

Data Importer

Setup

Install the Sitecore package. This will install templates, some core custom application items, a bin file, Sitecore modules files and a system folder under /Sitecore/system/modules/data imports. This system section is where you will define your imports. You can import data from the Sitecore tree or from a SQL database. You will need to configure any connection strings to the SQL server in your web.config or connectionstrings.config file.

Usage

There are two importer types: Sitecore Importer and SQL importer. The importers will contain an XPath or SQL Query respectively that will return a result set. This importer utility will loop through each row, Item or DataRow respectively, in that result set and transfer the individual mapping values you define from the existing Item or DataRow to a new item created in Sitecore. There are two mapping types: Fields and Properties. Property mappings only apply to Sitecore Imports and defines a property on a Sitecore Item, such as path, to a field on the new item created.

The main application is used from the desktop mode so you will need to log into that interface. Then go to the Sitecore start menu->Development Tools->Data Import. This will pop up an application window. Any SQL or Sitecore import maps you had previously defined would be displayed in the drop downs for each section respectively.

To create a new import map, right-click on the Data Imports folder and insert either a SQL Import or Sitecore Import. Fill out the fields. Some fields may not need to be filled out like the "Item Foldering" section. This applies only if you want to folder items by Name or Date. Under each importer there will be a fields folder and under the Sitecore importer there will be a properties folder. If you right-click and insert you can create several different types of field imports. A basic type is ToText. This will transfer the data from a field as text to the new field. There is also a ToStaticValue field type that will allow you to set a static value that every item should have when they're created.

Field Types

ListToGuid

This Type is used to match the import value to the display name on a list of child items. You would set the list in the field and all the child item's names will be compared. If there is a match then the GUID of the matched item will be stored in the new field. This is useful when you've imported an enumeration previously and you want to import items that point to the enumeration using link fields like DropLink

ToDate

This is used to import date values into DateTime field types.

ToStaticValue

This is used when all the import items need to have a field populated with the same value. For example if you're importing a list of blog articles and you want to set the value of a DropLink field to the same GUID

ToText

This is used to set the text value of the import to the TextField of the new item.

UrlToLink

This is used to set the import value which would be a url to the URL property of the LinkField..

Property Types

Property mappings only apply to Sitecore Imports and defines a property on a Sitecore Item, such as path, to a field on the new item created.

PathToText

This will set the Item.Paths.Path property of the imported item to a TextField on the new item

UrlToText

This will set the url of the imported item to the TextField on the new item.

Customization

The system was built to be modular so that you could define your own mappings or importers.

Custom Mappings

To create your own custom mappings you will need to create a template item and class to handle the template. You could also download the source code to build the class library with your custom mappings in them or add them to your own library and reference them properly as explained below. Property Mappings are only for Sitecore imports since properties are only related to Sitecore items.

Creating the Template

You would need to define a template and extend from either BaseField or BaseProperty depending on if you are creating a Field or Property Mapping. You could also inherit from another existing class. You can also add any fields to store information you want on this template. The field values on this template will be manipulated in the class you define later. Now you would need to create a Standard Values for that template. In the Standard Values item you will see there are two inherited fields: Handler Class and Handler Assembly. These are used to determine which class runs this mapping. The "Handler Assembly" field should be populated with the "Sitecore.SharedSource.DataImporter". This assumes you will be downloading the source and adding your fields to this class library. If you are not you can override the value in this field and then set the "Handler Class" field to the fully qualified name of your class (namespace and class name).

Creating the Class

The class you create should inherit from the BaseField or BaseProperty depending on if you are creating a Field or Property Mapping. You could also inherit from existing classes. In the new class the constructor should call the constructor of the BaseField or BaseProperty class. It should also implement the two FillField methods. One is to handle SQL imports and the other is to handle Sitecore imports. They're separated because the import rows are different types. The FillField methods can be used to manage one field or if you're in a hurry many fields. I was trying to define types that would be reusable so I decided to break each field into its own class but since you have access to both the existing DataRow and the new item. You can do much more than just handle a single field. You could write an entirely custom import in this one class. It's intended for you to be able to manage the process of transforming the data so do what you will.

Insert Options

Once you've created your mappings update the Fields or Properties Folder insert options. This will allow you to create the mappings in the system and run the import utility against them.

Custom Importer - Support for Oracle

I know that some people use Oracle. I haven't used Oracle so I don't know what form a resulting row is like. If it is a standard .NET DataRow You should be able to use the SQL importer. If this is not the case you need to create a new DataMap Class and sitecore template similar to the field and property mappings above. You would do the same for importing XML as well. From here you can add fields to the template and add them as properties to the class. The class should inherit from BaseDataMap and call the constructor on the BaseDataMap on its own constructor. You will also need to implement a method called Process. You can largely just copy from the SQLDataMap.Process method and make changes to query the data and loop through the result set.

When you create a new import map type you will be iterating through a different object type instead of DataRow or Items. To support this type you'll want to add a FillField method to the BaseField class so that each of the existing field classes will be forced to implement it to handle consuming the custom row's specific object type.

To support the right-click insert options you'll also want to create a branch template with a Field Folder as a child item. Change the branch item icon to match the template icon and add the new branch template to the insert options of the Data Import Folder and Import Map Folder.