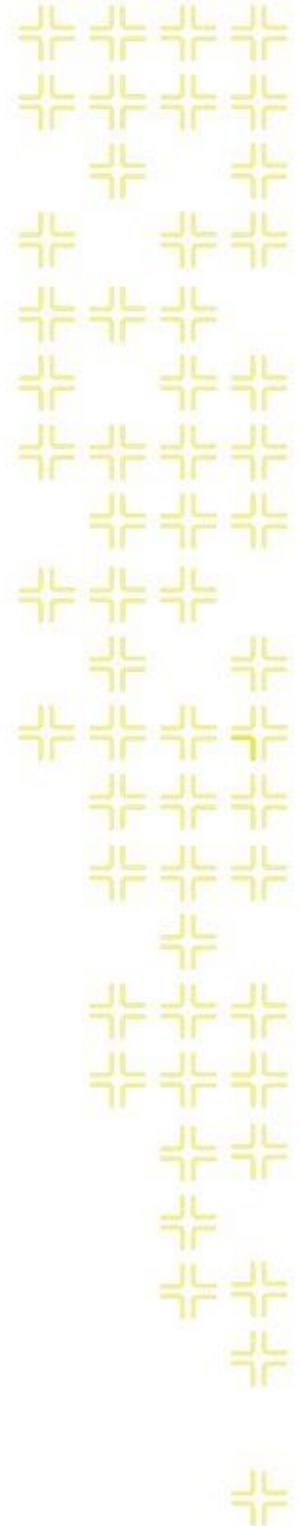




TRAINING GUIDE

Advanced Import & Update



Lucity – Advanced Import & Update Functionality

In this workbook, we will introduce you to advanced functionality in the Import and Update program. This includes formulas, Pre and Post Processing, importing from Web Services, importing and updating picklists, workstation specific settings, and helpful tips.

Table of Contents

- Formulas 2
 - Reserved Words..... 3
 - Example Formulas 3
- PreProcessing..... 5
 - File Download 5
 - External Process(es)..... 5
- PostProcessing 6
 - Every Record..... 6
 - Web Service 7
 - SQL Statement 7
 - Reserved Mappings 7
- Importing from Web Services 9
 - Building the Web Service Call 9
 - Web Service Schema 10
- Importing Picklists..... 11
- Workstation Settings..... 11
- Tips and Tricks..... 13
 - Displaying More Mapping Information 13
 - Increase Performance..... 13
 - Copy/Paste Correlations..... 14
 - Export Data 16
 - Global Updates 16
 - Prevent ReProcessing of Data 17

Formulas

The Import and Update allows you to utilize formulas in various locations. Formulas support three possible formats:

- Specifying an **Explicit Value** to be entered. This is achieved by typing the value as you would like it to appear in the field. The value, however, cannot start with a "+" or "=A" as these are reserved for the other two formats.
- Specifying a **Mapping** from the Source data. This is achieved by typing "=A" followed by a number which represents the column position in the Source Data. As an alternative to typing the mapping, the user can simply drag-and-drop from the Data Source grid to the Formula field in the Mapping grid. Ex. "=A5" indicates the data stored in the 5th column will be returned.
- Specifying a **Query** leveraging your Lucity Database Platform's built-in functionality. This is achieved by typing "+" followed by the database's syntax for the functionality you would like to perform. This allows for manipulating the source data, looking up values in the Lucity database, and utilizing Database Specific functions. The user may utilize Mappings within the query by specifying the Mapping notation, as specified above, within the query. Ex. "+LEFT('=A5', 5)" indicates the left 5 characters will be retrieved from the data stored in the 5th column.

Formulas are supported as follows:

Tab	Field(s)	Formats	Notes
Data Source	Data Select Sql Raw Web Request	Explicit Value, Query	Formula must be enclosed in double brackets (ex. [[+Getdate()]]) and cannot contain Mappings
Mapping	Formula	All	
PreProcessing	Program Arguments	Explicit Value, Query	Formula must be enclosed in double brackets (ex. [[+Getdate()]]) and cannot contain Mappings
PostProcessing	Program Arguments	All	Formula must be enclosed in double brackets (ex. [[+Getdate()]])

Reserved Words

Query Formulas support the use of Reserved Words to allow you to perform certain behaviors or utilize information pertaining to your Import and Update configuration. Reserved Words are enclosed in percent signs (%) to distinguish them and are replaced with the value they retrieve. The specific Reserved Words that can be used in a Formula vary and the best way to determine which Reserved Words you can be used is to right-click within a field that supports Formulas and review the list of Reserved Words.

The following lists the Reserved Words that are unique to the Import and Update. Additionally, the Reserved Words available in the Web for Filters and Data Drills are supported and behave identically.

%DocumentServerHive%	Automatically enters the path for the document server. This is taken from the Lucity Web > Admin Portal > Settings > System Settings > Documents > Path where uploaded documents are stored (Internal)(Document Hive) setting.
%DataSourcePath%	Automatically enters the path from your data source. This only is applicable if your Data Source references a file path.
%PromptForDocumentDirectory%	When a user begins an import, they will be prompted the location of their documents. This functionality is limited to the DocumentPath field in the Mapping table.
%ImportName%	Automatically enters the Name of the current Import and Update configuration
%ImportId%	Automatically enters the AutoID of the current Import and Update configuration. Note, that the Import and Update configuration needs to be Saved for this to behave properly.

Example Formulas

Source: 

#	Field Name	Value (A)
1	Column1	10.97
2	Column2	5
3	Column3	Concrete
4	Column4	PLK000078FD
5	Column5	20071023115523
6	Column6	Sewer Department

Description	Formula	Resolves To	Final Value
Map Source Data to Lucity	=A2	5	5
Insert a value into a Lucity Field	"2" or "1/1/2015" or "Bill"	2 or 1/1/2015 or Bill	2 or 1/1/2015 or Bill
Add values together	+=A1 + =A2	10.97 + 5	15.97
Concatenate values	+ '=A2 - =A3'	'5 - Concrete'	5 - Concrete
Extracting a value from the left of a string	+LEFT('=A4', 3)	LEFT('PLK000078FD', 3)	PLK
Extracting multiple values from a string and reformatting them together (dates)	+SUBSTRING('=A5', 5, 2) + '/' + SUBSTRING('=A5', 7, 2) + '/' + LEFT('=A5', 4)	SUBSTRING('20071023115523', 5, 2) + '/' + SUBSTRING('20071023115523', 7, 2) + '/' + LEFT('20071023115523', 4)	10/23/2007
Extracting part of a string and converting to a different format	+CAST(SUBSTRING('=A4', 4, 6) AS INT)	CAST(SUBSTRING(' PLK000078FD', 4, 6) AS INT)	78
Setting a value based on another value	+CASE WHEN '=A3' = 'Concrete' THEN '1' ELSE '2' END	CASE WHEN 'Concrete' = 'Concrete' THEN '1' ELSE '2' END	1
Insert the Current Date	+GETDATE()	GETDATE()	12/7/2015 10:48:50 AM
Set value based on the results of a lookup	+SELECT CT_ID FROM WKCAT WHERE CT_NAME = '=A6'	SELECT CT_ID FROM WKCAT WHERE CT_NAME = 'Sewer Department'	89
<ul style="list-style-type: none"> ▪ Example A 			
<ul style="list-style-type: none"> ▪ Example B 	+SELECT DESCRIPTION FROM WKFIELDSDESC WHERE ID = 841 AND CODE = =A2	SELECT DESCRIPTION FROM WKFIELDSDESC WHERE ID = 841 AND CODE = 5	Vacation Time
<ul style="list-style-type: none"> ▪ Example C 	+SELECT SUM(NT_LENGTH) FROM SWNET WHERE NT_MAT_TY = '=A3'	SELECT SUM(NT_LENGTH) FROM SWNET WHERE NT_MAT_TY = 'Concrete'	1023

PreProcessing

The PreProcessing tab provides a way to configure processes which must occur prior to the Import retrieving source data. This can include downloading a file or running external applications which may be necessary in order for the Import and Update to correctly process data.

File Download

The File Download section allows for configuring the download of a file from Amazon's S3 repository or from an FTP site. The file will be copied to the location specified in the Data Source field on the Data Source tab and renamed to match.

Downloading from Amazon S3 targets Software as a Service (SaaS) customers that are hosted by Lucity and provides a mechanism to transfer files to the cloud, so they can be processed.

File Download:

Download Source: Delete Source File after Download

Property Name	Value
Bucket	import-test-lucity.com
Region	us-east-1
KeyName	import/test.sql
Prefix	
AccessKey	
SecretKey	

Required

General | Data Source | Mapping | Correlations | PreProcessing

Data Source Type:

Data Source:

Data Delimiter: Tab Space Other:

The test.sql file will be downloaded as C:\Test\Data.txt

External Process(es)

The External Processes(es) section allows for calling external processes prior to data retrieval but after the File Download process. This allows moving or renaming files, calling external applications to potentially generate or retrieve data that will be processed, etc.

The below image provides an example of calling the DOS xcopy command to copy the ACKKEY.txt file and for the copy to be named ACKKEY_20180821.txt. This example includes a Formula in the Arguments field to retrieve the current date and append it to ACKKEY.

External Process(es):

Name	Order	Program	Arguments
Formula1	1	xcopy	C:\temp\Example\ACKKEY.txt C:\temp\Example\ACKKEY_[[+CONVERT(VARCHAR, GETDATE() , 112)]] .txt
*			

PostProcessing

The PosProcessing tab provides a way to configure processes which must occur after the import process has completed. The need for this could range from performing file copy/move/delete operations, updating the Data Source to reflect the data has been imported, running additional processes, or even exporting data from Lucity.

The External Process(es) section resembles the same section in the PreProcessing tab, however, it differs in some important ways:

- Allows for processes to be called for each row of data that was retrieved from the Data Source. This is controlled by the **Every Record** checkbox. If the **Every Record** checkbox is unchecked for a process, the process will only be called once.
- Allows for Web Service calls to be made, allowing updates to external products. This is controlled by the **Web Service** checkbox and the appropriate Web Service information in the Program field.
- Allows for executing SQL statements to the Data Source, allowing updates to external products. This is controlled by the **SQL Statement** checkbox. Note, this checkbox will not be present if the Data Source does not support SQL statements.
- Supports utilizing Mappings within Formulas. If your Program or Arguments values utilize a Formula, that Formula can reference the source data in the same manner as the Formula field on the Mappings tab. Utilizing Mappings is limited to processes running on Every Record.
- Supports Reserved Mappings within the Program and Arguments fields allowing access to the data that is in the Lucity Business Object associated to the Source Data. Utilizing the Reserved Mappings is limited to processes running on Every Record.

General Data Source Mapping Correlations PreProcessing PostProcessing Processed								
External Process(es):								
	Name	Order	Program	Arguments	Every Record	Web Service	SQL Statement	
▶	Mapping	1	xcopy	C:\Temp\Example\ACKEY.txt C:\Temp\Example\ACKEY[!=A1].txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	MappingAndFormule	2	xcopy	C:\Temp\Example\ACKEY.txt C:\Temp\Example\ACKEY[+[SYSTEM_USER + '=A1']].txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Property	3	xcopy	C:\Temp\Example\ACKEY.txt C:\Temp\Example\ACKEY%%WorkOrderNumber%%.txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
*					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Every Record

When this option is selected, the configured external process will run on every record that was retrieved from the Data Source and successfully processed. If this option is not selected, then the external process specified will run only once per run.

Web Service

When this option is selected, the configured external process will be handled as a Web Service call. The Web Service call must reside in the Program field and anything in the Arguments field will be ignored. Web Service call must be properly formatted and consist of the Method, URI, Protocol, Headers and Body. Reference the section on configuring the Import and Update for importing from Web Services for an explanation of these elements.

SQL Statement

When this option is selected, the configured external process will be assumed to be a sql statement and will be run on the configured Data Source on the Data Source tab. This option is only available if the Data Source supports running sql statements. The sql must reside in the Program field and anything in the Arguments field will be ignored. If multiple statements are to be run, all statements can be included in the Program field as long as the proper Data Source syntax is followed (i.e., each statement being separated by a semicolon).

Reserved Mappings

If an External Process is configured to run on Every Record, Reserved Mappings can be utilized to access the data in the Lucity Business Object for the row of data that was being imported. For example, if an Observation record was imported for a Sewer Pipe Inspection, the data for the resulting Observation record, the parent Inspection record and the Sewer Pipe can all be accessed.

To utilize the Reserved Mappings, it is important to understand the structure of a Lucity Business Object. When the Import and Update program processes a record, it retrieves a single "root" Business Object. This "root" Business Object can have "child" and "grandchild" objects. For example, a Sewer Pipe record is a "root" Business Object, an Inspection record is "child" object and an Observation record is a "grandchild". If you are talking about the Observation object, then the Inspection record would be its "parent" and the Pipe would be the "root".

Reserved Mappings are enclosed in double percent signs and refer to the Property Name of the Lucity Business Object. The Property Name is what you would see on the Mapping tab in the "Property Name" column. Unlike the standard Reserved Words and Mappings, the Reserved Mappings do not need to be part of a formula. There are three Reserved Mapping syntaxes as show below.

NOTES: _____

Reserved Mapping	Description
%%Property_Name%%	This syntax allows for access to the Properties of the Root Business Object. For example, if the Destination Component is "Sewer TV Observations", the Root is "Sewer Pipe". Replace "Property_Name" with the appropriate Property Name you wish to access
%%ParentBO.Property_Name%%	This syntax allows for access to the Properties of the Parent Business Object. This would be the Parent module of the Module selected on the General Tab for the Destination Component. For example, if the Destination Component is "Sewer TV Observations", the Parent is "Sewer TV Inspection". Replace "Property_Name" with the appropriate Property Name you wish to access
%%CurrentBO.Property_Name%%	This syntax allows for access to the Properties of the Business Object selected on the General Tab for the Destination Component. Replace "Property_Name" with the appropriate Property Name you wish to access

The below image shows an example of how the Reserved Mappings might be utilized. This example is making a Web Service call to update an external system to indicate a Lucity Work Request was generated from their data. %%WorkOrderUser24%% refers to the WorkOrderUser24 property of the Lucity Request and stores the external Service Order number. The %%RequestNumber%% refers to the RequestNumber property and is being sent to reflect the Lucity Request that was generated.

External Process(es):

Name	Order	Program	Arguments	Every Record	Web Service
Update CIS Request ID	1	PATCH https://cisapi.ocean.local/data/serviceorder/%%WorkOrderUser24%% HTTP/1.1 User-Agent: LucityImportAndUpdate Authorization: Basic TPVDSVRZ0lpwX2w4N0BrS3gylw== Host: cisapi.ocean.local Content-Type: application/json	[[{"op": "replace", "path": "/ServiceRequestID", "value": "%%RequestNumber%%"}]]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
*				<input type="checkbox"/>	<input type="checkbox"/>

NOTES: _____

Importing from Web Services

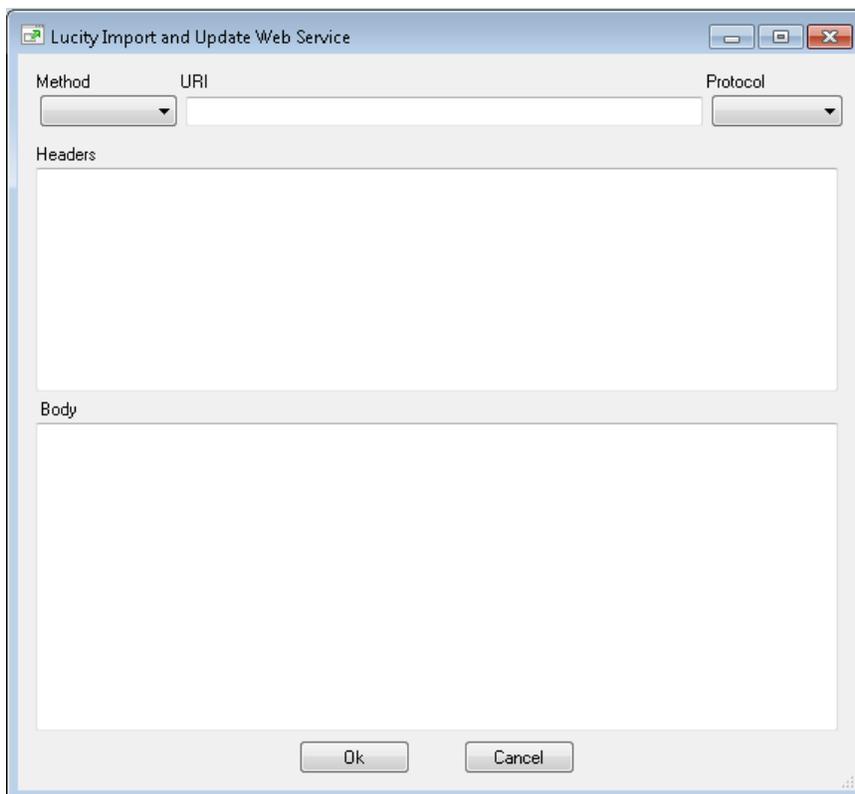
The Import and Update program supports importing data from Web Services. This allows for the Import and Update program to access data directly and remotely from third party systems. Additionally, the data retrieved is formatted and presented in the manner the external application intended.

Due to highly variable nature of Web Services, the responsibility of creating a properly formatted Web Request is put on the end user. However, the Import and Update does have a builder form to help facilitate building a Web Request.

Note that the Import and Update will only make a single Web Service call to retrieve data. Many Web Services allow for authentication information to be passed within the call to retrieve the data, however, some Web Services require a separate call to retrieve a login token or cookie. If a Web Service requires a separate authentication call, then Import and Update will not support connecting to these Web Services.

Building the Web Service Call

To import from a Web Service, select the **Web Service** item from the Data Source Type dropdown. If you already have a prepared Web Service call, you can paste it in the Raw Web Request field. If you need to build the Web Service call, then press the Build button. You will see the following dialog:



The dialog box is titled "Lucy Import and Update Web Service". It features a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is divided into several sections: "Method" (a dropdown menu), "URI" (a text input field), and "Protocol" (a dropdown menu). Below these is a large text area labeled "Headers". At the bottom is another large text area labeled "Body". At the very bottom are "Ok" and "Cancel" buttons.

Method	The Method is the verb, or action the Web Service Call is going to perform. The dropdown supports either GET or POST as these are the most common Methods used to retrieve data from a Web Service.
URI	The URI is the url or path to the Web Service Call. For a REST Service call, the URI could have variables included
Protocol	The Protocol indicates the version of HTTP protocol to utilize. The dropdown supports either HTTP/1.0 or HTTP/1.1. More than likely this will be HTTP/1.1
Headers	Headers are additional information that is to be passed to the Server about the request. This includes Authorization, format of the request, expected format of returned data, etc.
Body	The Body is the payload of data being sent to the Server. This would not be populated for REST service calls, however, may be populate for SOAP or other types of service calls. If populated, this must be formatted in the syntax that the Server expects or as described in the Headers section.

For a more complete explanation of the above items, please refer to HTTP Standards for Requests (<https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html>)

Once the appropriate Web Service properties are specified, pressing Ok will return you to the DataSource tab and the Raw Web Request field will be populated with the information from the Build form. You can edit the information in the Raw Web Request field.

Note, that if your Web Request requires a Method or Protocol not in the Build form's dropdown fields, you should not return to the Build form as it will remove those items from your Request.

Web Service Schema

The Schema is the structure of the data returned from the Web Service. When retrieving data, there is the possibility that the Schema may change from one call to the next and this varies based on the Web Service. For example, some Web Service responses may only include fields that are populated or if a child object exists. To allow for you to map the data, it is therefore necessary to store the full, expected Schema of the Web Service. It is important to note that the Web Service Request being used to retrieve the Schema retrieve an object(s) that will contain the expected fields, children, etc.

To retrieve the Web Service Schema, simply press the Retrieve Schema on the Data Source tab once the Raw Web Request is properly constructed. This will retrieve the data from the Web Service and store the Schema in the Web Service Schema grid. The user can manually add or modify items in this grid. Additionally, if the Web Service changes or adds items to its Schema, pressing the Retrieve Schema button will retrieve the Schema and prompt you if you would like to add any additional Schema items to the end of the list or if you would like to completely refresh the Schema. If you elect to completely refresh the Schema, it will be necessary for you to verify your mappings as the columns may no longer be in the same order.

You may notice that the Column Names listed in the Schema grid do not match the names indicated in the actual Web Service Schema. Rather, the names will have additional qualifiers added to them such as "Root_Row_". This is to accommodate property names that are identical but reference different objects within the Schema.

Importing Picklists

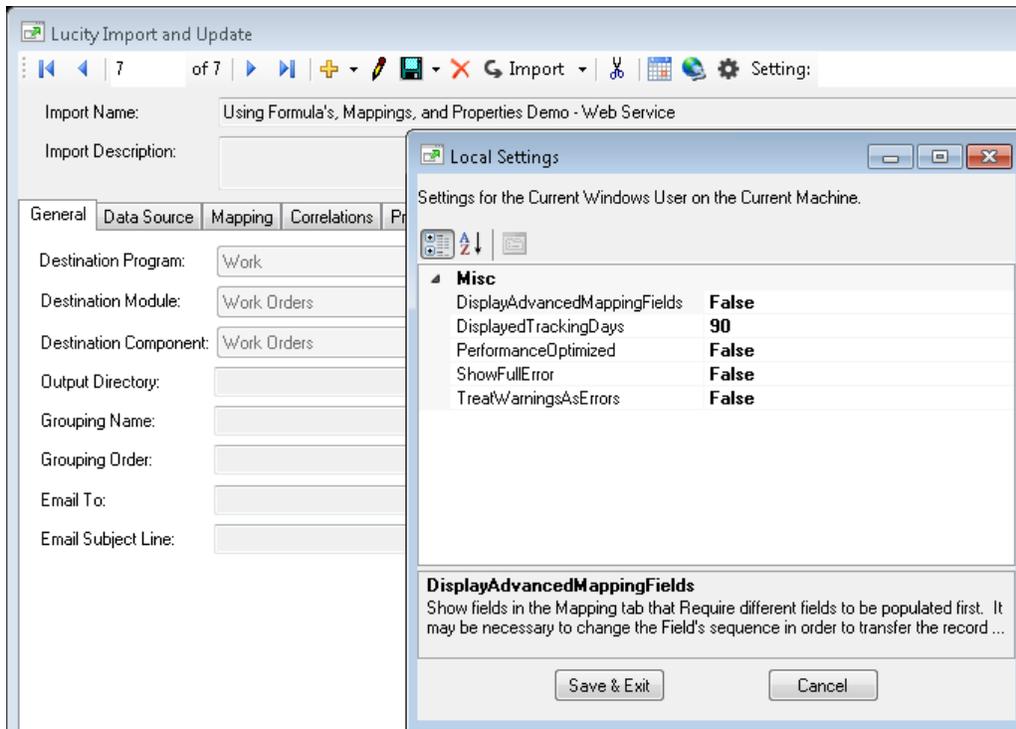
The Import and Update program supports imports for populating picklists in Lucity. To indicate a Picklist import, the Picklist Import checkbox on the General tab must be checked. All other requirements for configuring an Import are still necessary such as selecting the Module, checking the appropriate Add or Update checkboxes, and configuring your Data Source.

The Mapping tab will be limited to only list the Code and Type fields that you can import into, as well as a single GISCode field. The Import and Update supports imports into a single picklist per run, so only one Code field and its partnered Type field can be mapped, as well as the GISCode field.

During the execution of the Import and Update, if a code already exists for the value specified, the process will attempt to update the Type portion and the GISCode. If a code does not exist, a new entry is created with the Code, Type and GISCode value. There is no validation to prevent duplicate Type values in the Picklist as the unique identifier is considered to be the code value.

Workstation Settings

When running the Import and Update program, there are several local settings that control the way the program displays errors, processes data, etc. These settings are stored in the Lucity.ImportAndUpdate.xml file located in current Windows user's profile in the AppData folder (i.e., C:\Users\cwright\AppData\Roaming\Lucity). These setting can be changed directly in the xml file, but the Import and Update now allows for access of these settings via a Local Settings dialog as shown below. Do not change these settings without checking with Lucity first.



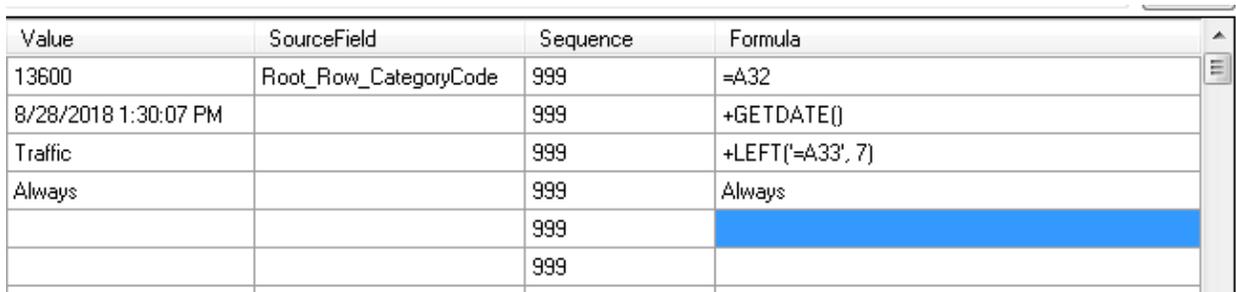
DisplayAdvancedMappingFields	Setting this option to "true" may display additional fields in the Mapping grid for a module that contains fields that are "Conditionally Writeable". For example, a module may require one field to be populated prior to another field becoming enabled. This second field would not normally show up in the Mapping grid as, by default, only writeable fields are displayed in the Mapping. If this option is used, please notify Lucity as to ensure there are no issues with mapping to the field.
DisplayedTrackingDays	Changing this option will indicated how many days of processed data will be displayed in the Processed tab.
PerformanceOptimized	Setting this option to "true" will result in the Import and Update bypassing certain checks, and therefore should only be set when the source data is pretty much guaranteed to not have any "problems" in it. If problematic data is being imported, setting this option to "true" may make it quite difficult to identify the source of the problem. Note that setting this option to "true" should still not allow problematic data into the system, it will just make it difficult to identify where the problem is. This option should only be set to "true" if experiencing extremely long import times.
ShowFullError	Setting this option to "true" will display much longer error messages than normally displayed. This should only be set for troubleshooting purposes. The longer message contains the call stack from the code.
TreatWarningsAsErrors	This option is defaulted to "false". Setting this option to "true" will treat Warnings as Errors. When using a Lucity application, the user may perform an action which results in the application asking the user to verify their action. These would be "Warnings". By Default, the Import assumes the user would have always elected to proceed with the action. By setting this option to "true", the Import will treat these Warnings as errors and not allow the record to be Imported.

Tips and Tricks

Displaying More Mapping Information

By default, the Destination Grid on the Mapping Tab will display the Property, Table, Field, Caption, Mask and your Formula. By Right-Clicking on the Grid, you can show additional columns as well as elect to hide existing fields. The settings do not persist, however, and will return to the default once the Import and Update program is closed. The additional columns that can be displayed are:

- Value** The value column will resolve your formulas allowing you to verify your formula is correct and returns the value you are expecting.
- SourceField** The SourceField column will display the column header from the Source grid so you can verify that you have mapped the correct column. A value is displayed only if a single Mapping is specified
- Sequence** The Sequence column identifies the order the column will be populated by the Import and Update process. By default, the value is 999, meaning that the column is not prioritized. It may be necessary to set a Sequence value if a field in a module requires another field be populated first. This column is mainly used for troubleshooting purposes.



Value	SourceField	Sequence	Formula
13600	Root_Row_CategoryCode	999	=A32
8/28/2018 1:30:07 PM		999	+GETDATE()
Traffic		999	+LEFT(=A33', 7)
Always		999	Always
		999	
		999	

Increase Performance

If you have a complicated Import to run and are not importing from a Text or XML file, it may considerably speed up the processing time if you can rectify the value in your Data Select Sql or Raw Web Request on the Data Source tab as opposed to using query formulas in the Mapping tab.

When the Import and Update runs, any query formulas in the Mapping tab (designated with a leading plus "+") are resolved for each record being processed. If you are importing 1,000 records and have a Mapping of "+GETDATE()", then a sql statement is sent to the database 1,000 times to rectify the value. However, if you include "GETDATE()" in your Data Select Sql, then the value only has to be resolved once and can be mapped to the appropriate field.

Copy/Paste Correlations

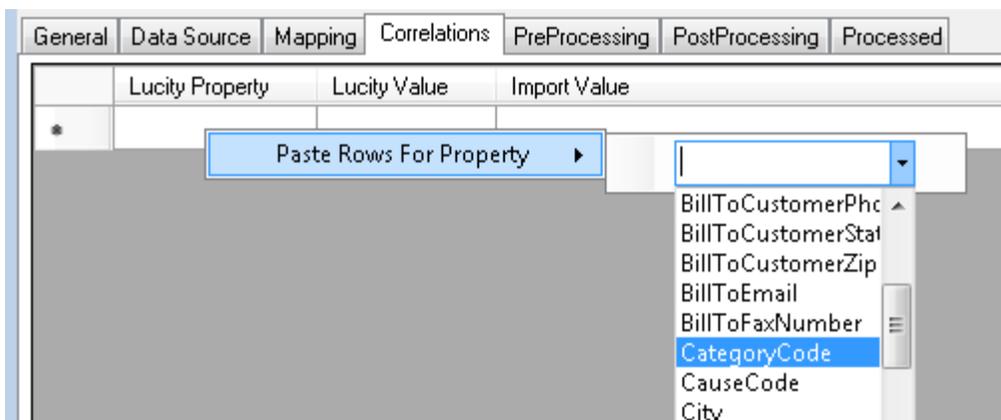
If you must configure numerous correlations in the Correlations tab, it may become very tedious and error prone to manually enter the correlations. If you have a spreadsheet or delimited text file, you can copy and map the values into the Correlations grid via a special Paste dialog. The Paste dialog is very similar to the Mapping tab in the Import and Update and you are essentially Mapping the data you copied to the appropriate fields in the Correlations grid.

To use this functionality, your source data must have the source value(s) and the Lucity value(s) to correlate. If you are adding correlations for multiple Lucity Properties, then it would be beneficial for the source data to have the Lucity property name, otherwise you will need to repeat the process for each Property you want to configure correlations for.

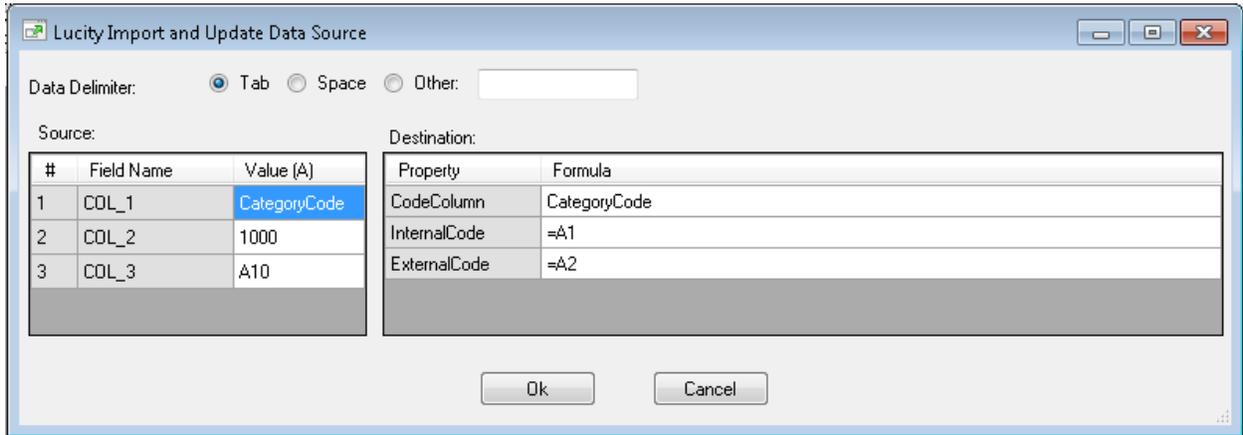
For Example, if you had a spreadsheet resembling:

	A	B	C
1	LucityProperty	LucityValue	PublicStuffValue
2	CategoryCode	1000	A10
3	CategoryCode	2000	B10
4	ProblemCode	10	5
5	ProblemCode	11	8

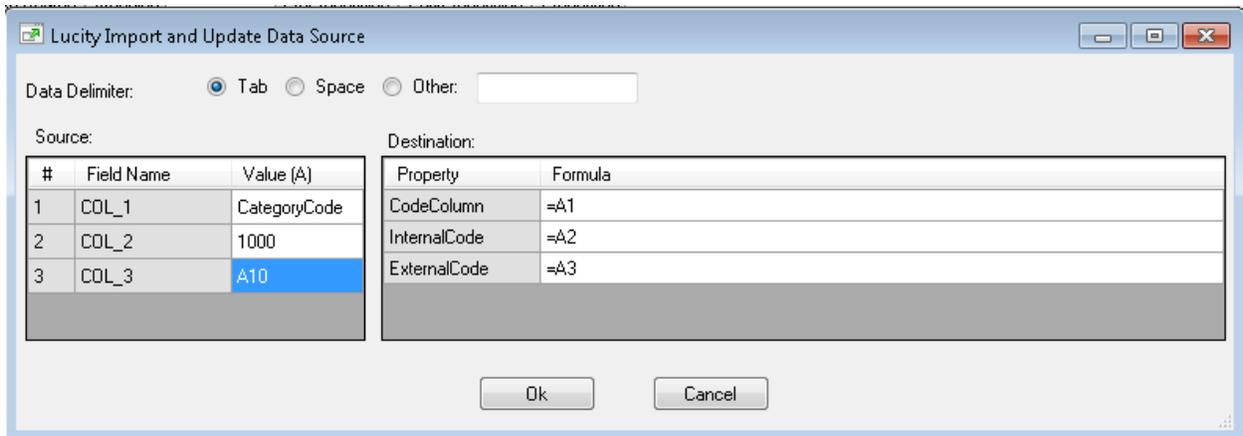
You would select the data and copy it to the clipboard (pressing Ctrl+C or Right-click and Select "Copy"). Do not Copy the Column Headers. Then in the Import and Update program, enter Edit mode and on the Correlation's grid Right-Click and select "Paste Rows For Property" and in the Dropdown, select "CategoryCode"



You will see a dialog like the below. In the case of this example, it would be necessary to change the Destination Formulas as the dialogue defaults to populating correlations for a single Property.



Just like on the Mapping tab, you can simply drag-and-drop to change the mappings.



Pressing OK will then add the information to the Correlations grid.

General	Data Source	Mapping	Correlations	PreProcessing	PostProcessing	Processed
	Lucity Property	Lucity Value	Import Value			
	CategoryCode	1000	A10			
	CategoryCode	2000	B10			
	ProblemCode	10	5			
	ProblemCode	11	8			

Export Data

In addition to Importing and Updating, the program can be used as an Export tool. While not explicitly designed for this, the addition of the PostProcessing tab made it possible to Export data.

To Export data, you would select the Module you will be wanting the data from in the General tab and uncheck the "Allow Adding New Records" and "Allow Updating of Existing Records" options to ensure you do not accidentally import any data.

On the Data Source tab, configure the Data Source to retrieve the Lucity objects you are wanting to export. This can be accomplished by referencing an external list or setting your Data Source to be the Lucity database and directly query the table.

On the Mapping tab, map data to the Required and Unique identifier fields only.

On the PostProcessing tab, you would then configure an External Process to write the appropriate data to a text file, or export it to a database or a Web Service. Refer to the PostProcessing section for specifics.

Below is an example of how you might export the data to a Text file using DOS commands. This example assumes that you wrote a query in the Data Source to select the columns you wish to export. However, you have many options here as you can reference the data directly from the Lucity Object as mentioned in the PostProcessing section of this document.

External Process(es):

	Name	Order	Program	Arguments	Every Record	Web Service	SQL Statement
▶	Export 1	1	cmd	/C echo [[*=A98, =A19, =A22]] >> C:\TEMP\TEST.TXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Global Updates

While Lucity Web and Lucity Desktop allow you to perform Global Updates, the Import and Update can be used to perform these operations as well and provides more robustness in doing so. Consider utilizing the Import and Update in the following scenarios:

- It is necessary to perform Global Updates repeatedly and/or on Schedule
- A Global Update needs to be performed that does Calculations. For example, if you would like to sum the values of two other fields to populate a third field.
- If the data for a Global Update needs to be manipulated or if other functionality that the database supports must be performed. For example, if you want to Trim spaces, populate fields with the Current Date, split data, etc.

Prevent ReProcessing of Data

In some cases, the Source Data may have data that has previously be processed. For example, if your Data Source is a Web Service that does not support filtering, you may retrieve a complete list of records every time and only want to process the new records. There are several solutions to prevent reprocessing of the data.

One solution is to use "Track Processed Records" option on the General tab. This will require that you Map data to the "TrackingIdentifier" field in the Mapping tab, so that a unique identifier can be stored. Whenever a record is processed, it will then result in a Tracking Record being written to the Lucity database. Depending on your situation, you have two options to prevent reprocessing:

- 1) Add a Formula to the "SkipProcessingIfNotBlank" Property in the Mapping grid that references the Tracking table (CMIMPROCESSED) that returns a value if the record has already been processed. This formula could resemble "**+SELECT TOP 1 PRC_ID FROM CMIMPPROCESSED INNER JOIN CMIMPORTUPDATE ON IMP_ID = PRC_IMP_ID WHERE IMP_NAME = 'Your Import Name' AND PRC_UNIQUE_ID = '=A12'**" where =A12 would be the same mapping you used for the TrackingIdentifier property.
- 2) If your Data Source is a database and it has access to the Lucity database, you may be able to add a subquery to exclude the records that have been written to the Tracking table (CMIMPROCESSED). For example, your Data Source query might resemble "**SELECT * FROM DATASOURCE WHERE DATASOURCE_ID IS NOT NULL AND DATASOURCE_ID NOT IN (SELECT PRC_UNIQUE_ID FROM Lucity.dbo.CMIMPPROCESSED INNER JOIN Lucity.dbo.CMIMPORTUPDATE ON IMP_ID = PRC_IMP_ID WHERE IMP_NAME = 'Your Import Name')**"

Another solution to prevent reprocessing of data is use the "SkipProcessingIfNotBlank" Property in the Mapping tab to return a value based on if the object already exists in Lucity. For example, if an Import is generation Lucity Requests, the mapping for the "SkipProcessingIfNotBlank" Property might have a formula similar to "**SELECT RQ_ID FROM WKREQ WHERE RQ_USERS = '=A5'**" where =A5 might be a unique identifier in the Source Data.

NOTES: _____

