Import Tool

For Dynamics CRM 2015

User Guide

Agreement as to warranties, limitation of liability, remedies and damages will survive termination.

5. LIMITED WARRANTY

THIS SOFTWARE IS LICENSED "AS IS." THE COMPANY MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY OF FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. THE COMPANY DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO RESULTS AND PERFORMANCE OF THE SOFTWARE, AND ITS INTERACTION WITH OTHER EQUIPMENT OR SOFTWARE OWNED OR USED BY YOU, IS ASSUMED BY YOU. SOME TERRITORIES DO NOT PERMIT THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL THE COMPANY, AND ITS DIRECTORS, OFFICERS OR AGENTS (COLLECTIVELY THE COMPANY) BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR THE INABILITY TO USE THE SOFTWARE EVEN IF THE COMPANY INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Table Of Contents

What's in this manual

DPS Import Tool is a powerful application designed specifically to import data into Microsoft Dynamics CRM. It can easily handle complex import scenarios and it allows you to import or update all elements of Product Catalog, any core Microsoft CRM entity (Accounts, Contacts, Leads, etc.) and also custom entities.

This manual contains the following main chapters:

- **Import Tool Overview**
- **Running the Import Tool for the first time**
- **Using the Import Tool**
- **Import Tool Setup**
- **Creating a New Project**
- **Fields Mapping**
- **Validating and running the import**
- **Expressions**
- **Functions**
- **Scripts**

Chapter 1: Import Tool Overview

Import Tool is a powerful application specifically designed to simplify the data import to Microsoft Dynamics CRM. It allows you to import or update all core entity types (Account, Contact, Lead, etc.), all elements of Product Catalog or any custom entities.

With the Import Tool you can:

- Import data from different sources:
  - SQL Server
  - Microsoft Excel
  - Text files (csv, tsv, etc)
  - Any ODBC data provider
  - Email mailbox using POP3 or IMAP protocol
  - Or you can develop your own data source plugin.
- Save import project containing mapping schema, use functions to transform the input data, create custom scripts and much more.
- Use different import modes: import new records, update existing records, perform both at the same time, delete unwanted records or change record status (for example activate, deactivate, etc.).
- Import data into custom entities and fields.
- Run import process from a command line which can be used to automate or schedule the import process.

Versions

Import Tool comes in two versions: **Data Migration** version and **Full** version.

Data Migration version is designed for the initial data imports and will work only for 60 days. This version should help customers to import essential data into Microsoft Dynamics CRM, test and make all necessary data updates at the beginning of the implementation process.

Full version is ideal for initial and on-going data imports as it has no expiration date. This is the best choice for customers that require regular updates to the data stored in Microsoft Dynamics CRM, for example when price lists need to be updated regularly, new products are frequently imported and so on.

In addition to that, Import Tool is divided into 7 modules which allow importing or updating different groups of entities. Customers that do not need to import all types of records can choose only a single module or multiple modules that are critical for their business. Furthermore import of custom entities is always included when at least one module is registered.

Below table lists the modules available and Dynamics CRM entities they will allow to import.

Module	Dynamics CRM Record	Dynamics CRM Entities
Customer Records	Account	account
	Address	customeraddress
	Competitor	competitor
	Contact	contact
	Currency	transactioncurrency
	Customer Relationship	customerrelationship
	Lead	lead
	Opportunity Relationship	customeropportunityrole
Activities	Appointment	appointment
	Campaign Activity	campaignactivity
	E-mail	email
	E-mail Template	template
	Fax	fax
	Letter	letter
	Phone Call	phonecall
	Recurring Appointment	recurringappointmentmaster
	Service Activity	serviceappointment
	Task	task
Product Catalog	Currency	transactioncurrency
	Discount	discount
	Discount List	discounttype
	Price List	pricelevel
	Price List Item	productpricelevel
	Product	product
	Product Substitute	productsubstitute
	Unit	uom
	Unit Group	uomschedule
Sales	Account	account
	Competitor	competitor
	Currency	transactioncurrency
	Document	document
	Goal	goal
	Goal Metric	metric
	Invoice	invoice
	Invoice Product	invoicedetail
	Lead	lead
	Marketing List	list
	Marketing List Member	listmember
	Opportunity	opportunity
	Opportunity Product	opportunityproduct
	Opportunity Relationship	customeropportunityrole
	Order	salesorder
	Order Product	salesorderdetail
	Queue	queue
	Quote	quote
	Quote Product	quotedetail
	Rollup Query	goalrollupquery
	Sales Literature	salesliterature
	Sales Attachment	salesliteratureitem

	Territory	territory
Service	Account Article Article Comment Article Template Case Case Resolution Contact Contract Contract Line Contract Template Currency Facility/Equipment Queue Resource Group Service Activity Site Subject	account kbarticle kbarticlecomment kbarticletemplate incident incidentresolution contact contract contractdetail contracttemplate transactioncurrency equipment queue constraintbasedgroup serviceappointment site subject
Marketing Campaigns	Account Campaign Campaign Activity Campaign Item Campaign Response Contact Currency Lead Marketing List Marketing List Member	account campaign campaignactivity campaignitem campaignresponse contact transactioncurrency lead list listmember
System	Business Unit Connection Connection Role Currency Note Queue Resource Group Site Subject Team Territory User	businessunit connection connectionrole transactioncurrency annotation queue constraintbasedgroup site subject team territory systemuser
Full version	All Above	All Above

Product Licensing

Demo Licenses

Import Tool is licensed per Organization. Downloaded version is fully functional limited to importing or updating 5 records only. This is the only restriction when no registration key has been entered. Evaluation license is not needed.

Compatibility

Import Tool 2015 is compatible with all deployment types of Microsoft Dynamics CRM 2015.

1. **Go to All Programs and find "*DPS Limited*" folder**

   If you are using 32-bit version of Windows the following shortcuts will be available:

   - Import Tool 2015
   - Import Tool 2015 Command Line

   If you are using 64-bit version of Windows there will be two more shortcuts available:

   - Import Tool 2015 (32-bit)
   - Import Tool 2015 Command Line (32-bit)

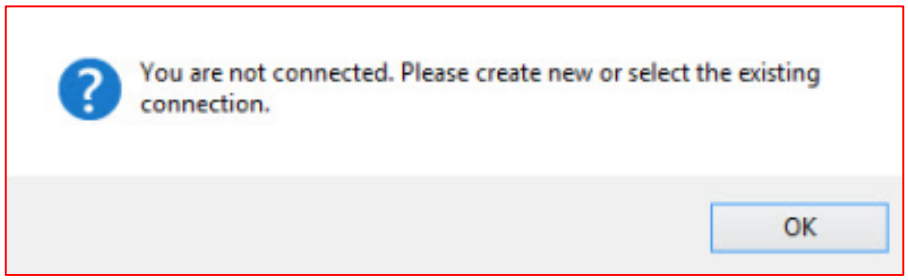
   Both 32- and 64-bit versions are fully compatible with Dynamics CRM 2015 and provide the same functionality. The only difference is that if you want to use for example Excel file as your data source and you are running 64-bit Windows, there is no 64-bit ODBC provider for Excel and you will have to use 32-bit version of the Import Tool. And this is the only reason why we are providing 32-bit version of the Import Tool.

2. **Select "*Import Tool 2015*"**

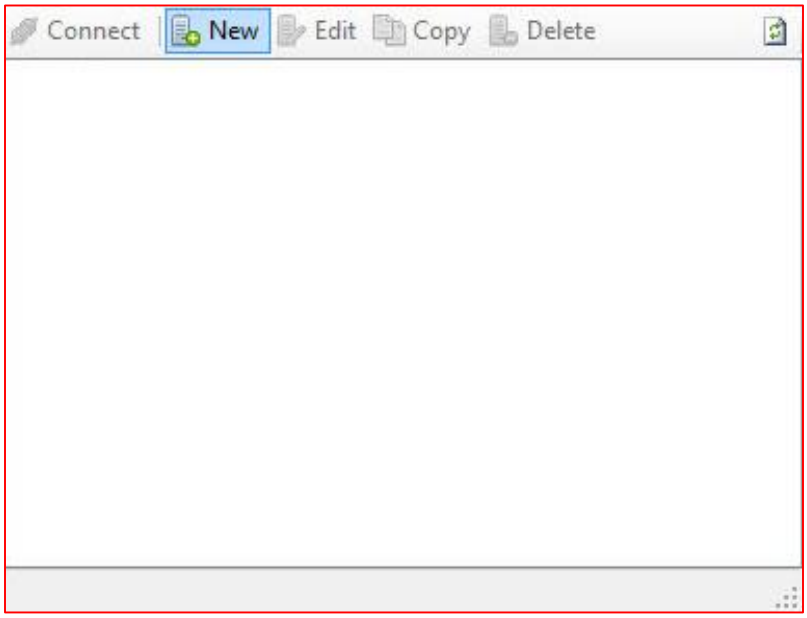
This will start the Import Tool application.

Connection Configuration

When running the Import Tool for the first time you will be asked to create a new connection.

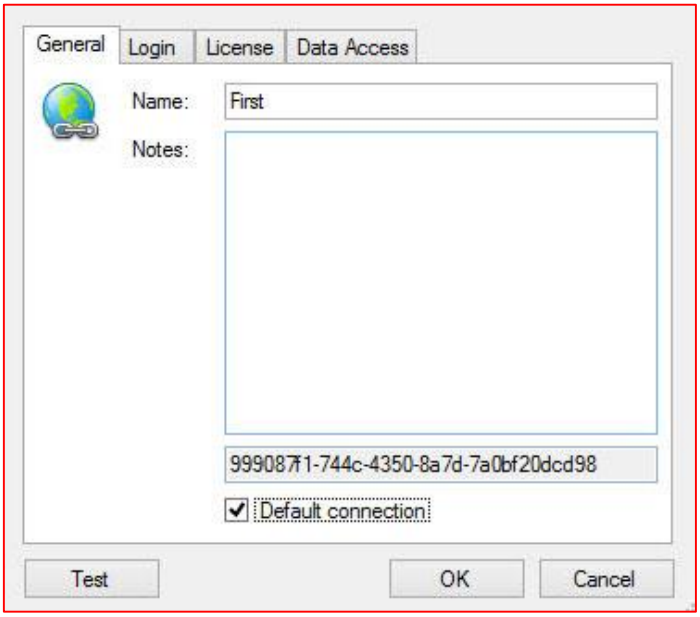


Click "OK" to continue.



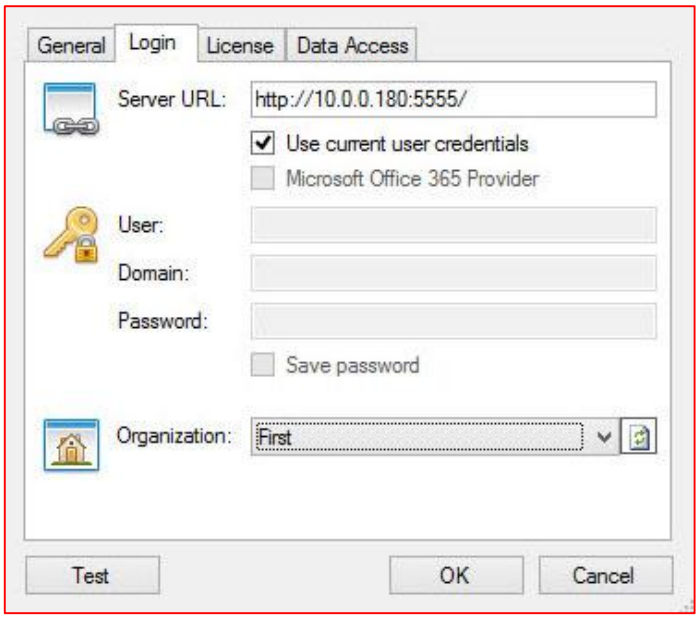
Click "New" button. "Connection Editor" window will open. It has four tabs:

- **General**



Specify the name for your connection. You can also add some notes (optional). If you tick "Default connection" then the connection you are creating will be selected when launching the Import Tool next time (assuming you have more than one connection available, otherwise if there is only one connection it is always treated as a default).

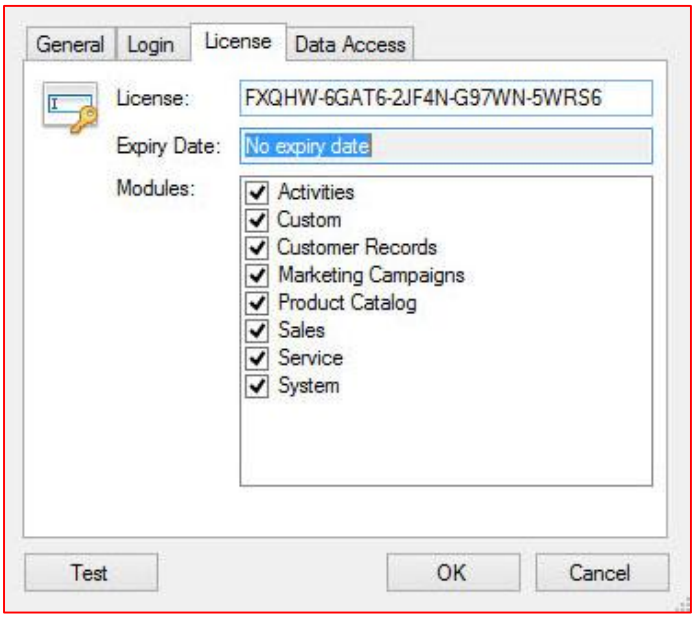
- **Login**



Enter the URL of your CRM server. If you are connecting to the local CRM installation using Active Directory authentication you can tick "Use current user credentials" if you want to use credentials of the currently logged on user. If you are connecting to other type of CRM deployment (Live, IFD, etc.) you have to provide user name and password and optionally domain name. If you don't want to be prompted for the password every time application starts, make sure that "Save password" is ticked. "Microsoft Office 365 Provider" should be ticked if you have your Live CRM 2011 provisioned through Microsoft Office 365.

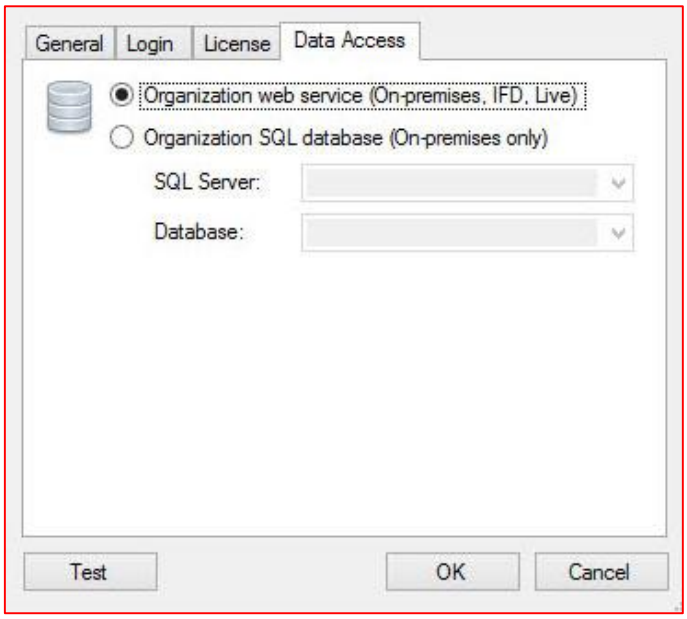
Click  to populate the organizations and select the organization you want to connect to. At this stage you can test your connection by clicking "Test" button.

- **License**



Enter your registration key if you already have it. You can always enter the registration key later using this window or from the Main Menu by going to Tools → Registration. Without registration the application is fully functional, but is limited to process (import) first five records only.

- **Data Access**

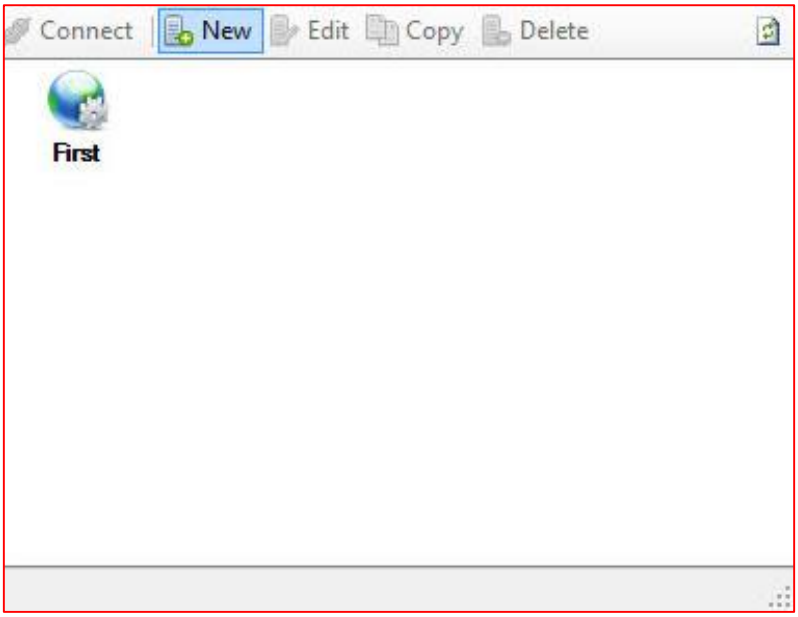


Data Access settings specify how the organization data is accessed. There are two modes:

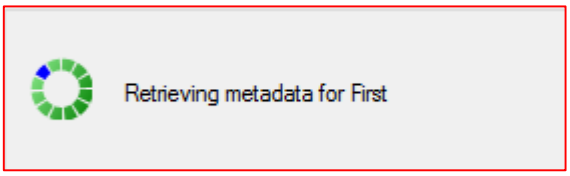
- *Organization web service* – this is recommended setting. When selected, CRM data will be accessed using the CRM web services only.
- *Organization database* – this option can be enabled to improve performance when accessing the CRM data (database is used for read operations only). However this can only be used in on-premises installation types as you will have to specify SQL Server and Database that should be used (make sure that it matches the selected organization).

Connection Manager can be access from the main menu: Connections → Manage.

Once your connection has been defined, click "Connect".

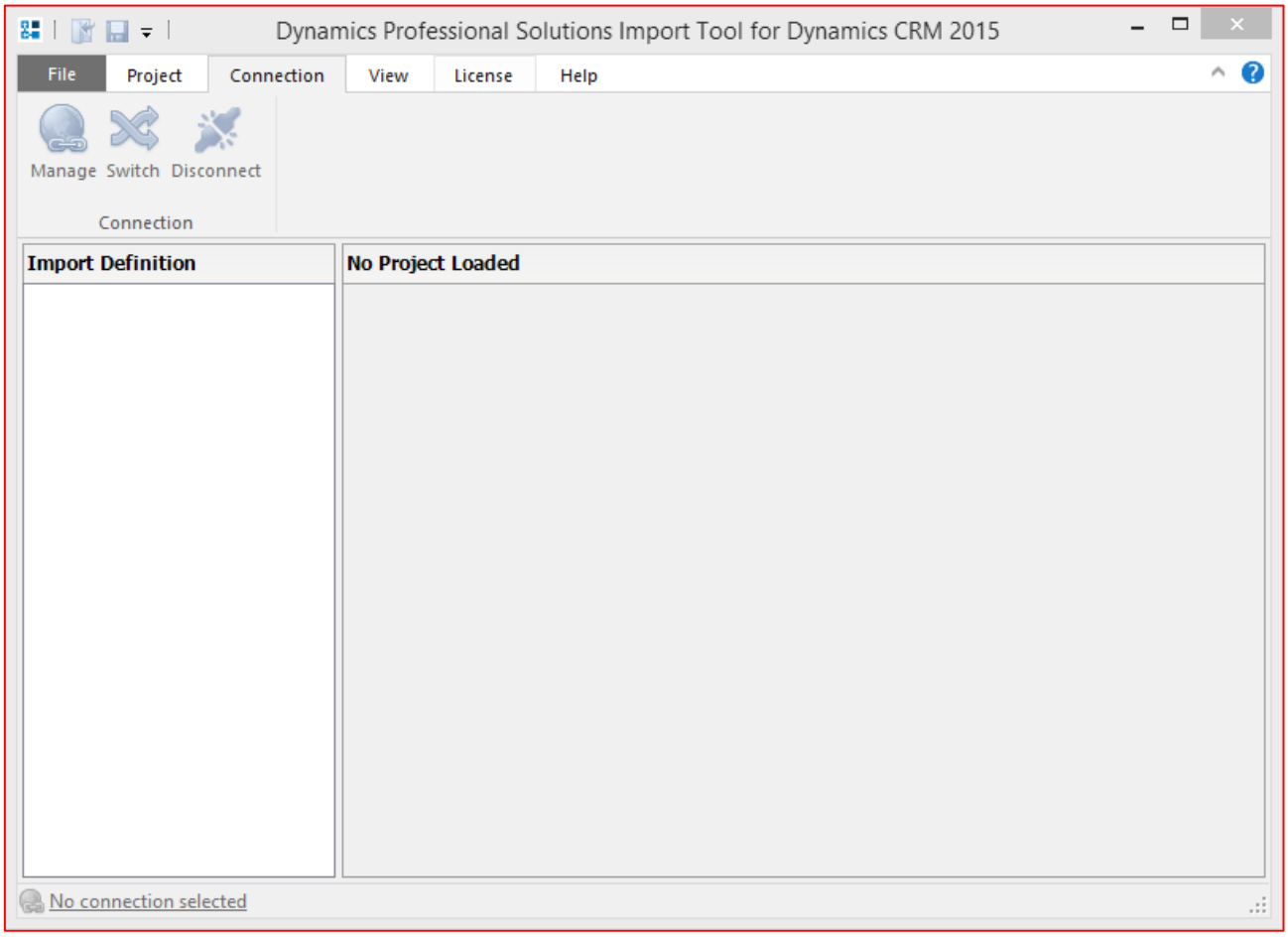


3. Progress window will be displayed:



4. It can take couple minutes for the first time to complete.

To access the Import Tool, go to Start → All Programs → DPS Limited → Import Tool 2011. This is the main application window.

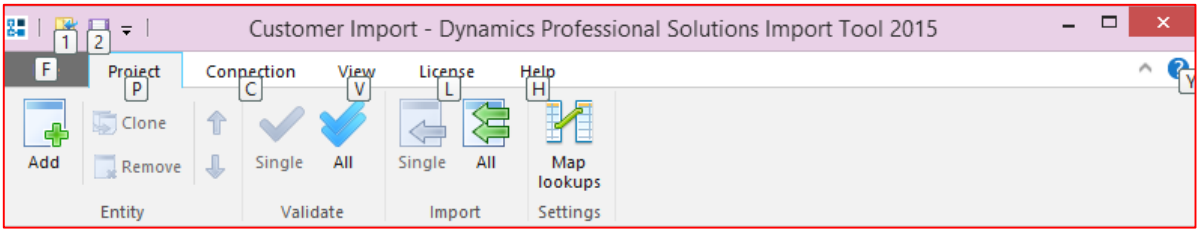


Ribbon

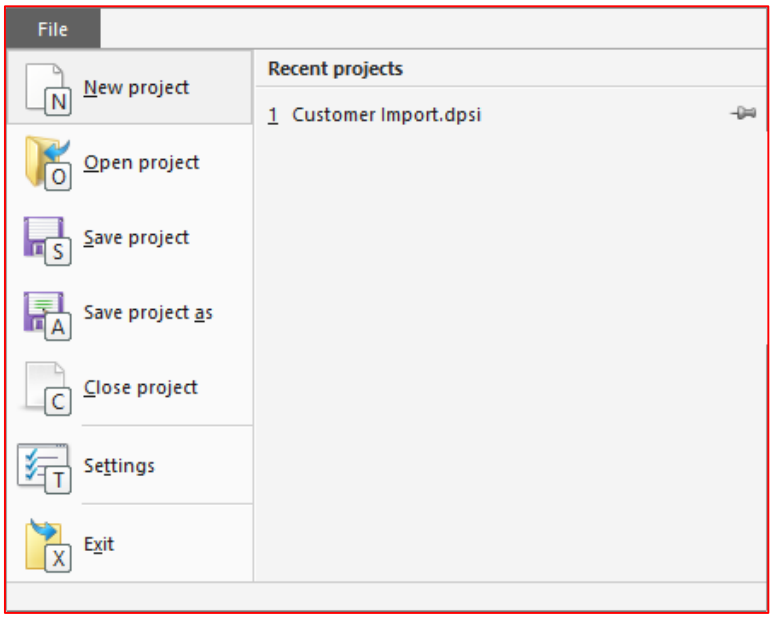
In this version of the Import Tool, menu and toolbar has been replaced by Ribbon. This is similar interface to Microsoft Office and CRM 2011.

Note that if you are still using older version of Windows, you will be presented with standard menu and toolbar instead. Minimum operating system version for Ribbon support is Windows Vista SP2 with Platform Update. See ***Appendix A*** *for the interface reference.*

There are 6 tabs available:

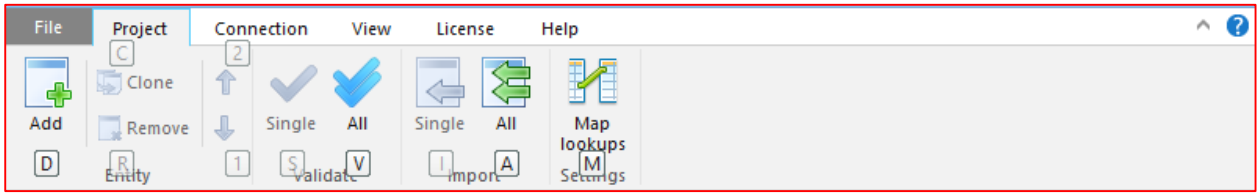


1. **File tab**



Command	Description
New project	Creates a new empty import project.
Open project	Opens the existing import project.
Save project	Saves the import project.
Save project as	Saves the import project under a different name.
Close project	Closes the import project.
Settings	Opens the Settings window.
Exit	Closes the Import Tool application.
Recent projects	Contains recently opened import projects.

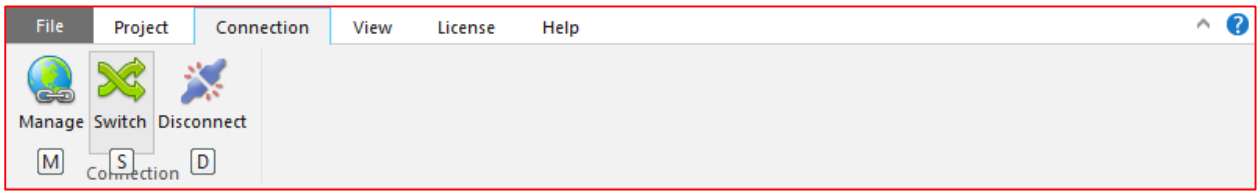
2. Project tab



Group	Command	Description
Entity	Add	Adds a new entity import definition to the project.
Entity	Clone	Creates a copy of the currently selected entity import definition.
Entity	Remove	Removes the currently selected entity from the import project.
Entity	Move up	Moves the currently selected entity one level up in the hierarchy.
Entity	Move down	Moves the currently selected entity one level up in the hierarchy.
Validate	Single	Validates the import for currently selected entity.
Validate	All	Validates the import for all enabled entities within the whole project.
Import	Single	Runs the import for currently selected single entity only.

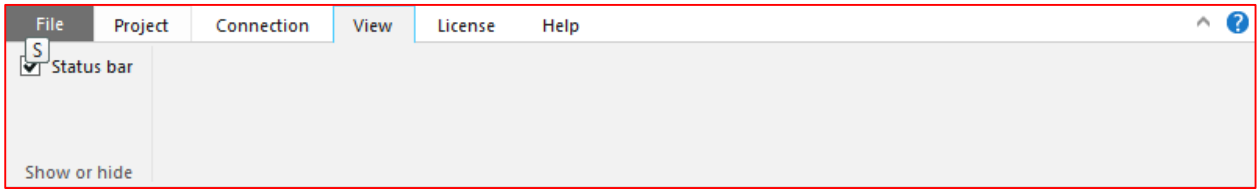
Import	All	Runs the import for all enabled entities within the whole project.
Settings	Map lookups	Allows changing default fields used for finding related records.

3. Connection tab



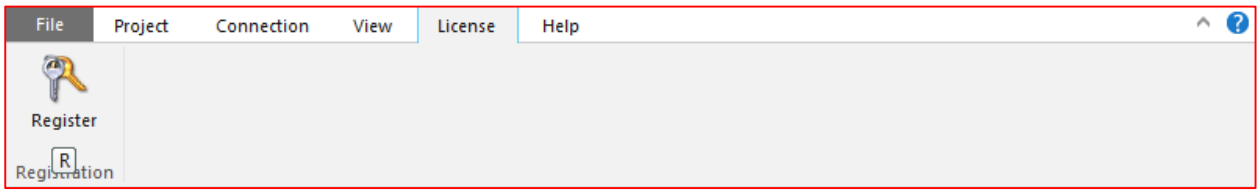
Group	Command	Description
Connection	Manage	Opens the Connection Manager window.
Connection	Switch	Opens the Switch Connection dialog. Command is available if more than one connection has been defined.
Connection	Disconnect	Closes current connection.

4. View tab



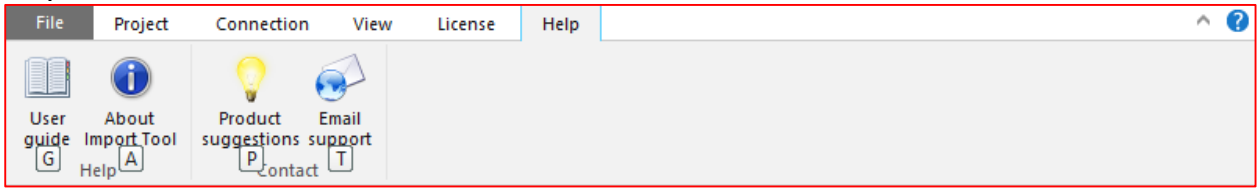
Group	Command	Description
Show or hide	Status bar	Shows or hides the status bar.

5. License tab



Group	Command	Description
Registration	Register	Opens the registration window.

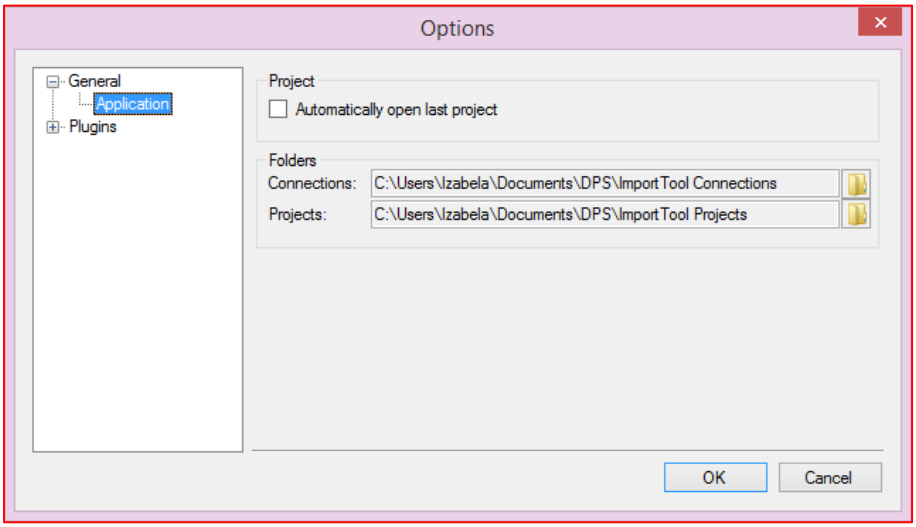
6. Help tab



If you are using older version of Windows, instead of Ribbon you will be presented with standard menu and toolbar. Minimum operating system version for Ribbon support is Windows Vista SP2 with Platform Update.

1. Settings (Ribbon → File tab)

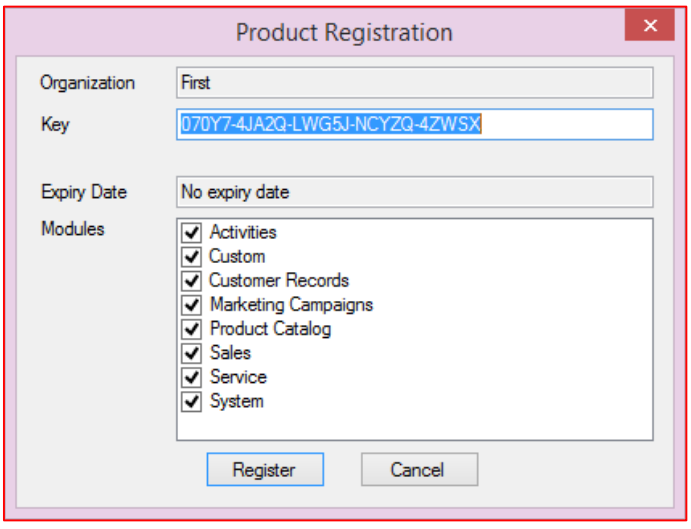
Use this screen to specify general application settings like whether to automatically open your last project; default locations for the connection definitions and project files. *Note if you change the default locations, your existing connections and /or projects files will not be automatically copied to the new location.* You can also specify your own custom data source plugins.



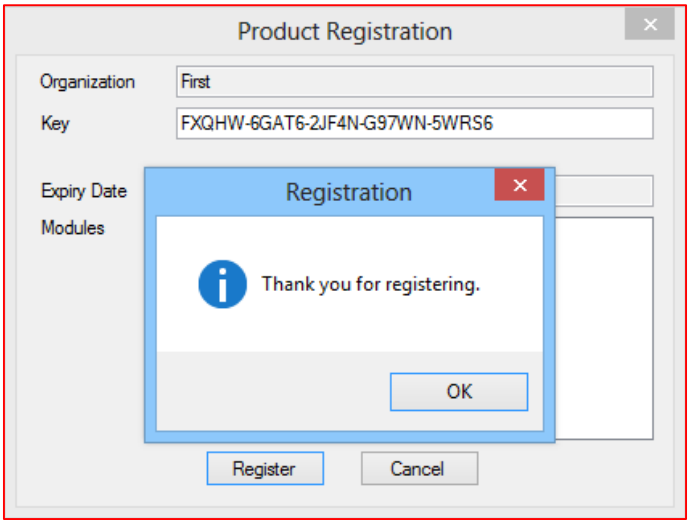
2. Product registration (Ribbon → License → Register)

The downloaded version of the Import Tool contains a fully functional product with a limit of five records that can be processed (created, updated, etc.). This restriction is waived upon product registration.

After purchasing the full product version you will receive an e-mail with the registration key for your organization. Enter the registration key you have received. System will show eventual expiry date and modules that you have purchased. Please verify if this information is correct. You can also enter the registration key in the connection setup window.

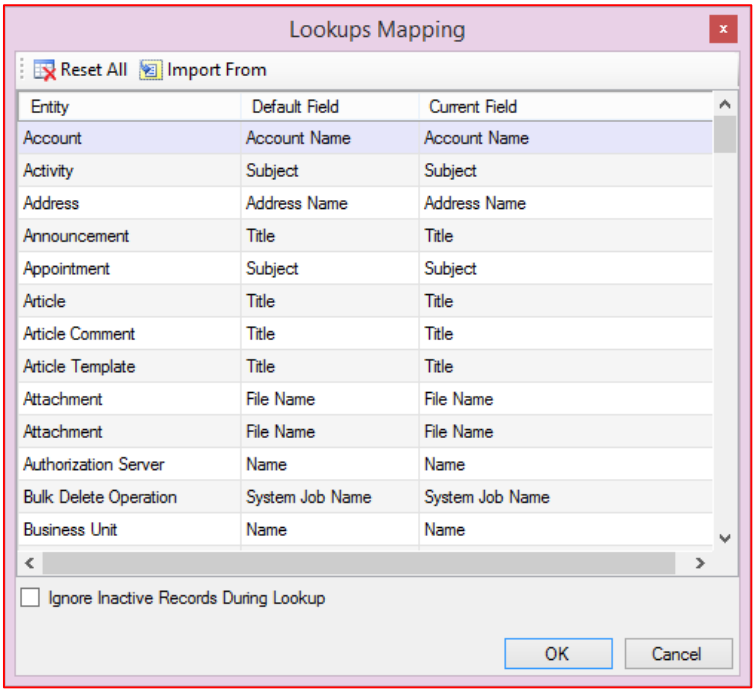


Enter the key and click "Register".



3. Lookups Mapping

Lookup fields are used to relate one record type to another. For example when creating a new task for a customer contract we are using a related field to associate task with the necessary contract record. Import Tool uses predefined default field (provided by Microsoft Dynamics CRM) to recognize what record you are referring to. In our example this will be the Contact Name that Import Tool will use by default; if in your import data you have a Contact ID instead, you can change the default lookup setup in the Lookup Mapping window.

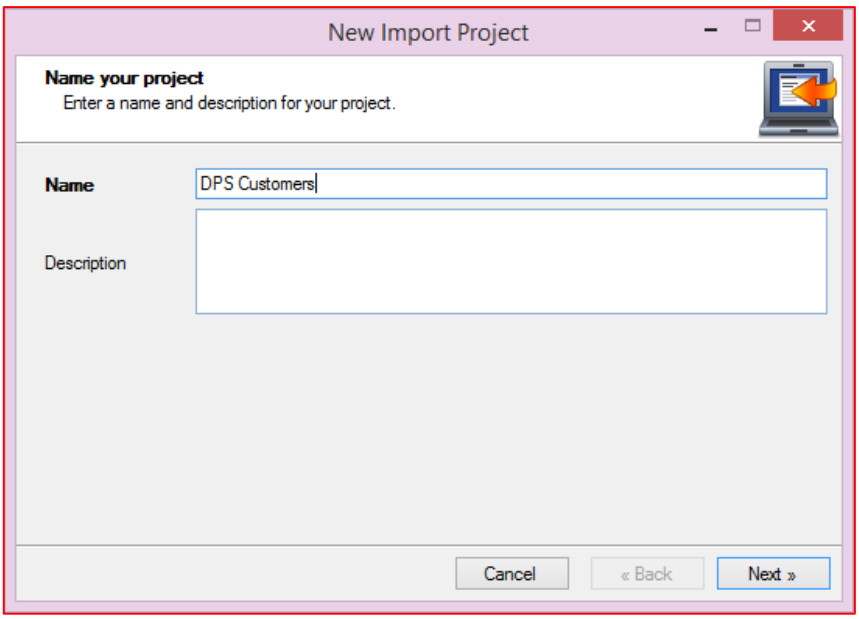


"Reset All" button resets all lookups to use default field. As this information is stored in the project file, you can import mapping from another project file by selecting "Import From".

Follow below steps to create new import project:

1. Select New Project from the File tab (Ctrl+N).

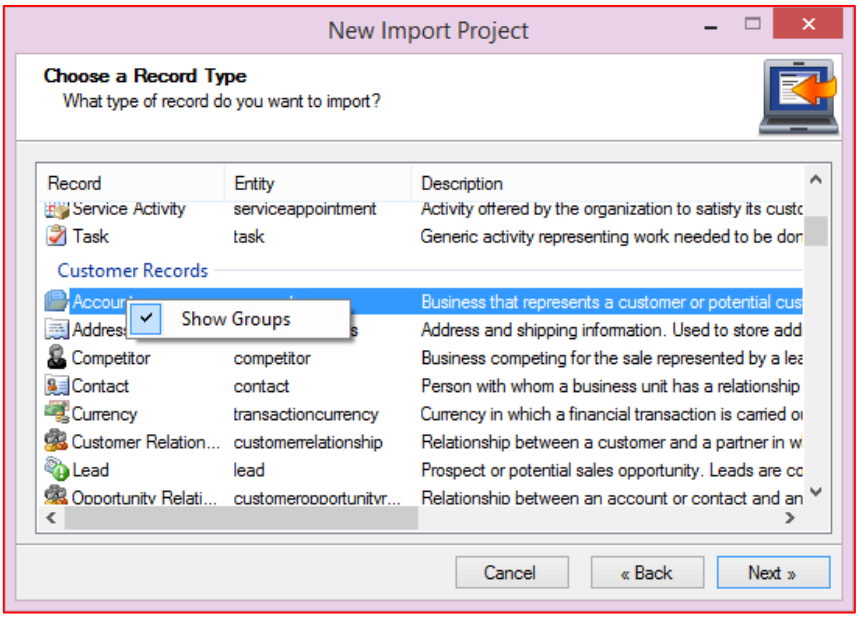
In the "Name your project" screen enter the name and optional description of your project. Click "Next".



4.

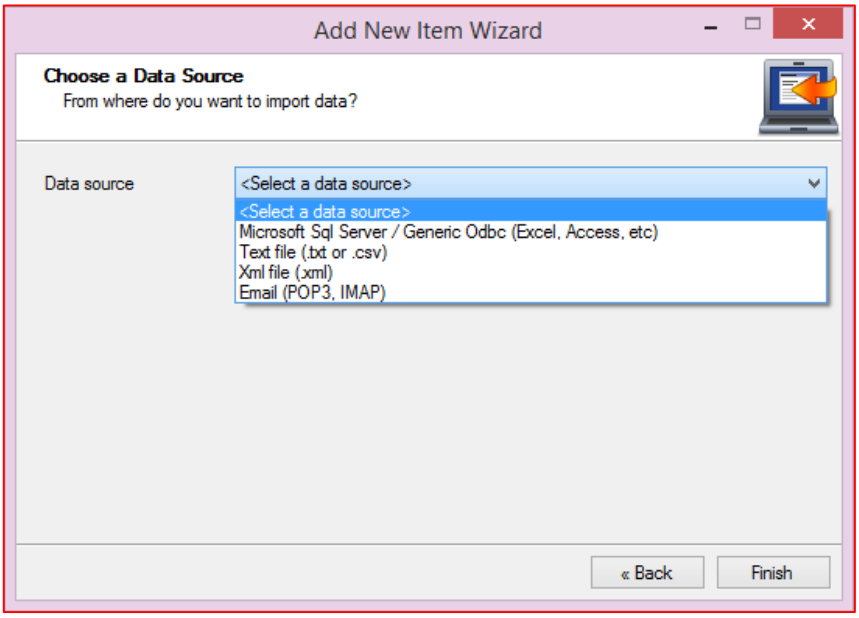
2. Choose a Record Type

In the "Choose a Record Type" screen select the entity you are planning to import data to. All Dynamics CRM entities that are available for import are displayed regardless of the registration status. By default available entities are shown in groups by module; if you want to change that, right click on the list and unmark "Show Groups" option. Click "Next".



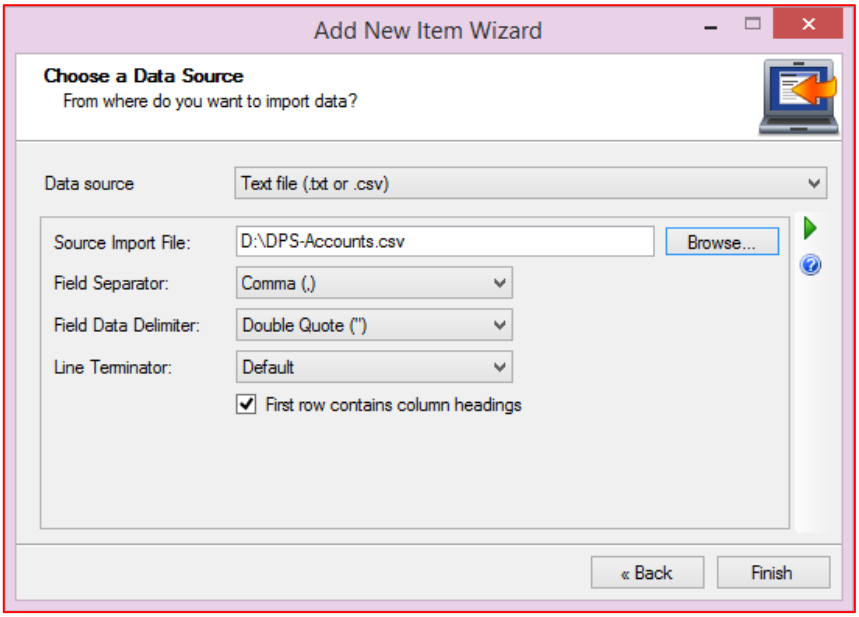
3. Data Source

In the "Choose a Data Source" screen select a source of the data you would like to import from. Then specify information needed by the selected data source type. Click "Finish"



To import from a text file select the "Text file (.txt or .csv)" option form the Data source dropdown. Use Browse button to locate your source file. This technically can be any text file:

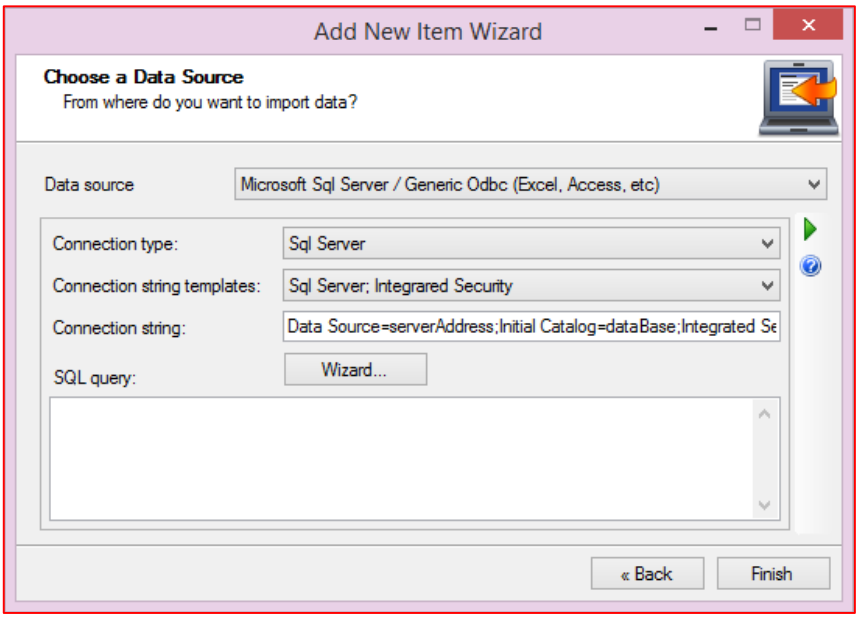
- Fields should be separated by a comma, tab, semicolon or colon.
- If any field value contains field separator, it must be delimited by single or double quote. Otherwise delimiter is not needed. For example if you have selected comma as a separator, any field value containing comma must be enclosed by "" or '' ("Smith, John" or 'Smith, John').
- Line terminator is either CRLF , CR, LF or default. In most cases there is no need of changing it unless you want to import multi line fields.
- First row may contain a header. This is highly recommended as it later greatly helps to map fields.



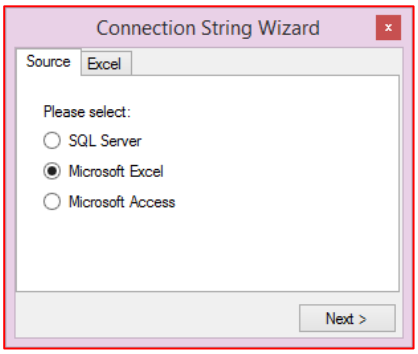
Clicking on  will test the connection.

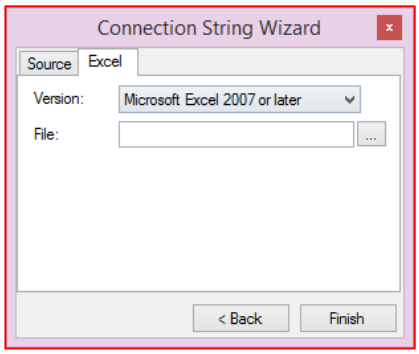
To import from SQL Server or any ODBC data sources select Microsoft SQL Server/Generic ODBC option form the Data Source dropdown.

Select your connection type and enter a connection string. You can use one of the templates if you are not sure about connection string syntax.
SQL Server connection type is optimized for Microsoft SQL Server. **Generic ODBC** can be used for any other data source, like Microsoft Access, Microsoft Excel, My SQL, etc.



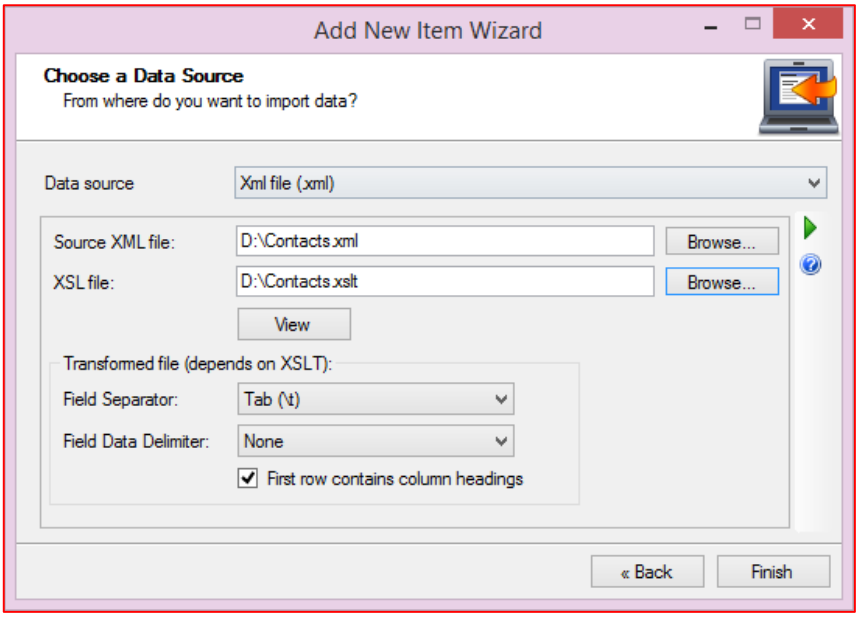
Wizard button can help you to define a connection string. First select your data source (SQL, Excel, Access), click "Next" button and provide all required information. This Wizard is optional and is designed to help you with SQL Server, Microsoft Excel and Microsoft Access connection string creation. You can always enter a connection string manually.





Then enter your query in the SQL query field. This can be a simple or complex statement depending on your needs. Clicking on  will test the connection.

To import data from xml file select XML file (.xml) option from the Data source dropdown. Because xml file structure cannot be automatically determined it is required to provide an additional transformation file that tells how to extract the information from the source XML file. Clicking on  will test the connection.



Below is an example of XML file containing contact data and transformation that is used to convert the source XML into a tab separated file. You can test your transformation by clicking "View" button.

XML file:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Sample xml file to be used with Contacts.xslt -->
<Contacts>
  <Contact     City="Sacramento"     State="CA"     Country="US"     PostalCode="94987"
EmailAddress="MaryBergstrom@ads.com">
    <Address>
      1178
      Sandy Blvd.
    </Address>
    <FirstName>Mary</FirstName>
    <LastName>Bergstrom</LastName>
  </Contact>
  <Contact     City="Springfield"     State="IL"     Country="US"     PostalCode="62333"
EmailAddress="RobertBorges@adc.com">
    <Address>1074 Lori Drive &gt;X&lt; </Address>
    <FirstName>Robert</FirstName>
    <LastName>Borges</LastName>
  </Contact>
  <Contact     City="Hartford"     State="CT"     Country="US"     PostalCode="06198"
EmailAddress="StephenAcevedo@acm.com">
    <Address>
      1172
      Flamingo
      Dr.
    </Address>
    <FirstName>Stephen</FirstName>
    <LastName>Acevedo</LastName>
  </Contact>
  <Contact     City="Dallas"     State="TX"     Country="US"     PostalCode="20313"
EmailAddress="AdrianDumitrascu@ac.com">
    <Address>100 Red Oak Lane</Address>
    <FirstName>Adrian</FirstName>
    <LastName>Dumitrascu</LastName>
  </Contact>
</Contacts>
```

Transformation (XSL) file:

```
<?xml version="1.0" encoding="utf-8"?>

<!--
```

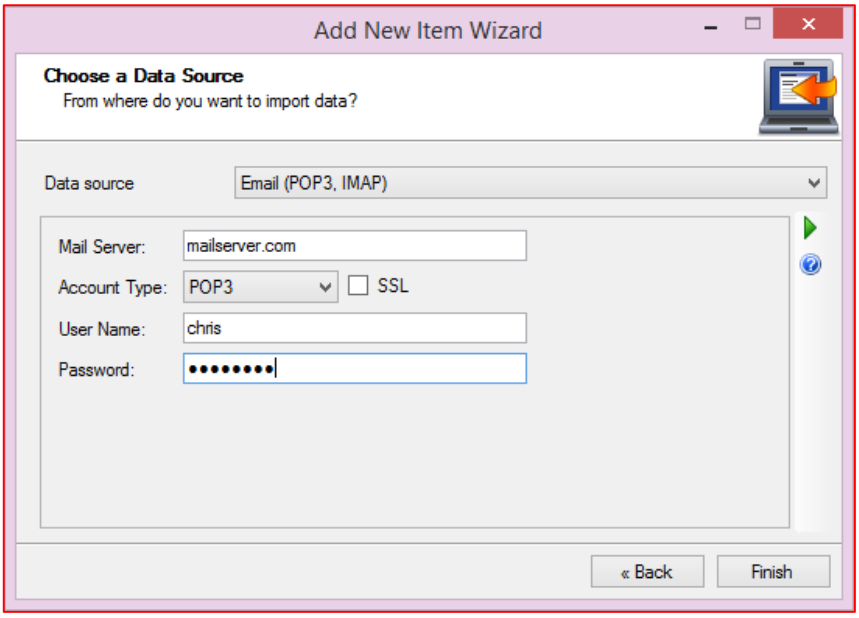
```
    This is an example of simple xsl transformation file. It matches Contacts.xml
structure.
-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <!-- Set output format: Unicode, no xml header, text only. -->
  <xsl:output encoding="utf-8" omit-xml-declaration="yes" method="text" />

  <xsl:template match="/">
    <!--
        Output document header. Number of items should match fields.
        Name can be the same as xml element or attribute, but doesn't have to.
        Fields are separated by tab (&#9;); new rows by CRLF (&#xD;&#xA;).
    -->
    <xsl:text>Address1_Line1&#9;</xsl:text>
    <xsl:text>Address1_City&#9;</xsl:text>
    <xsl:text>Address1_StateOrProvince&#9;</xsl:text>
    <xsl:text>Address1_Country&#9;</xsl:text>
    <xsl:text>Address1_PostalCode&#9;</xsl:text>
    <xsl:text>EmailAddress&#9;</xsl:text>
    <xsl:text>FirstName&#9;</xsl:text>
    <xsl:text>LastName&#xD;&#xA;</xsl:text>

    <!--
         Output fields. Number of items should match header.
         Fields are separated by tab (&#9;); new rows by CRLF (&#xD;&#xA;).
    -->
    <xsl:for-each select="/Contacts/Contact">
      <xsl:value-of select="normalize-space(Address)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(@City)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(@State)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(@Country)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(@PostalCode)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(@EmailAddress)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(FirstName)" />
      <xsl:text>&#9;</xsl:text>
      <xsl:value-of select="normalize-space(LastName)" />
      <xsl:text>&#xD;&#xA;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

To import data from an email account select Email (POP3, IMAP) option from the Data source dropdown.

Specify your email server address and type (POP3 or IMAP). You can also specify a port in the mail server address if non-standard port should be used (for example: pop3:myserver.com:3200). Mark SSL if your server requires secure connections. You also have to provide user name and password. Clicking on  will test the connection.

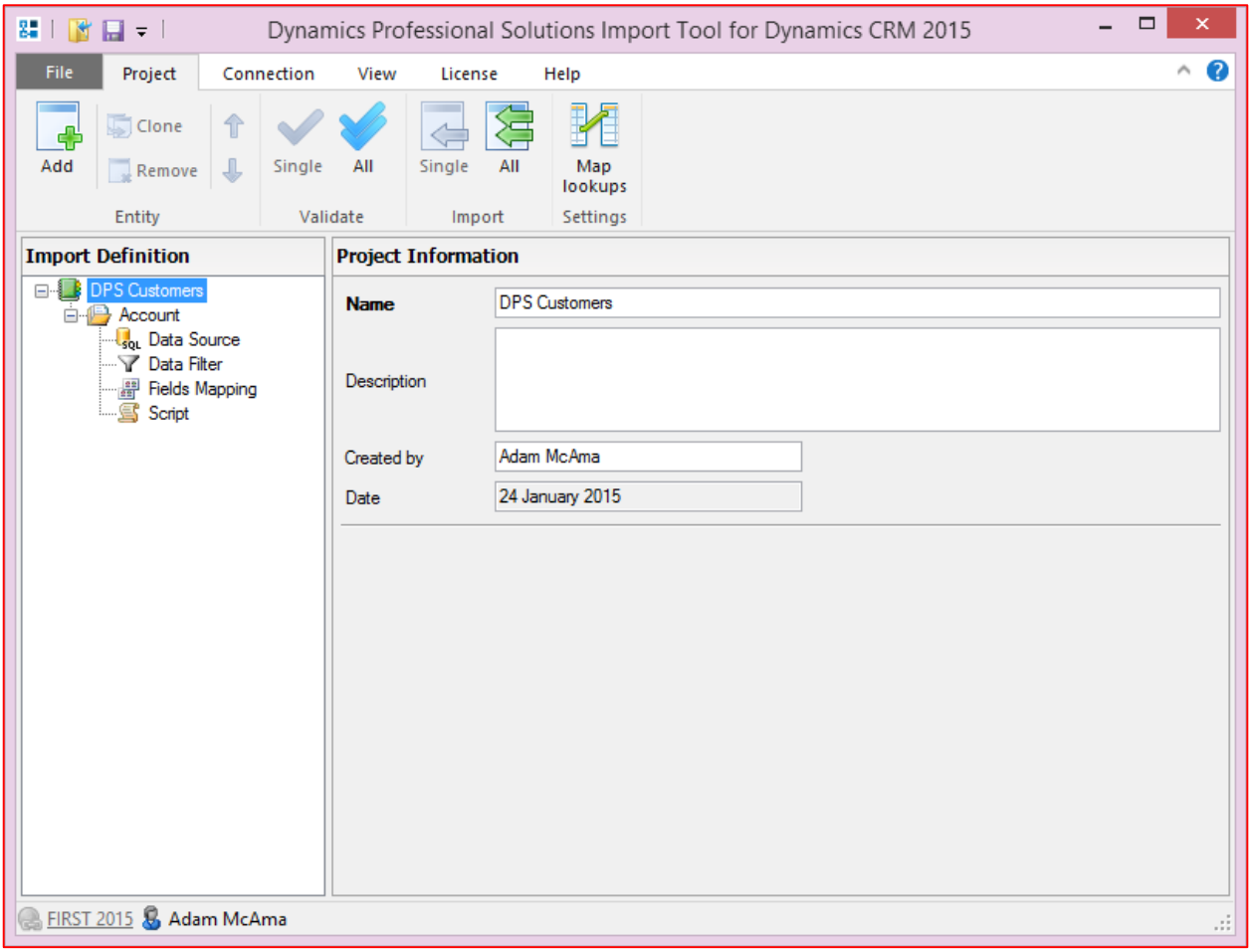


When you have finished defining the data source, click "Finish" button and you will see the main screen window.

The main screen is dived into two sections. Left section contains a tree structure and is called Import Definition. It shows the entities included in the project.

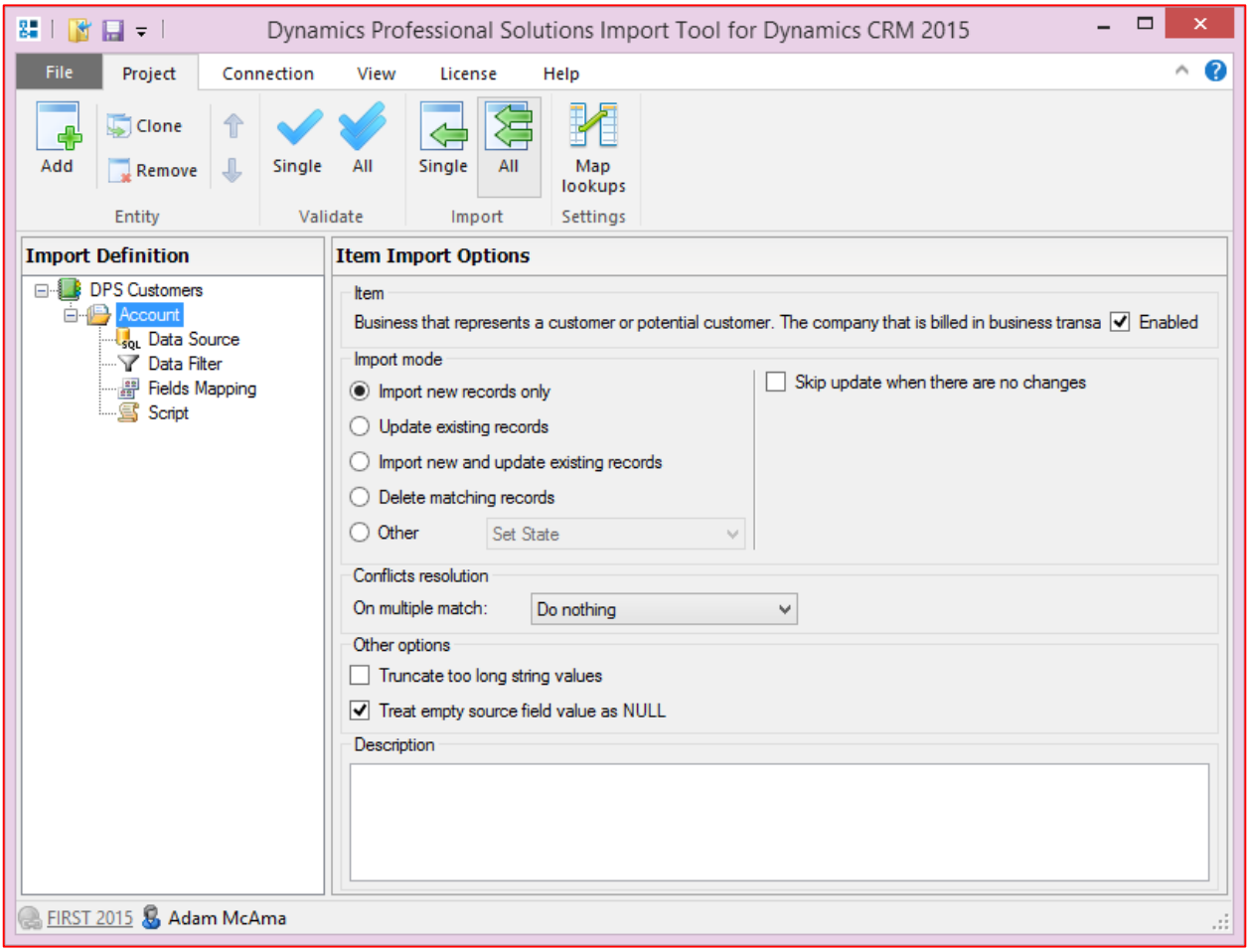
Right section is content sensitive and shows detailed information for what is selected in the left section. Section title will change depending on what level is selected in the left pane. You can select a top-level Project item, Entity, Data Source, Data Filter, Field Mapping or Script.

By selecting a project in the left pane **Project Information** will be displayed in the right section of the main screen.



Name	Name of your project.
Description	Optional description.
Created by	System will populate this field with a user name that created the project. You can change this information if you wish.
Date	System will display date when you project was created.

By selecting the entity in the left pane **Item Import Options** will be displayed in the right section of the main screen.



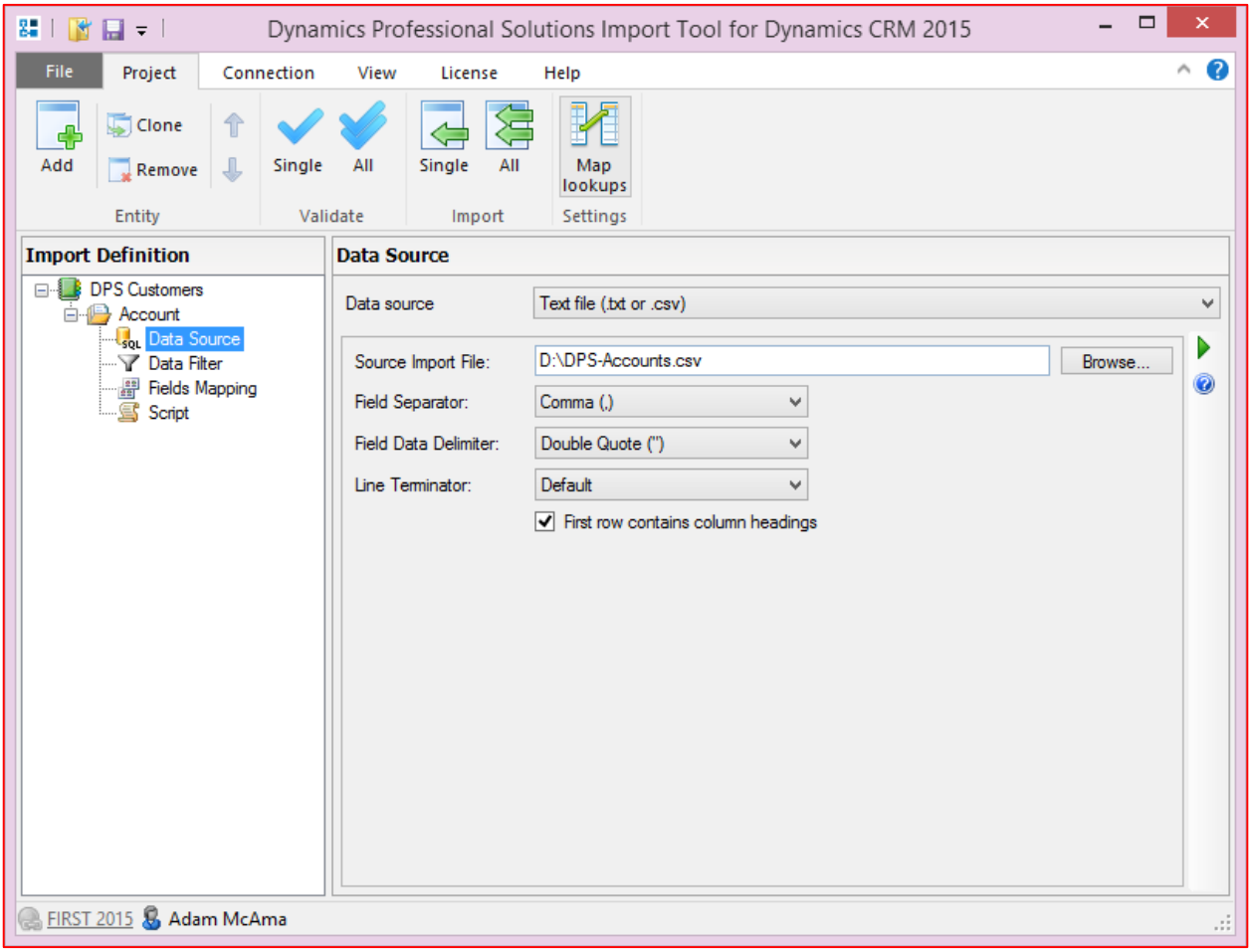
Item	Contains entity description and technical name.
Enabled	Specifies whether this entity should be included in the import. Disabled items will be skipped during the import.

Import mode	Specifies one of the import modes.
	If you want to update records make sure you have selected at least one key in a Fields Mapping Section.
	You may also delete records from the system or change record status.
	Available options:
	• **Import new records only** – only new records will be imported, if record already exists it won't be updated.
	• **Update existing records** – only updates existing records, if record doesn't exist it won't be created.
	• **Import new and update existing records** – combination of the two above options. New record will be created if it doesn't exist, otherwise existing record will be updated.
	• **Delete matching records** – matching records will be deleted.
	• **Other**
	  ○ **Activate** – use this option to activate inactive records.
	  ○ **Deactivate** – use this option to deactivate records.
	  ○ **SetState** – use this option to change record state.

Conflict resolution	Decide what to do when multiple records match the selected key. • **Do nothing** – no action will be executed if multiple existing records are found. • **Update all matching records** – action will be executed against all matching records.

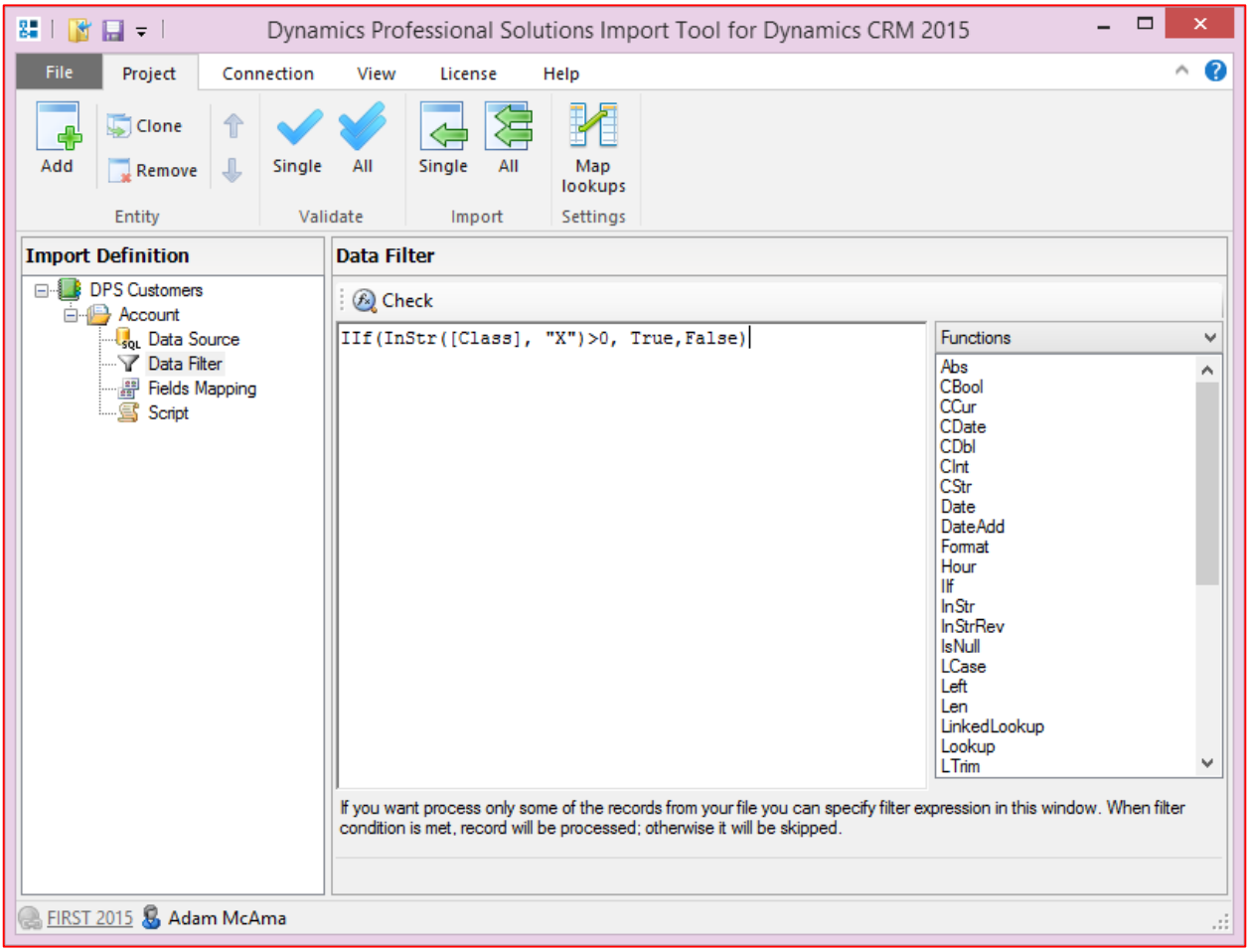
Truncate too long string value	If this option is marked system will truncate too long strings to the maximum allowed length of the specific field. Otherwise error will occur.
Treat empty source value as null	If this option is marked system treats empty source value as a NULL; otherwise as an empty string.
Skip update when there are no changes.	If this option is marked update command won't be executed if there are no changes. This means last modified column won't be updated. This option might be useful if you for example don't want to execute some workflow on save and there are no changes.
Description	Enter an optional description for the entity import.

By selecting Data Source for the specific entity in the left pane you can access **Data Source** settings in the right section of the main screen.



In this screen you can modify or change data source configuration.

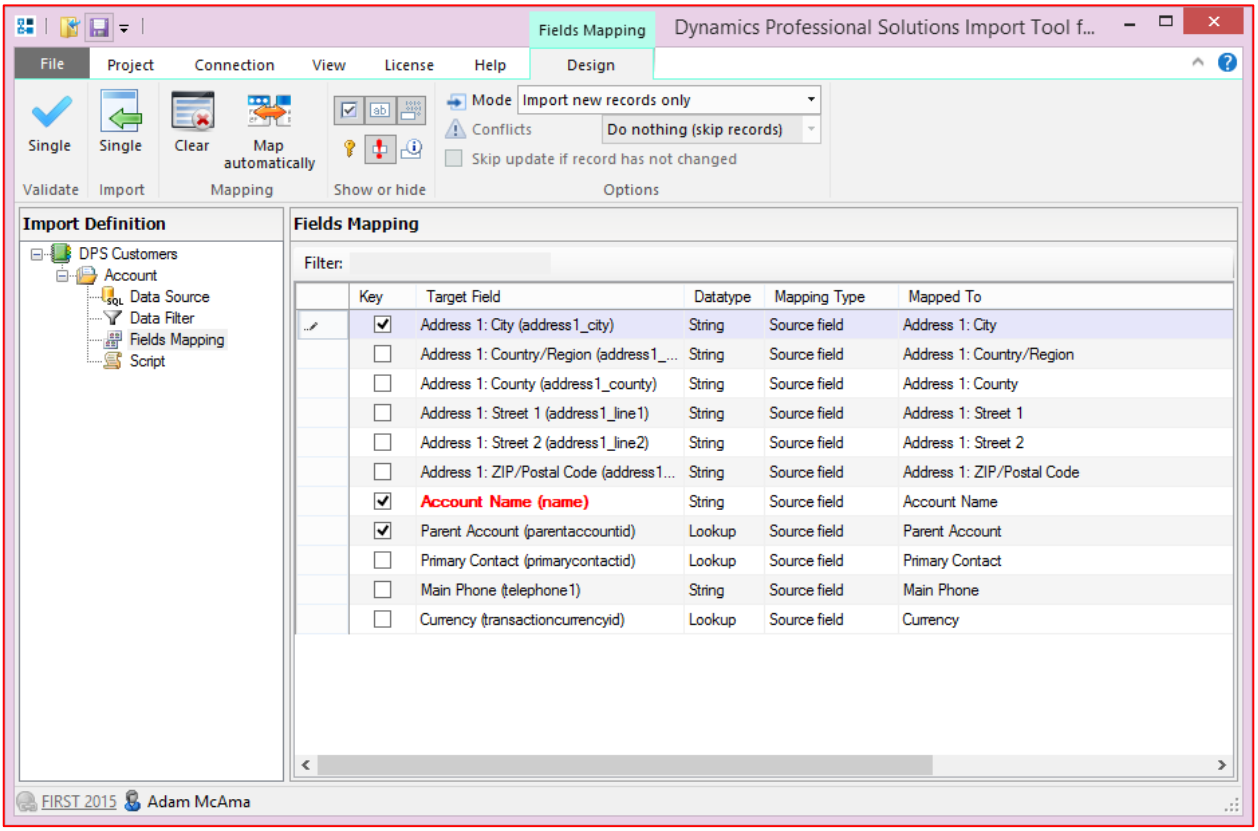
By selecting Data Filter for the specific entity in the left pane you can access **Data Filter** in the right section of the main screen.



Data filter may be useful if you want to process only some of the records that are in the source file and you are unable to filter the data using other means (for example using a data source query). When filter condition is met, record will be processed otherwise it will be skipped.

In the example above, if source field “Class” contains character “X” record will be imported and other records will be skipped. You can use Functions and Operators when defining a filter. More about functions and operators you can find in later chapter.

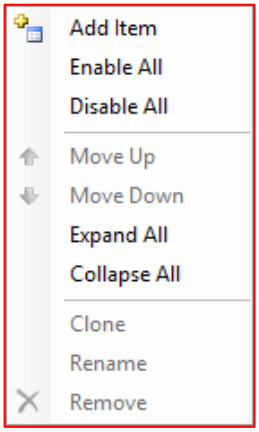
By selecting the Fields Mapping for the specific entity in the left pane you can access **Fields Mapping** in the right section of the main screen.



In this section you can map source fields to the CRM fields. This part will be described in details in the Chapter 6 of this manual.

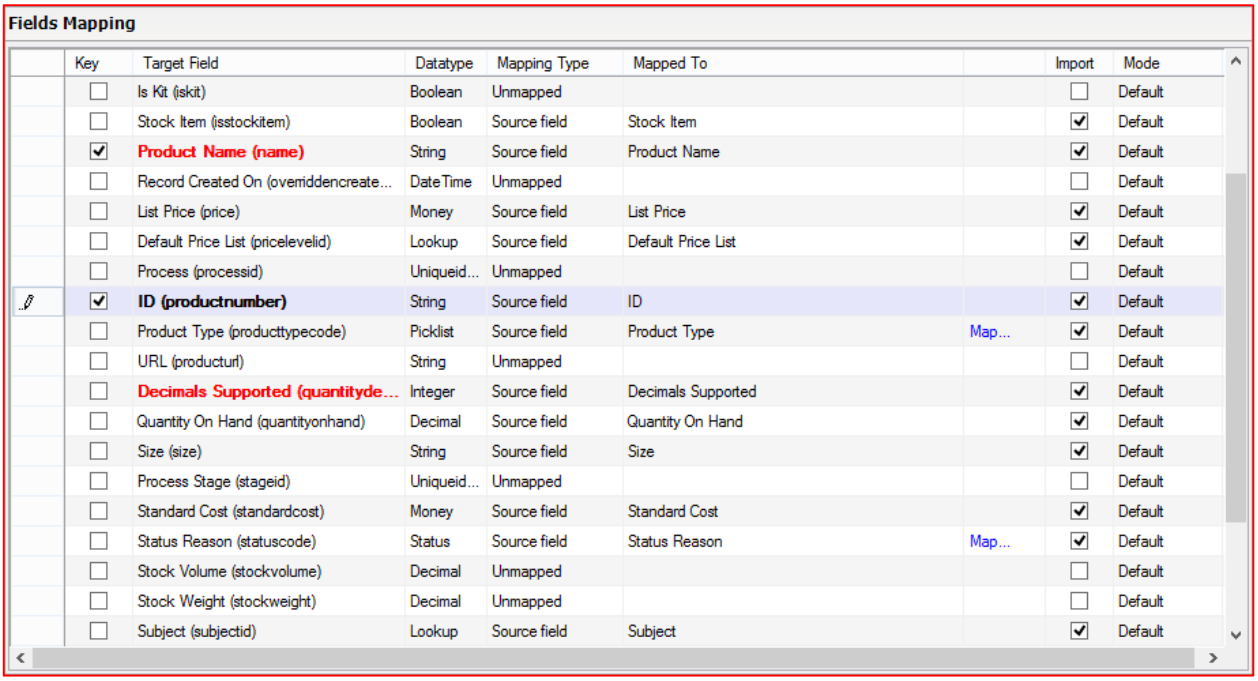
Left pane context menu

In the left pane you can right click in order to open a context menu.



Add Item	Adds a new item to the current project (new entity to import).
Enable All	Enables all items in the current project. Only enabled items will be imported.
Disable All	Disables all items in the project. Disabled items will not be imported.
Move Up	Moves the selected item up in the hierarchy. Items are imported in the order as displayed in the tree.
Move Down	Moves the selected item down in the hierarchy. Items are imported in the specified order.
Expand All	Expands all items in the project.
Collapse All	Collapses all items in the project.
Clone	Creates a copy of the currently selected item.
Rename	Renames the selected item.
Remove	Removes the selected item form the project.

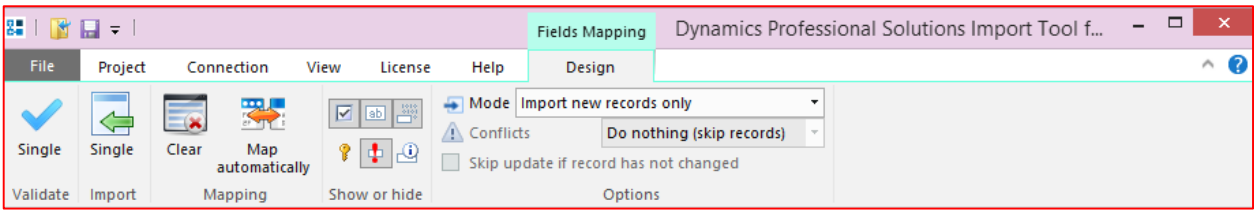
From the left pane in the main screen select Fields Mapping for the item you want to import.



Key	Indicates that this is the field that should be used to find existing record in the CRM. More than one field can be marked as key field. This basically determines which field or fields determine that record is unique. For example if you are importing contacts it can be email address (single key) assuming there can be only one contact with a given email address. Or it can be first and last name (two key fields). You don't have to select any key fields, but in this case duplicate records can be created. This can be resolved later by using duplicate detection in the CRM.
Target Field	This column lists all available fields for the selected entity type (e.g. fields in the CRM). Custom fields will be listed, too.
Data Type	This column provides information about data type of the target (e.g. CRM) field. Most of the types are self-explanatory. There is one special type that is worth mentioning: **Lookup**. This is a reference to another record in the CRM. For example Unit has Unit Group lookup field which specifies the parent unit group. Similarly parent customer for a contact, etc.
Mapping Type	This column tells the system where to get the imported value from. Available options are: • Unmapped – target field is not mapped to anything and won't be imported. • Source field – target field is mapped to the data source field. • Expression – target field is mapped to an expression (for example calculated value, constant). More about functions you can find in Chapter 9. • For some type of fields like Status, Picklist, Owner there will be another option specific to this type.
Mapped To	If the selected mapping type is "Source field" this column will list all

	fields available in the data source. If the selected mapping type is "Expression" this column will show inline expression editor.
	This unnamed column is used only when selected mapping type is "Expression" or if a target field is of specific type (for example Status, Picklist, Owner, etc.). In this case you will be provided with a list of available values.
Sample Data	This column displays sample data from the data source.
Import	This column indicates whether the field should be imported or not.
Mode	For a field to be imported specifies import mode: • Default – import always updates CRM field with value from a source (or expression). • Skip Empty – import updates CRM field only when source is not empty. • Append – import appends source value instead of overwriting existing value. Available for string and memo fields only. It is useful for appending new lines to memo fields.

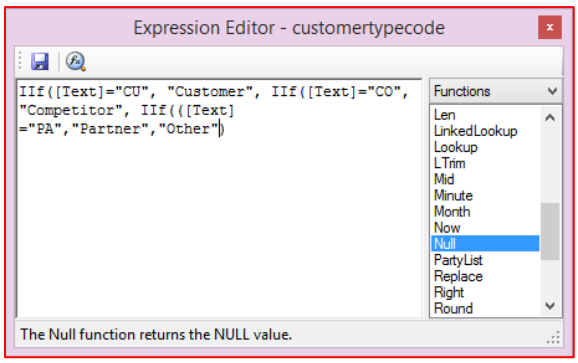
Ribbon commands and options – this is ribbon is visible when Fields Mapping is selected in the tree view:



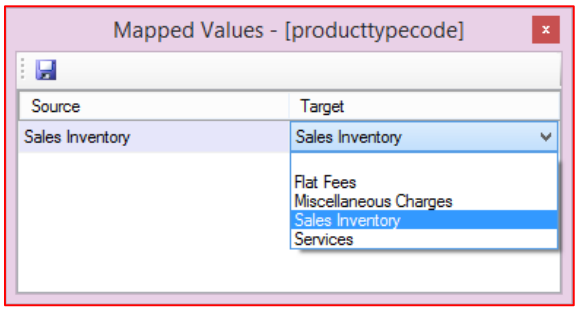
Group	Command	Description
Validate	Single	Validates the import for currently selected entity.
Import	Single	Runs the import for currently selected entity.
Mapping	Clear	Clears all mappings.
Mapping	Ma automatically	Tries to map source and target fields automatically based on matching target and source field names.
Show or hide		Show mapped fields only – hides or shows rows with unmapped fields.
Show or hide		Show sample data column – hides or shows the sample data column.
Show or hide		Show data type column – hides or shows the data type column.
Show or hide		Show primary key field – hides or shows a row with field that represent the ID of a record (GUID). This field cannot be imported to or updated, but can be used to uniquely identify the existing record.
Show or hide		Highlight required fields – when selected, business required fields will be highlighted. These are the fields required when **creating** new record.
Show or hide		Show target field information – hides or shows the panel with additional information about CRM field.
Options	Mode	See page 27 for more details about import modes.
Options	Conflicts	See page 27 for more details about conflict resolution.
Options	Skip update if records has not changed	See page 28 for more details about this option.

When selected mapping type is “Expression”, you can use Expression Editor to build expression. Click “Edit to access the **Expression Editor**. Build your expression using fields from the data source and/or available functions. You can drag and drop items from the list displayed on the right hand side. There are two buttons:

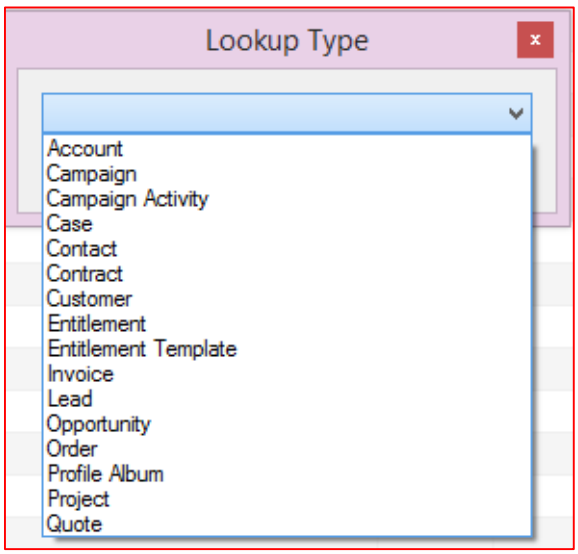
- Save button – saves the expression to the row currently being edited.
- Verify button – checks the expression syntax.



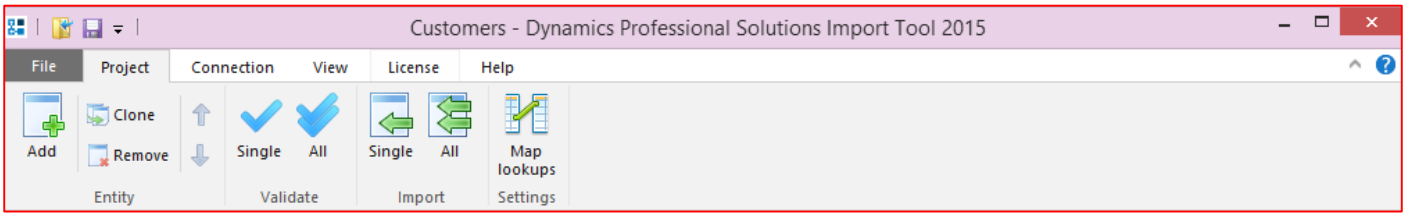
For some field types, like Picklist, Status, Owner, etc. and when selected mapping type is “Source field” you will have to map source values to the CRM values. In this case click on “Map…” link:



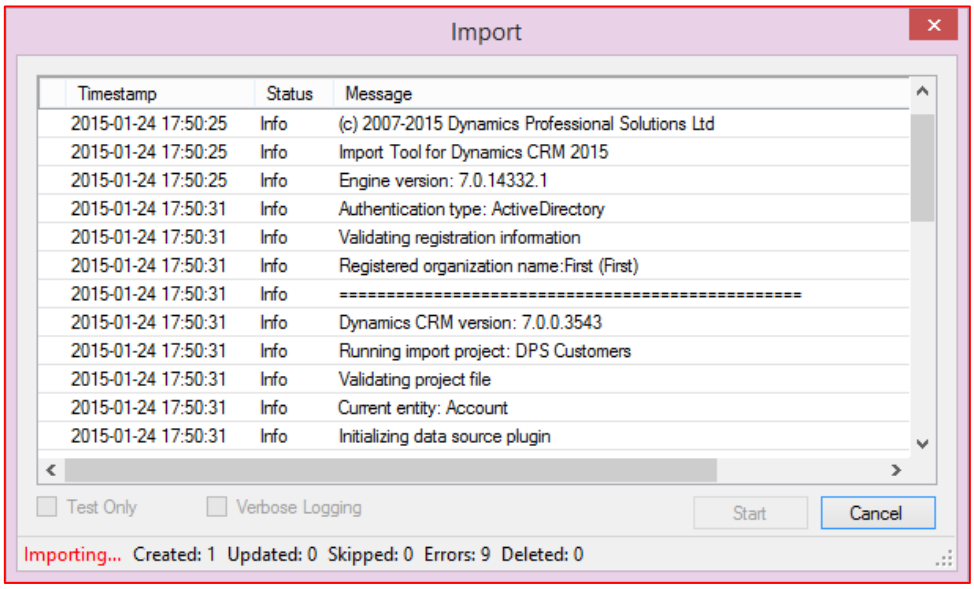
For Customer or some Lookup type fields and when selected mapping type is “Source field” you will have to specify the target lookup type. In this case click on “Target…” link:



When all project settings have been specified, you can validate whether your import will work the way it was intended to using “Validate Single” or “Validate All” commands.



At this point no import is actually executed. The program only simulates the import process to locate any potential problems. You can study the results in the Import screen by scrolling through the validation results. To save results to a text file or copy displayed information to the Clipboard, right click on the window (or click the appropriate icon near “Save” button) and select the action. Verbose Logging option specifies that much more information will be logged and displayed in the progress window. It is usually useful when troubleshooting import issues.



To perform the import, select “Import Single” or “Import All” command. If you have just performed validation and Import screen is still open, simply unmark the “Test Only” checkbox in the left bottom corner of the screen and click the Start button.

Another way of running the import is to use the command line version of the ImportTool **ImportToolCmd.exe** which can be found in the Import Tool folder (typically "C:\Program Files\Dynamics Professional Solutions\ImportTool").

This can be useful when you want to run import on a regular basis (scheduled). For example you can create a schedule task using Windows Task Scheduler or any other scheduler software that will execute ImportToolCmd.exe. Please note that ImportToolCmd.exe must run under account that is a valid CRM user with all required privileges.

Options

ImportToolCmd /p projectfile [/c connection] [/pwd password] [/f pluginfile] [/l logfile] [/s]

Option	Description
/p	Required. Specifies the import project file to run.
/c	Specifies the connection to use. If you have only one connection defined then this parameter is optional. Otherwise it is required.
/pwd	If password is not saved in the connection then it is required to provide a password.
/f	Optional. Specifies the custom plugin file. May be specified more than once.
/l	Optional. Specifies the log file name. When parameter is not specified all messages are written to the screen only.
/s	Optional. When specified import runs in the simulation mode. Otherwise import runs in normal mode.

Examples

Run import (default connection):

ImportToolCmd.exe /p "C:\My Projects\ImportAccounts.dpsi"

Run import with specified connection:

ImportToolCmd.exe /p "C:\My Projects\ImportAccounts.dpsi" /c "My CRM 2011"

Run import and log messages to Import.log:

ImportToolCmd.exe /p "C:\My Projects\ImportAccounts.dpsi" /l C:\Import.log

Run import in test (validation) mode:

ImportToolCmd.exe /p "C:\My Projects\ImportAccounts.dpsi" /s

Display help:

ImportToolCmd.exe

One of the powerful features you can find in the Import Tool are expressions. Expressions are useful in complex import scenarios where default field mapping is not sufficient.

Syntax used by the expressions in the Import Tool is very simple and if you are familiar with basic concepts of programming they should be easy to use. There are four main elements used in the expressions:

- Literals
- Variables
- Operators
- Functions

1. Literals

Literals represent values that are basically constant.

For example:

- "John Smith" is a string
- 127.54 is a decimal
- 13 is an integer
- #2007-12-27# is a date

2. Variables

Variables represent fields that are available in the input file. For example, if your input file has a column named "Last Name" it can be referred as [Last Name]. Value of [Last Name] depends on the actual data in your input file. You can use any column name that is available in your input file regardless whether this column is imported or not. Generic rule is that you should use square brackets ([]) to indicate input file field.

3. Operators

Operators are special functions that operate on literals, variables or other functions. The list of available operators:

- Addition '+'
- Subtraction '−'
- Multiplication '*'
- Division '\'
- Equal '='
- Not equal '<>'
- Less than '<'
- Less than or equal to '<='
- Greater than '>'
- Greater than or equal to '>='
- Logical AND 'AND' or '&'
- Logical OR 'OR' or '|'
- Logical NOT 'NOT' or '!'

For example:

- "John" + " Smith" equals "John Smith"

- 127.12 + 3 equals 130.12

It is important that operators require compatible arguments. For example "#2007-01-12# + 10" is invalid since first argument is a date, second is a number. In this case we can use a function if intention was to add 10 days to the specified date: DateAdd("d", 10, #2007-01-12#).

4. Functions

Functions are described in detail in the next chapter.

Some examples of expressions that can be used in the Import Tool:

- *IIf([Last Name] ="Smit", [Last Name] + "h", [Last Name])*
  This example appends *missing "h" letter in the "Smit" name.*
- *DateAdd*("m", 1, Date())
  This example returns a date that is 1 month from today.
- *[Current Price] + 10*
  This example increments current price by 10.

This chapter describes useful function that can be used within DPS Import Tool.

1. Conversion functions

CBool Function

The CBool function converts an expression to type Boolean.

Syntax

CBool(expression)

Parameter	Description
expression	Required. Any valid expression. A nonzero value returns True, zero returns False. An error occurs if the expression cannot be interpreted as a Boolean value.

Example 1

CBool (0)

Returns False

Example 2

CBool(32)
Returns True

Example 3

CBool(“True”)
Returns True

CCur Function

The CCur function converts an expression to type Currency.
Note: This function is locale and culture aware.

Syntax

CCur(expression)

Parameter	Description
expression	Required. Any valid expression.

Example 1

CCur(

“100.25”)
Returns 100.25

CDate Function

The CDate function converts a valid date and time expression to type Date.
Note: This function is locale and culture aware.

Syntax

CDate(expression)

Parameter	Description
expression	Required. Any valid expression (like Date() or Now()).

Example 1

Date only
CDate("2007-09-01")

Example 2

Date and Time
CDate("2007-09-01 11:12:23")

CDbl Function

The CDbl function converts an expression to type Double.
Note: This function is locale and culture aware.

Syntax

CDbl(expression)

Parameter	Description
expression	Required. Any valid expression.

Example 1

CDbl("28.11")
Returns 28.11

CInt Function

The CInt function converts an expression to type Integer.

Syntax

CInt(expression)

Parameter	Description
expression	Required. Any valid expression.

Example 1

CInt(127.98)
Returns 128

CStr Function

The CStr function converts an expression to type String.

Syntax

CStr(expression)

Parameter	Description

expression	Required. Any valid expression. If expression is: Boolean - then the CStr function will return a string containing True or False. Date - then the CStr function will return a string that contains a date in the short-date format. Numeric - then the CStr function will return a string that contains the number.	Example 1 CStr(127.98) Returns "127.98"

2. Date and time functions

Date Function

The Date function returns the current system date.

Syntax

```
Date()
```

Example 1

```
Return current system date
Date()
```

DateAdd Function

The DateAdd function returns a date to which a specified time interval has been added.

Syntax

```
DateAdd(interval, number, date)
```

Parameter	Description
interval	Required. The interval you want to add. Can take the following values: yyyy - Year q - Quarter m - Month y - Day of year d - Day w - Weekday ww - Week of year h - Hour n - Minute s - Second
number	Required. The number of interval you want to add. Can either be positive, for dates in the future, or negative for dates in the past.
date	Required. Variable or literal representing the date to which interval is added.

Example 1

```
Add one month to current date
DateAdd("m", 1, Date())
```

Example 2

```
Subtract 14 days from current date
DateAdd("d", -14, Date())
```

Hour Function

The Hour function returns a number between 0 and 23 that represents the hour of the day.

Syntax

```
Hour(time)
```

Parameter	Description
time	Required. Any expression that can represent a time.

Example 1

```
Return current hour
Hour(Now())
```

Minute Function

The Minute function returns a number between 0 and 59 that represents the minute of the hour.

Syntax

```
Minute(time)
```

Parameter	Description
time	Required. Any expression that can represent a time.

Example 1

```
Return current minute
Minute(Now())
```

Month Function

The Month function returns a number between 1 and 12 that represents the month of the year.

Syntax

```
Month(date)
```

Parameter	Description
date	Required. Any expression that can represent a date.

Example 1

```
Return current moth
Month(Date())
```

Now Function

The Now function returns the current date and time according to the setting of your computer's system date and time.

Syntax

```
Now()
```

Example 1

```
Return current date and time
Now()
```

Second Function

The Second function returns a number between 0 and 59 that represents the second of the minute.

Syntax

```
Second(time)
```

Parameter	Description
time	Required. Any expression that can represent a time.

Example 1

```
current seconds
Second(Now())
```

Time Function

The Time function returns the current system time.

Syntax

```
Time()
```

Example 1

```
Return current time
Time()
```

Year Function

The Year function returns a number that represents the year.

Syntax

```
Year(date)
```

Parameter	Description
date	Required. Any expression that can represent a date.

Example 1

```
Return current year
Year(Date())
```

3. String functions

InStr Function

The InStr function returns the position of the first occurrence of one string within another.

Syntax

```
InStr(string1, string2[, ignoreCase])
```

Parameter	Description
string1	Required. The string to be searched.
string2	Required. The string expression to search for.
ignoreCase	Optional. Specifies the string comparison method. Default is True.

Example 1

```
InStr( “Hello”, “o”)
Returns 5
```

Example 2

```
InStr(“Hello”, “L”, True)
Returns 0
```

Example 3

```
InStr(“Hello”, “l”)
Returns 3
```

InStrRev Function

The InStrRev function returns the position of the first occurrence of one string within another. The search begins from the end of string, but the position returned counts from the beginning of the string.

Syntax

```
InStrRev(string1, string2[, ignoreCase])
```

Parameter	Description
string1	Required. The string to be searched.
string2	Required. The string expression to search for.
ignoreCase	Optional. Specifies the string comparison method. Default is True.

Example 1

```
InStrRev(“Hello”, “o”)
Returns 5
```

Example 2

```
InStrRev(“Hello”, “L”, True)
Returns 0
```

Example 3

```
InStrRev(“Hello”, “l”)
Returns 4
```

LCase Function

The LCase function converts a specified string to lowercase.

Syntax

```
LCase(string)
```

Parameter	Description
string	Required. The string to be converted to lowercase.

Example 1

```
LCase(“Hello”)
Returns “hello”
```

Left Function

The Left function returns a specified number of characters from the left side of a string.

Syntax

```
Left(string, length)
```

Parameter	Description
string	Required. The string to return characters from.
length	Required. Specifies how many characters to return. If set to 0, an empty string ("") is returned. If set to greater than or equal to the length of the string, the entire string is returned.

Example 1

```
Left(“Hello”, 2)
Returns “He”
```

Example 2

```
Left(“Hello”, 20)
Returns “Hello”
```

Example 3

```
Left(“Hello”, 0)
Returns “”
```

Len Function

The Len function returns the number of characters in a string.

Syntax

```
Len(string)
```

Parameter	Description
string	Required. A string expression.

Example 1

Len(“Hello”)
Returns 5

LTrim Function

The LTrim function removes spaces on the left side of a string.

Syntax

LTrim(string)

Parameter	Description
string	Required. A string expression.

Example 1

LTrim(“ Hello ”)
Returns “Hello ”

Mid Function

The Mid function returns a specified number of characters from a string.

Syntax

Mid(string, start ,length)

Parameter	Description
string	Required. The string to return characters from.
start	Required. Specifies the starting position. If set to greater than the number of characters in string, it returns an empty string ("").
length	Required. The number of characters to return.

Example 1

Mid(“Hello”, 2, 2)
Returns “el”

Example 2

Mid(“Hello”, 1, 1)
Returns “H”

Example 3

Mid(“Hello”, 3, 20)
Returns “llo”

Right Function

The Right function returns a specified number of characters from the right side of a string.

Syntax

Right(string, length)

Parameter	Description
string	Required. The string to return characters from.
length	Required. Specifies how many characters to return. If set to 0, an empty string ("") is returned. If set to greater than or equal to the length of the string, the entire string is returned.

Example 1

Right(“Hello”, 2)
Returns “lo”

Example 2

Right(“Hello”, 20)
Returns “Hello”

Example 3

Right(“Hello”, 0)
Returns “”

RTrim Function

The RTrim function removes spaces on the right side of a string.

Syntax

RTrim(string)

Parameter	Description
string	Required. A string expression.

Example 1

RTrim(“ Hello ”)
Returns “ Hello”

Space Function

The Space function returns a string that consists of a specified number of spaces.

Syntax

```
Space(number)
```

Parameter	Description
number	Required. The number of spaces you want in the string.

Example 1

```
Space
(2)
Returns “  ”
```

Trim Function

The Trim function removes spaces on both sides of a string.

Syntax

```
Trim(string)
```

Parameter	Description
string	Required. A string expression.

Example 1

```
Trim(
“ Hello ”)
Returns “Hello”
```

UCase Function

The UCase function converts a specified string to uppercase.

Syntax

```
UCase(string)
```

Parameter	Description
string	Required. The string to be converted to uppercase.

Example 1

```
UCase
(“Hello”)
Returns “HELLO”
```

5.

4. Math Functions

Abs Function

The Abs function returns the absolute value of a specified number.

Syntax

```
Abs(number)
```

Parameter	Description
number	Required. A numeric expression.

Example 1

```
Abs(-2)
Returns 2
```

Round Function

The Round function rounds a number.

Syntax

```
Round(expression, numdecimalplaces)
```

Parameter	Description
expression	Required. The numeric expression to be rounded.
numdecimalplaces	Required. Specifies how many places to the right of the decimal are included in the rounding.

Example 1

```
Round(10.244, 2)
Returns 10.24
```

5. Miscellaneous functions

Format Function

The Format function returns an expression formatted using specified format.
Note: This function is locale and culture aware.

Syntax

```
Format(expression, format)
```

Parameter	Description
expression	Required. The expression to be formatted.
format	Required. The format to apply. Can be any .NET Framework format.

Example 1

```
Format(20, "C")
Returns "$20"
```

Example 2

```
Format(64, "X")
Returns "40" (hexadecimal)
```

IIf Function

The IIf function returns one of two objects depending on the evaluation of the condition.

Syntax

```
IIf(condition, truePart, falsePart)
```

Parameter	Description
condition	Required. The condition to be evaluated.
truePart	Required. Part will be returned when condition is True.
falsePart	Required. Part will be returned when condition is False.

Example 1

```
IIf(5 > 1, "It's true", "It's false")
Returns "It's true"
```

Example 2

```
Format([Field]="", Null(),[Field])
Converts empty string to NULL value.
```

IsNull Function

The IsNull function replaces NULL with the specified value.

Syntax

```
IsNull(expression, replacement)
```

Parameter	Description
expression	Required. The expression to check.
replacement	Required. The replacement value to use when expression is NULL.

Example 1

```
IsNull([Field], "Null replacement")
When [Field] is NULL it will return "Null replacement". Otherwise [Field] will be returned.
```

Null Function

The Null function returns the NULL value.

Syntax

```
Null()
```

Example 1

```
Return NULL value
Null()
```

6. CRM Functions

Lookup Function

The Lookup function returns a primary key (ID) for the specified entity. This function can to be used for any CRM field of type Lookup.

Syntax

```
Lookup(entity, value[, field])
```

Parameter	Description
entity	Required. Any valid CRM entity.
value	Required. A value being looked up.
field	Optional. Specifies a primary field used for queries. When parameter is missing the default primary field will be used.

Example 1

```
Lookup("account", "John Smith")
Returns the account object with the specified name.
```

Example 2

```
Lookup("product", "Bicycle-100")
Returns the product object.
```

Example 3

```
Lookup("account", "AB0201", "accountnumber")
Returns the account object with the specified account number.
```

LinkedLookup Function

The Lookup function returns a primary key (ID) value of the specified entity. It also allows linking to another entity. One of the examples is if you want to retrieve a unit record with the specified name for the specified unit group name. Maximum of 4 entities can be linked.

Syntax

```
LinkedLookup(entity1, field1, value1, [entity2, field2, value2…])
```

Parameter	Description
entity1	Required. Primary entity which is any valid CRM entity name.
field1	Required. Entity field that is used as a condition criteria.
value1	Required. Value for the primary condition.
entity2	Secondary CRM entity name that primary entity links to.
field2	Secondary entity field that is used as condition criteria for this entity.
value2	Value for the secondary condition.
…	Additional optional linked entities (entity3, field3, value3), (entity4, field4, value4).

Example

```
LinkedLookup("uom", "name", [UoM], "uomschedule", "name", [UoMGroup])
Returns the unit with name matching data source column UoM and unit group name matching data source column UoMGroup.
```

Sql Function

The Sql function executes a custom query against specified entity.
Note: If this function returns more than one row an error will be raised.

Syntax

```
Sql(entity, field, condition)
```

Parameter	Description

entity	Required. Any valid CRM entity name.
field	Required. A field to return.
condition	Required. A condition. It corresponds to SQL WHERE clause and therefore any valid SQL statement can be used.

Example 1

```
Sql("account", "websiteurl", "name='Mountain Toy Store' AND statuscode=1")
Returns the website url for the account with the specified name.
```

Example 2

```
Sql("product", "productnumber", "description='Aluminum alloy cups; large diameter spindle.'")
Returns the product number for the product with the specified description.
```

PartyList Function

The PartyList function returns a set of entities.
This function is similar to Lookup functions. The difference is that it can return more than one value. It can only be used for fields of PartyList type.

Syntax

```
PartyList(entity1, value1[, entity2, value2, [entity3, value2]...])
```

Parameter	Description
entity1	Required. Any valid CRM entity name.
value1	Required. A value being looked up.
entity2	Optional. See entity1.
value2	Optional. See value1.
...	Additional items to look up.

Example 1

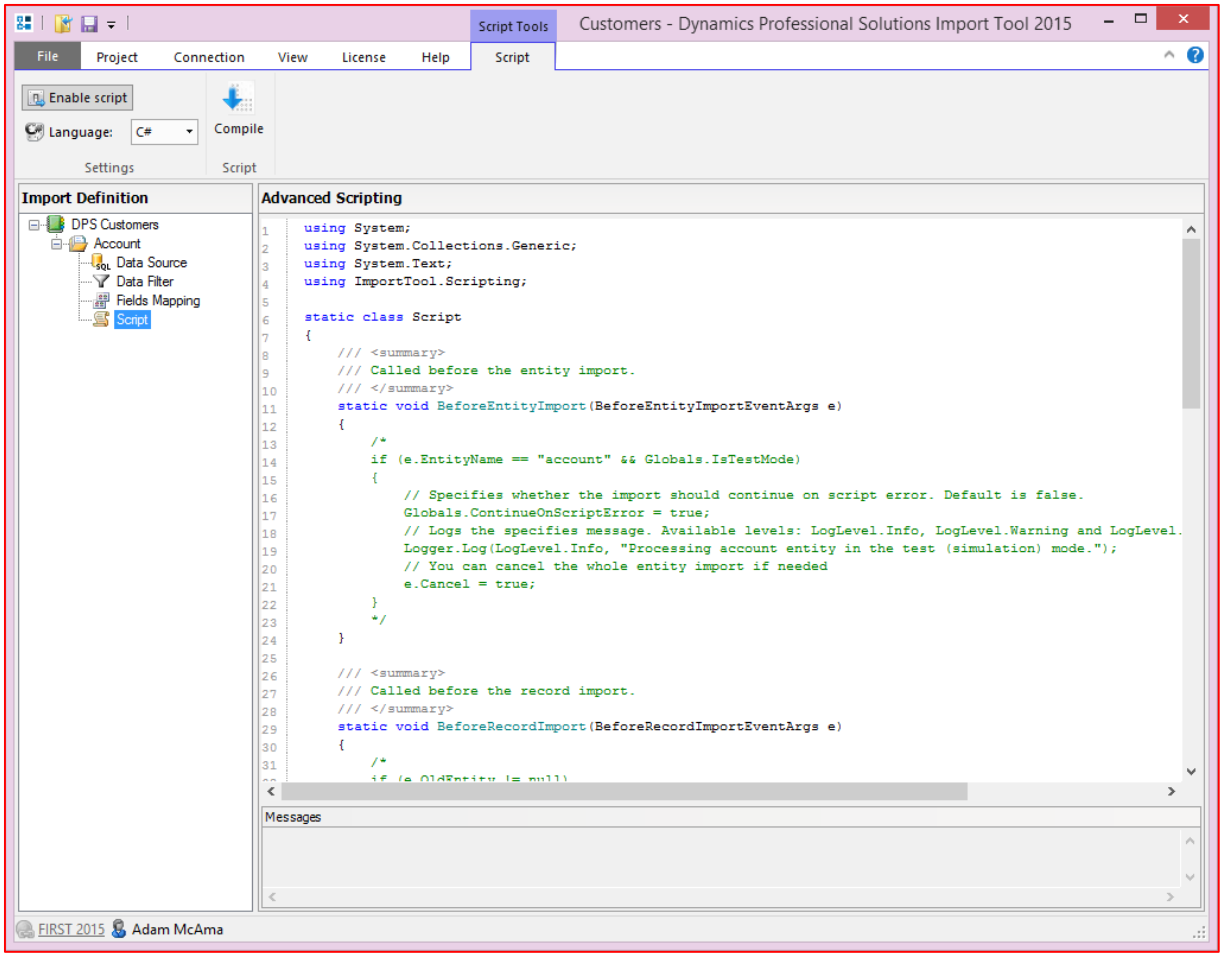
```
PartyList("systemuser", "John Smith", "account", "A Bike Store")
Returns two items.
```

In this version of the Import Tool we are introducing extremely powerful scripting feature that can be used in the scenarios where standard mapping and/or expressions are not sufficient. This is more advanced feature and requires some programming knowledge.

There are two scripting languages currently supported: VB.NET and C#. We recommend using C#, however exactly the same can be achieved using VB.NET.

In the Import Definition for every entity there are now four items available:
- Data Source,
- Data Filter,
- Fields Mapping and
- Script.

When you select Script, Advanced Scripting pane will be displayed. Scripting is disabled by default. To enable it, simply click on the button which shows the current script status (e.g. "Disabled"). Now it should show the "Enabled" status. Clicking on the button when script is enabled will disable the scripting.



Once you enabled the scripting, you will have to decide which language you prefer:

- C# or
- VB.NET.

We recommend using C#, however if you are more familiar with Visual Basic then go ahead and select VB.NET. And remember, once you selected your preferred language, should you decide to change it later you will have to write your script again. There is no automatic conversion between script languages.

When you have selected your preferred language, the template script will be created. Now you will be able to add your own import business logic. But first let's explain the structure of the script.

Use "Compile" button to test the syntax of the script.

a. Default namespaces

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using ImportTool.Scripting;
```

VB.NET

```
Imports System
Imports System.Collections.Generic
Imports System.Text
Imports ImportTool.Scripting
```

This tells which namespaces should be used. You can add additional namespaces if you need, refer to.NET Framework reference on MSDN for more details. You can also specify additional assemblies you would like to use in the application configuration file:

```
<add key="CustomScriptAssemblies" value="System.Data.dll;System.Web.dll"/>
```

b. Script class

This is the main class that must exist. You can safely remove the methods you are not going to use, but you have to keep main class definition. Otherwise your script will not compile.

C#

```
static class Script
{
    static void BeforeEntityImport(BeforeEntityImportEventArgs e)
    {
    }

    static void BeforeRecordImport(BeforeRecordImportEventArgs e)
    {
    }

    static void AfterRecordImport(AfterRecordImportEventArgs e)
    {
    }

    static void AfterEntityImport(AfterEntityImportEventArgs e)
    {
    }
}
```

VB.NET

```
Friend Class Script
    Private Shared Sub BeforeEntityImport(ByVal e As BeforeEntityImportEventArgs)
    End Sub

    Private Shared Sub BeforeRecordImport(ByVal e As BeforeRecordImportEventArgs)
    End Sub

    Private Shared Sub AfterRecordImport(ByVal e As AfterRecordImportEventArgs)
    End Sub
```

```
    Private Shared Sub AfterEntityImport(ByVal e As AfterEntityImportEventArgs)
    End Sub
End Class
```

There are four methods (events) you can add your code to:

- **BeforeEntityImport**
  Method is called when the import of the entity is about to start. You can programmatically decide whether to continue the import of this entity or to skip it.
  BeforeEntityImportEventArgs has the following properties:

Property	Type	Description

Cancel	bool	Specifies whether to skip the import of this entity. If set to true then import of the entity will be skipped. For example:
		C#
		`if (e.EntityName == "product")`
		`{`
		`  e.Cancel = true;`
		`}`
		VB.NET
		`If e.EntityName = "product"`
		`  e.Cancel = True`
		`End If`

EntityName	string	Specifies the name of the entity being imported, for example “account”, “contact”, “product”, etc. Read-only. For example:
		C#
		`if (e.EntityName == "account")`
		`{`
		`  Logger.Log(LogLevel.Info, "Hello from script: " + e.EntityName);`
		`}`
		VB.NET
		`If e.EntityName = "account"`
		`  Logger.Log(LogLevel.Info, "Hello from script: " + e.EntityName)`
		`End If`

- **BeforeRecordImport**
  Method is called when the import of a record is about to start. You can programmatically decide whether to continue the import of this record or to skip it.
  BeforeRecordImportEventArgs has the following properties:

Property	Type	Description

Cancel	bool	Specifies whether to cancel the import of the current record. If set to true then import of the current entity will not start. For example:
		C#
		`if (e.OldEntity != null)`
		`{`
		`  if ((string)e.OldEntity["name"] == "A Store (sample)")`
		`  {`
		`  // Cancel current record`
		`  e.Cancel = true;`
		`  }`
		`  else`
		`  {`
		`  e.NewEntity["description"] = e.OldEntity["name"];`
		`  }`
		`}`
		VB.NET
		`If (Not e.OldEntity Is Nothing) Then`
		`  If (CStr(e.OldEntity.Item("name")) = "A Store (sample)") Then`
		`  ' Cancel current record`

		e.Cancel = True
		Else
		e.NewEntity.Item("description") = e.OldEntity.Item("name")
		End If
		End If

DataSource	DataSource	Provides access to the data used as a data source. You can access data source fields either by index or name. For example:
		C#
		object fieldValueByName = e.DataSource["SourceFieldName"];
		object fieldValueByIndex = e.DataSource[2];
		VB.NET
		Dim fieldValueByName = e.DataSource("SourceFieldName")
		Dim fieldValueByIndex = e.DataSource(2)

NewEntity	Entity	This is the new entity that will be imported (created, updated, etc). You can modify it before the import, for example you can set additional entity property values (even if not configured for import). Generally you have full access to the entity before the actual import starts, so you can apply your own business logic.
OldEntity	Entity	This is the original (existing) entity that will be updated. This property will not be specified for new records.

Entity data type is fully described in Microsoft Dynamics CRM 2011 SDK which can be downloaded from MSDN.

- **AfterRecordImport**
  Method is called after the import of a record.
  AfterRecordImportEventArgs has the following properties:

Property	Type	Description
Id	Guid?	Specifies the Id of the newly created record.

		Specifies whether to cancel the import of the current record. If set to true then import of the current entity will not start. For example:
		C#
		if (e.OldEntity != null)
		{
		if ((string)e.OldEntity["name"] == "A Store (sample)")
		{
		// Cancel current record
		e.Cancel = true;
		}
		else
		{
		e.NewEntity["description"] = e.OldEntity["name"];
		}
		}
		VB.NET
		If (Not e.OldEntity Is Nothing) Then
		If (CStr(e.OldEntity.Item("name")) = "A Store (sample)") Then
		' Cancel current record
		e.Cancel = True
		Else
		e.NewEntity.Item("description") = e.OldEntity.Item("name")
		End If
		End If

DataSource	DataSource	Provides access to the data used as a data source. You can access data source fields either by index or name. For example:
		C#
		object fieldValueByName = e.DataSource["SourceFieldName"];
		object fieldValueByIndex = e.DataSource[2];

		VB.NET
		`Dim fieldValueByName = e.DataSource("SourceFieldName")`
		`Dim fieldValueByIndex = e.DataSource(2)`

NewEntity	Entity	This is the new entity that will be imported (created, updated, etc). You can modify it before the import, for example you can set additional entity property values (even if not configured for import). Generally you have full access to the entity before the actual import starts, so you can apply your own business logic.
OldEntity	Entity	This is the original (existing) entity that will be updated. This property will not be specified for new records.

- **AfterEntityImport**
  Method is called after the import of an entity.
  AfterEntityImportEventArgs provides the arguments for the method:

Property	Type	Description

EntityName	string	Specifies the name of the entity that was imported, for example "account", "contact", "product", etc. Read-only. For example:
		C#
		`if (e.EntityName == "account")`
		`{`
		`  Logger.Log(LogLevel.Info, "Hello from script: " + e.EntityName);`
		`}`
		VB.NET
		`If e.EntityName = "account"`
		`  Logger.Log(LogLevel.Info, "Hello from script: " + e.EntityName)`
		`End If`

c. *Global objects and helper methods*

Logger

It represents the Logger class you can use in your script to log errors, warnings and messages. There are 3 levels available: Info, Warning and Error. Messages will appear in standard import log.

Examples:

C#

```
Logger.Log(LogLevel.Info, "Information");
Logger.Log(LogLevel.Warning, "Warning");
Logger.Log(LogLevel.Error, "Error");
Logger.Log("Information too");
```

VB.NET

```
Logger.Log(LogLevel.Info, "Information")
Logger.Log(LogLevel.Warning, "Warning")
Logger.Log(LogLevel.Error, "Error")
Logger.Log("Information too")
```

Globals

It represents the class that provides some global variables.

Property	Type	Description

ContinueOnScriptError	bool	Specifies whether the import should continue if there is an error in the custom script. Default value is true.
		Examples:
		C#
		`Globals.ContinueOnScriptError = true;`
		VB.NET
		`Globals.ContinueOnScriptError = True`

CustomData	object	This property can be used to store any data during the import. It can be useful if you need to preserve some information about previously imported records, etc.
IsTestMode	bool	Indicates whether the import is running in the test mode or not.
OrganizationServiceProxy		Provides the direct access to the organization web service. More information about this class you can find in the Dynamics CRM SDK.

Query

It provides methods for querying the CRM.

- `List<Guid> Lookup(string entityName, string attributeName, object value)`
- `EntityCollection Execute(QueryExpression query)`

Above methods are provided for the convenience only as you can use Globals.OrganizationServiceProxy to the same.

Support

If you have any questions or problem with our products contact our support team. Submit your question using New Support Request Form on our Web page. A member of our support team will get in touch with you as soon as possible.

Support:	http://www.dynamics-pros.com/support
Download:	http://www.dynamics-pros.com/downloads

DYNAMICS
PROFESSIONAL SOLUTIONS