

Using the Import/Export module in Dynamicweb eCommerce

LEGAL INFORMATION

© Copyright 2007 Dynamicweb Software A/S (Ltd).

All rights reserved. Alteration or reproduction of this document or parts hereof is strictly prohibited, regardless of form or means, unless explicit permission has been acquired from Dynamicweb Software.

Dynamicweb is a registered trademark of Dynamicweb Software.

Company and product names mentioned in this document may be registered trademarks or trademarks of third parties.

CONTENTS

Introduction	v
What this document is about.....	v
Who this document is for.....	v
Related documents.....	v
1 Import/Export module	1
1.1 Pipeline overview.....	1
1.2 Editing interface	2
• 1.2.1 Interface elements.....	2
• 1.2.2 Drag and drop.....	3
1.3 Activities	5
• 1.3.1 Files.....	5
• 1.3.2 eCom specialized	7
• 1.3.3 Transitions.....	10
• 1.3.4 Adding your own activities	11
1.4 Run pipeline.....	11
1.5 Reuse pipeline.....	11
2 Sample pipelines	12
2.1 Import product groups from XML file to database....	12
2.2 Export product groups from database to XML file	13
2.3 Import products from XML file to database	14
2.4 Export products from database to XML file.....	15
3 Activity - Code sample	17
3.1 Datatype conditions.	17
3.2 How do I decide what my activity should do?	17
3.3 Code sample	17

INTRODUCTION

What this document is about

This document gives a general description of how to use the *Import/Export* module for Dynamicweb eCommerce.

Who this document is for

This document is primarily for developers who need to import and export data to and from a Dynamicweb eCommerce solution.



A basic knowledge of XML, XSLT and XPath is required in order to use the *Import/Export* module.

Related documents

- **8881 Dynamicweb eCommerce, Pre-Release Notes:** Gives a general description of the contents of Dynamicweb eCommerce.
- **9221 Document Conventions:** Gives an overview of the typographical and writing conventions used in manuals and documentation from Dynamicweb Software.

1 IMPORT/EXPORT MODULE

The following describes how to use the Dynamicweb *Import/Export* module. This includes setting up specific data transfer pipelines through a series of steps.

Basically the Import/Export module enables you to import data (e.g. products, prices, etc.) from other off-line systems. You can also export eCommerce data for further processing in other, off-line systems.

Import and export of data are set up as so-called pipelines, which is a term used for moving data in and out of a system, such as Dynamicweb eCommerce. Clicking *Import/Export* in the navigation pane will display a list of already created pipelines.



How are pipelines used in Dynamicweb eCommerce?

The *Import/Export* module uses pipelines to move data in and out of Dynamicweb eCommerce. The way data is moved depends on how you configure the pipeline.

Pipelines are configured by choosing activities and placing them in a sequence.

The module stores all pipelines in an XML file format in the Dynamicweb *File Manager*.

1.1 Pipeline overview

The module settings are accessed through the Dynamicweb eCommerce *Navigation pane* by clicking the *Import/Export* icon. Here, you can see a list of all pipelines created for your Dynamicweb eCommerce solution.

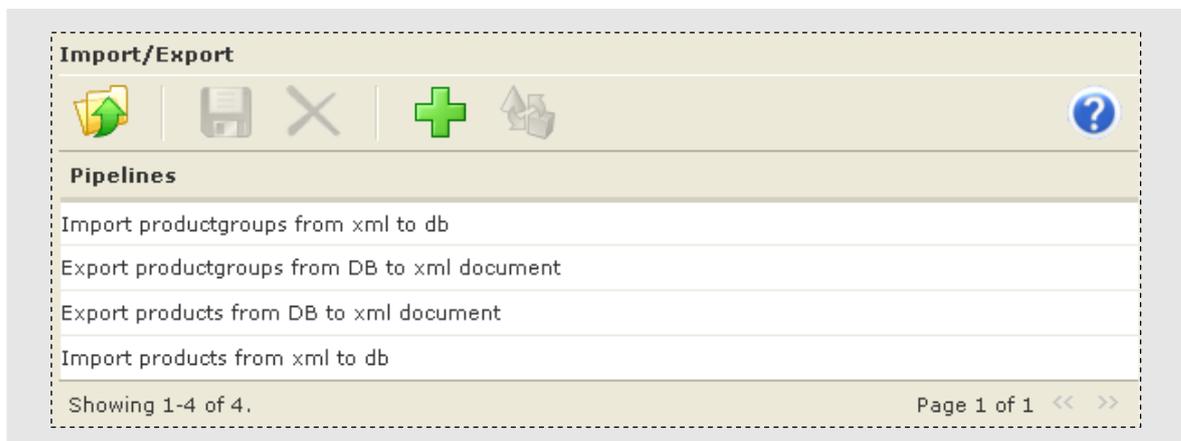


Figure 1. Pipeline list.

The first time you access the module settings, you will see four pipelines in the list. These pipelines are delivered with the Import/Export module as sample pipelines for you to use as you like (see 1.5 Reuse pipeline).

You can add new pipelines by clicking the *New pipeline* icon in the toolbar (see *Figure 1. Pipeline list.*). This opens a separate pipeline design interface where you can configure new and existing pipelines. You can read about the different possibilities in the following sections.

1.2 Editing interface

The *Import/Export* module offers a slightly different editing interface than what you might know from other places in Dynamicweb and Dynamicweb eCommerce. The module offers extensive use of drag-and-drop functionality.

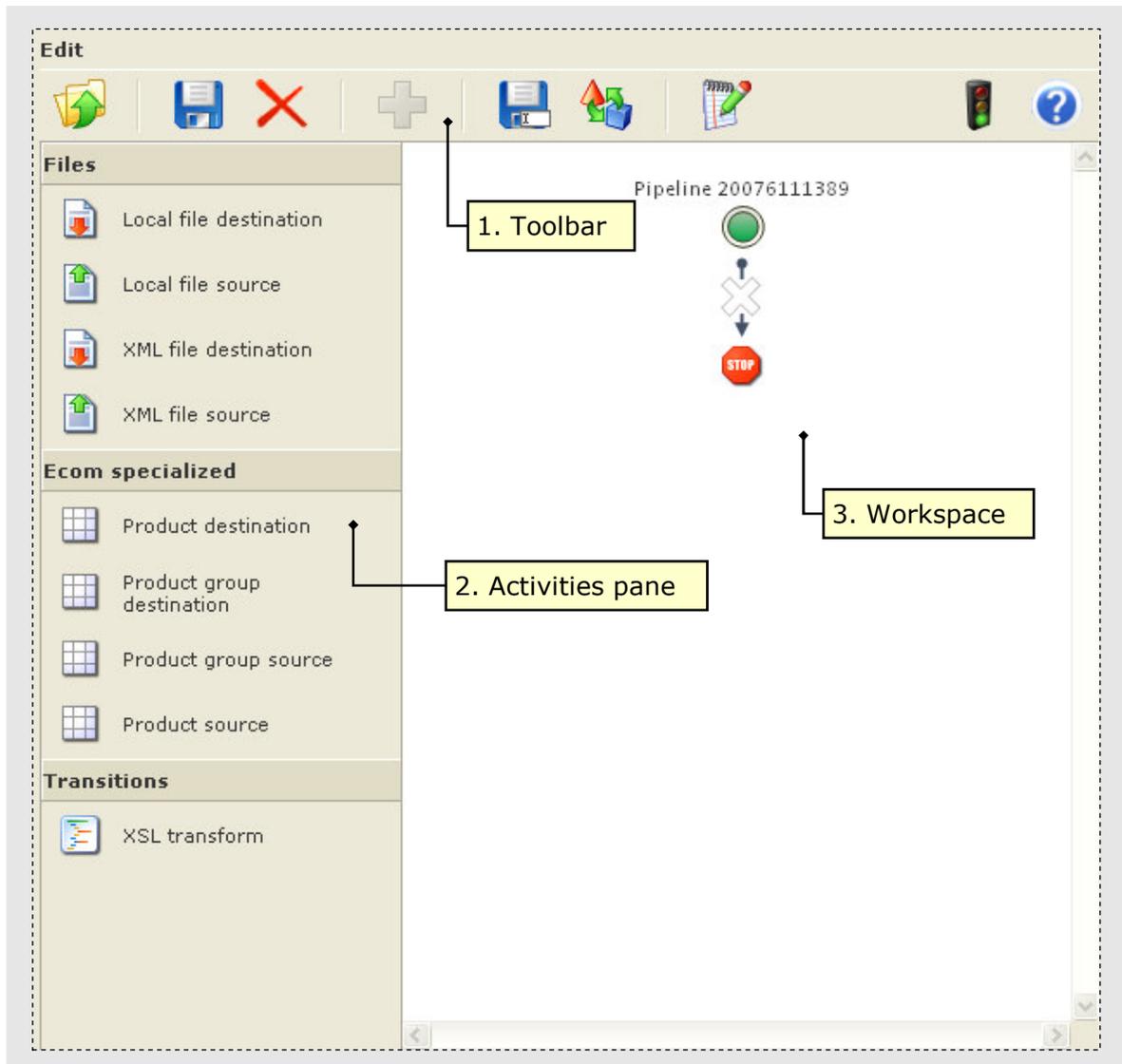


Figure 2. The pipeline editing interface.

1.2.1 Interface elements

The pipeline editing interface consists of three main elements:

1. **Toolbar**

In the toolbar you have a number of functions available, which you can use to edit, save and run import/export procedures.



Save as...

Click this icon to save the pipeline with a new name. See [1.5 Reuse pipeline](#)



Run

Click this icon to run the pipeline. See [1.4 Run pipeline](#).



Edit pipeline

Click this icon to edit the pipeline properties.

**“Traffic light”**

Indicates whether an activity is successfully inserted in a sequence.

2. Activities pane

The *Activities pane* is where you can choose different elements for the data import/export. Consider this pane as a toolbox where you can pick up the tools to create the most fitting import or export of data to and from your Dynamicweb eCommerce solution.

3. Workspace

The *Workspace* is where you manage your pipelines. In order to create pipelines, you must drag elements from the *Activities pane* onto the *Workspace*. Once an element is placed on the *Workspace*, you can move it around as you like though with a few restrictions such as placing a file destination before a file source.

On the workspace you can see a graphical representation of a data sequence. The sequence starts with a green icon and stops by the *Stop* icon. In between the two icons, you can place as many activities from the *Activities pane* as you like. Above the green starting point, you can see the pipeline name which you can change by clicking *Edit* in the toolbar.

1.2.2 Drag and drop

Dragging elements onto the *Workspace* is a central part of configuring and editing pipelines. As explained under the *Activities pane* explanation, you can select an element from the *Activities pane* and drag the element to the *Workspace*. Once the activity is hovering over the *Workspace* a green plus icon will indicate where you can place an activity in the sequence. Placing the activity over the plus icon will transform this icon to a green line. This indicates that the activity will be placed that particular place in the sequence.

If you regret the placement of an activity, you can delete it by selecting the activity on the *Workspace* and press . If you want to move an activity to another place in the sequence then simply select the activity and drag it to the desired place. Hovering over an activity will display a tooltip that in short explains the purpose of the activity.

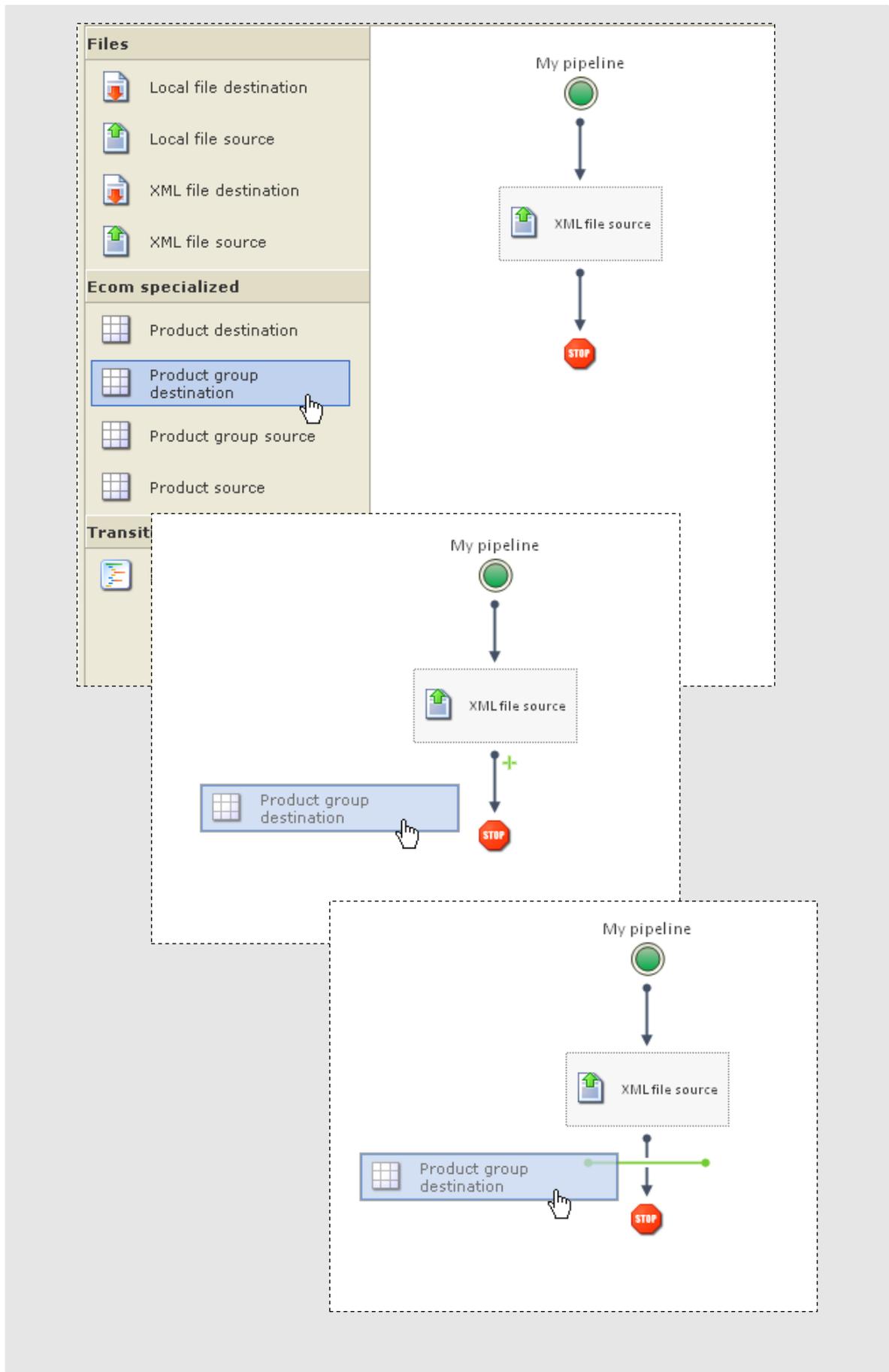


Figure 3. Dragging an activity onto the Workspace.

1.3 Activities

The activities represented in the *Activities pane* are the building blocks of your import/export pipelines. The different activities have various purposes that need to be placed in the right order on the *Workspace*. You can read more about the different activity types in the following.



An activity in detail

An activity is like a link in a big chain. The first link in the chain retrieves data and gives it to the next link, and that link gives it to the next and so on until the end of the chain is reached. This procedure is done over and over again until the first link has no more data.

The *Import/Export* module offers the following standard groupings of activities:

1. **Files**

Contains activities that reads or writes data to and from local files such as CSV and XML files.

2. **Ecom specialized**

Contains activities for placing and extracting data to and from Dynamicweb eCommerce.

3. **Transitions**

Contains an activity for transforming data that is either extracted from or about to be imported to Dynamicweb eCommerce.

Right-clicking an activity on the *Workspace* allows you to change the activity properties. In the following sections you can read about how to configure the different activities.

1.3.1 Files

Local file destination

The activity allows you to make a reference to a destination file in the *File Manager*.

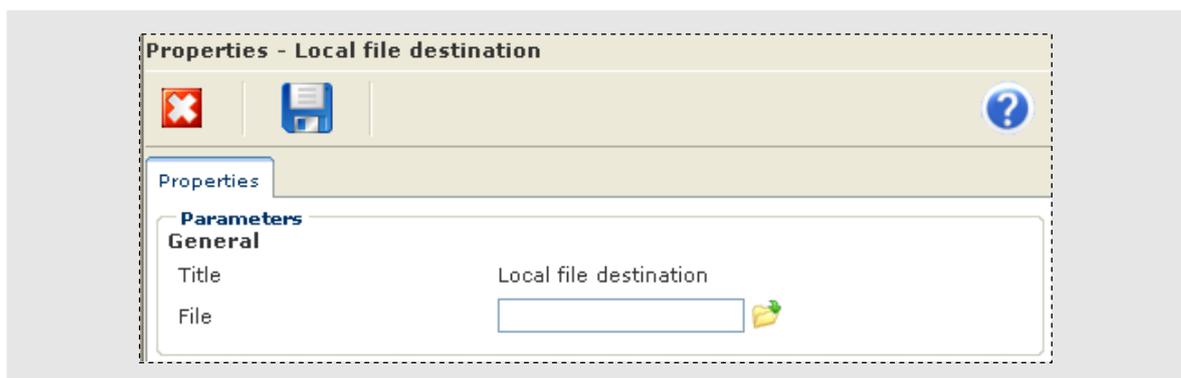


Figure 4. Local file destination properties.

- **File**

Here, you can choose a file from the *File Manager* where the data exported from Dynamicweb eCommerce will be inserted.

Local file source

The activity allows you to make a reference to a source file in the *File Manager*.



Figure 5. Local file source properties.

- File**
 Here, you can choose a file from the *File Manager* from where data (such as product information) has to be inserted into Dynamicweb eCommerce.

XML file destination

The activity allows you to make a reference to an XML destination file in the *File Manager*

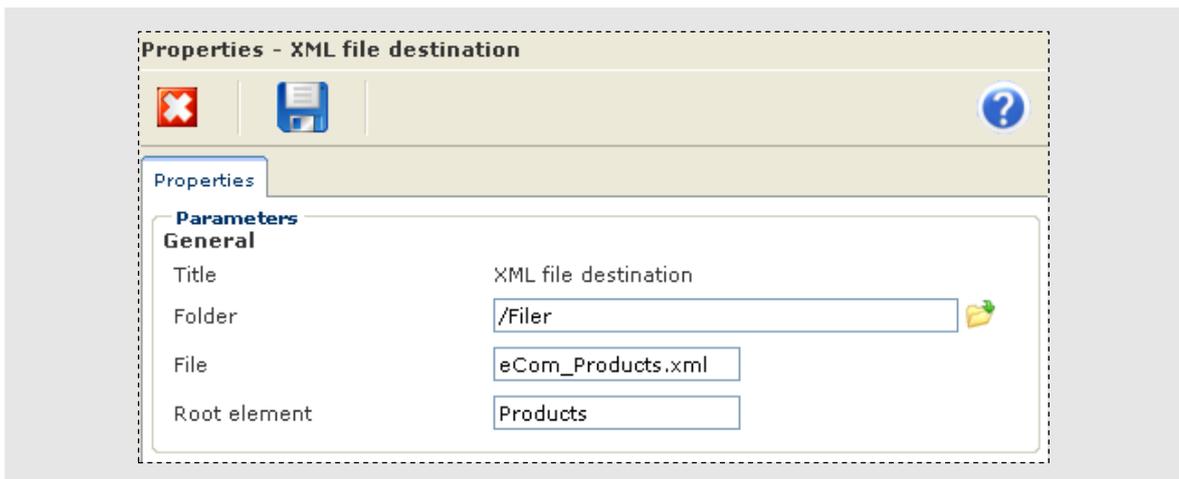


Figure 6. XML file destination properties.

- Folder**
 Here, you can choose the folder in the *File Manager*, where the XML file you want to export data to, is placed.
- File**
 Here, you must enter the exact XML file name, e.g. `eCom_ProductGroups.xml`.
- Root element**
 Enter the root element of the XML file chosen above, e.g. *Groups*.



What is a root element?

Each XML document has exactly one root element and is the parent element to all other elements in the XML document. In an XML file relevant for Dynamicweb eCommerce this could be *Groups*.

XML file source

The activity allows you to make a reference to an XML source file in the *File Manager*.

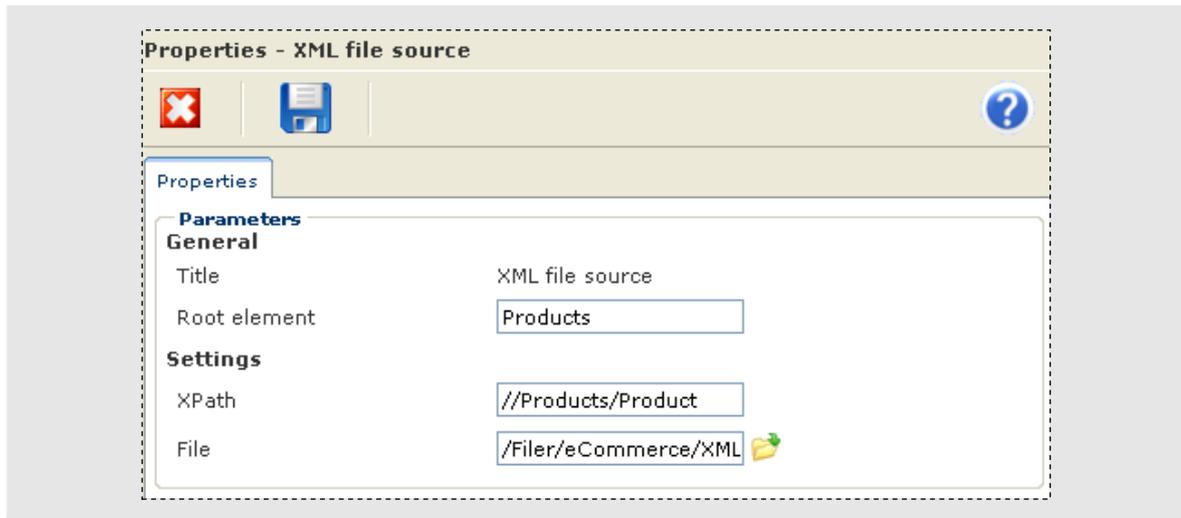


Figure 7. XML file source properties.

General

- **Root element**

Enter the root element of the XML file chosen under *Settings*, e.g. *Groups*.

Settings

- **XPath**

Here, you can enter an XPath statement in order to filter the XML data before it is inserted in Dynamicweb eCommerce.



What is XPath?

Basically, XPath is a language for addressing parts of an XML document. An XPath statement uses the same path notation as in URLs for navigating through the hierarchical structure of an XML document. An example of an XPath statement for Dynamicweb eCommerce relevant files could be `//Products/Product`.

- **File**

Here, you must enter the exact XML file name where data has to be read from, e.g. `eCom_ProductGroups.xml`.

1.3.2 eCom specialized

The *eCom specialized* activity group contains activities that insert or return data to and from product groups in Dynamicweb eCommerce.

Product destination

The activity allows you to make a reference to a product group in Dynamicweb eCommerce.

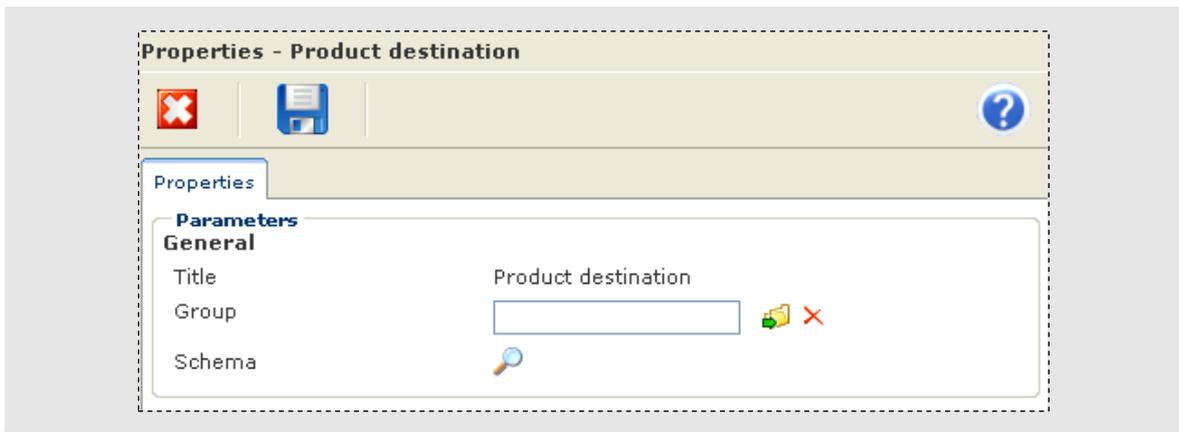


Figure 8. Product destination properties.

General

- **Group**
Click the *Choose folder* icon to choose one or more product groups from the group navigation
- **Schema**
Click the *View document* icon to view the XML schema used in Dynamicweb eCommerce.



What is an XML schema?

Basically, the XML schema describes the structure of a given XML document. This means that the main purpose of the XML schema is to define what you could call the allowed building blocks of the XML document.

Product group destination

The activity allows you to make a reference to a product group in Dynamicweb eCommerce

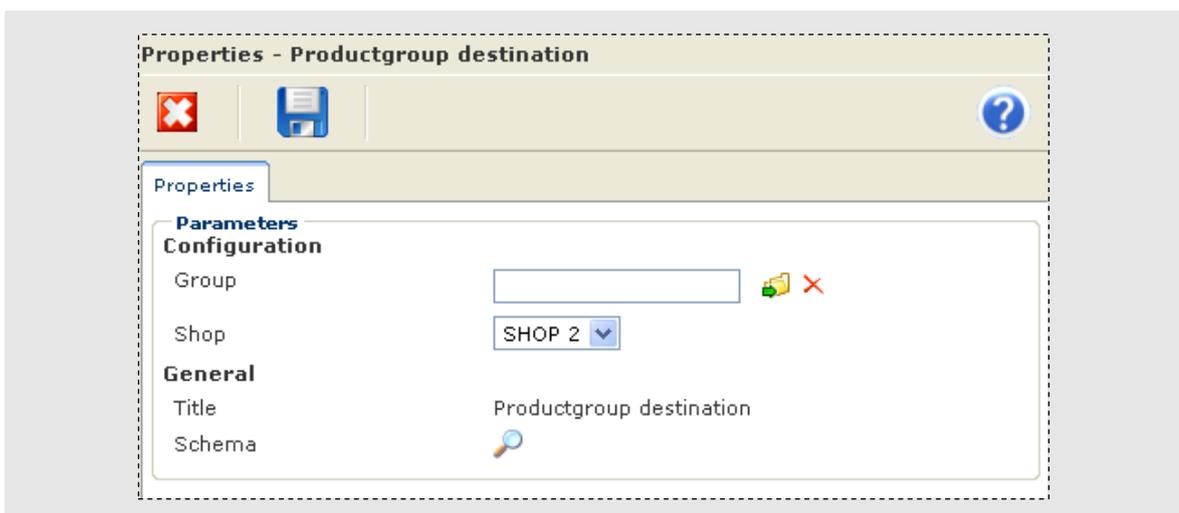


Figure 9. Product group destination properties.

Configuration

- **Group**
Click the *Choose folder* icon to choose one or more product groups from the group navigation.
- **Shop**
Choose the shop that you want to insert the product group into. You

can choose between all created shops in your Dynamicweb eCommerce solution.

General

- **Schema**

Click the *View document* icon to view the XML schema used in Dynamicweb eCommerce.

Product group source

The activity allows you to make a reference to a product group in Dynamicweb eCommerce.

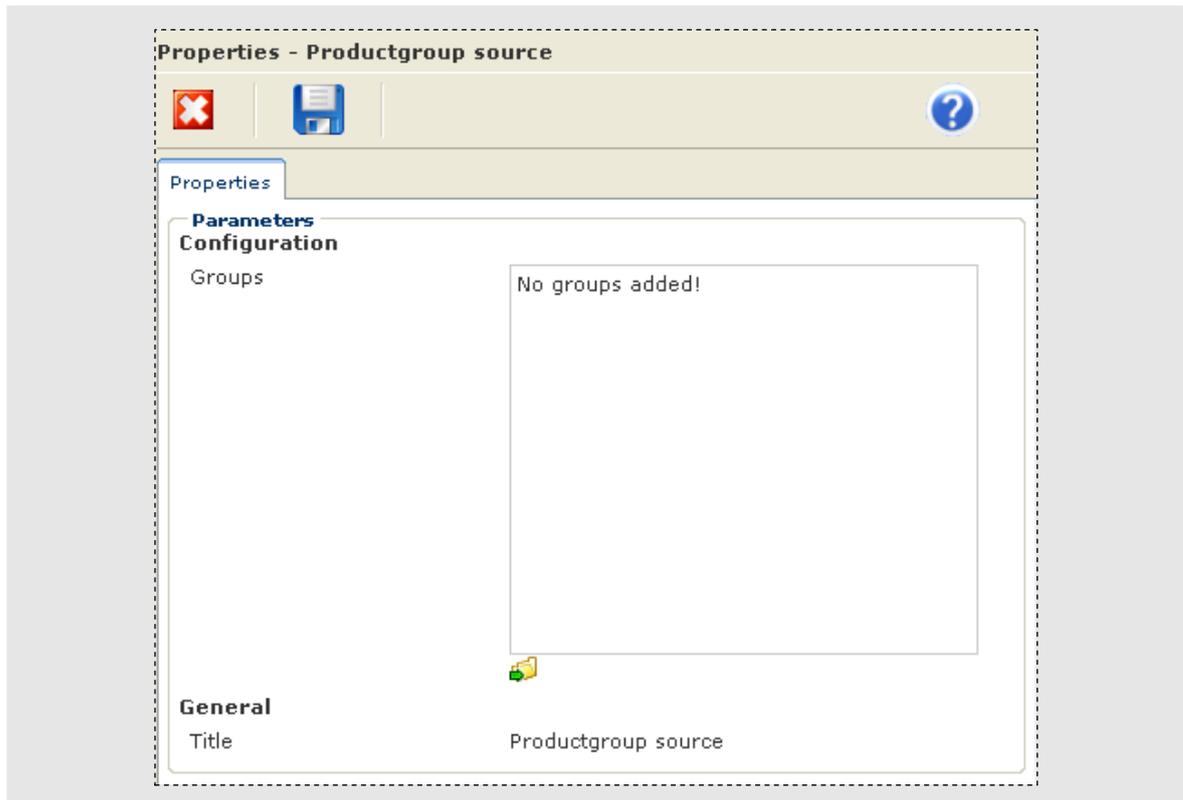


Figure 10. Product group source properties.

Configuration

- **Groups**

Click the *Add groups* icon to choose one or more product groups from the group navigation.

Product source

The activity allows you to make a reference to a product group in Dynamicweb eCommerce.

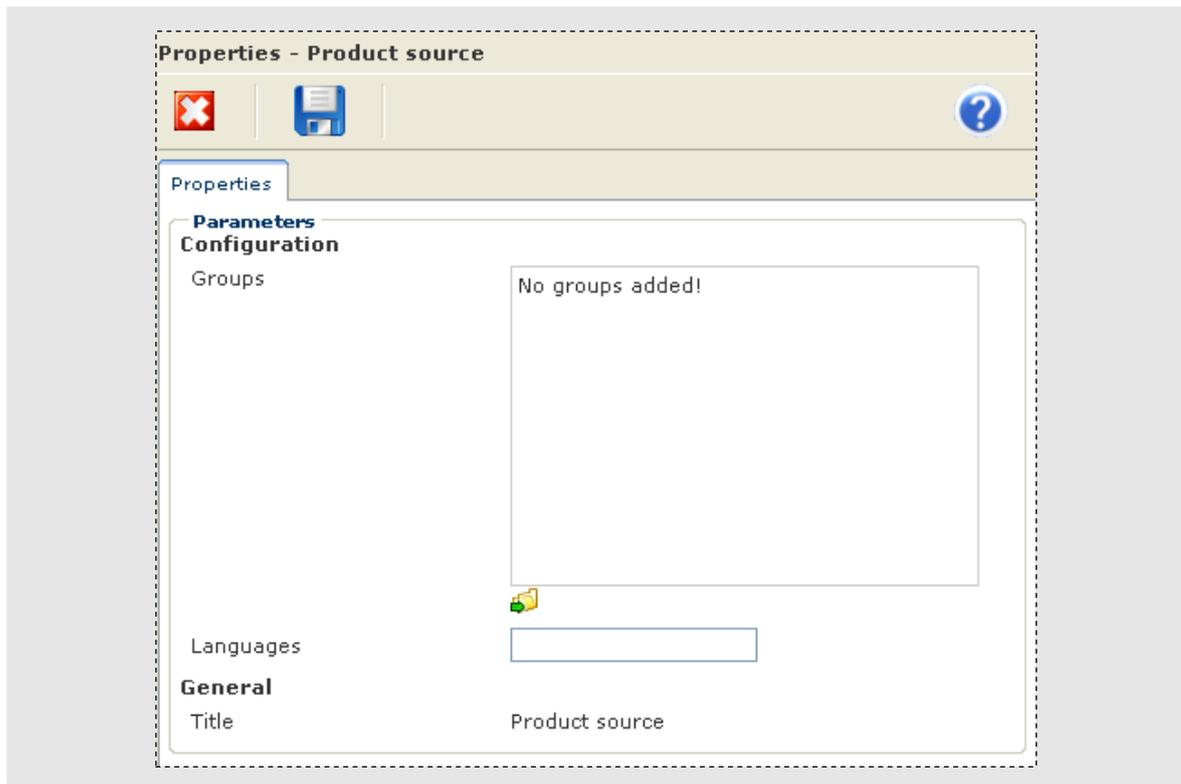


Figure 11. Product source properties.

Configuration

- **Groups**
Click the *Add groups* icon to choose one or more product groups from the group navigation.
- **Languages**
Use the *Languages* field to restrict the data transfer to products in a certain language, e.g. Danish or English products.

1.3.3 Transitions

XSL transform

The activity allows you to transform XML data from one structure to another. The activity can be placed after any file source in the pipeline. You can either add a reference to an XSLT file in the *File Manager* or enter the XSLT in a separate text field.

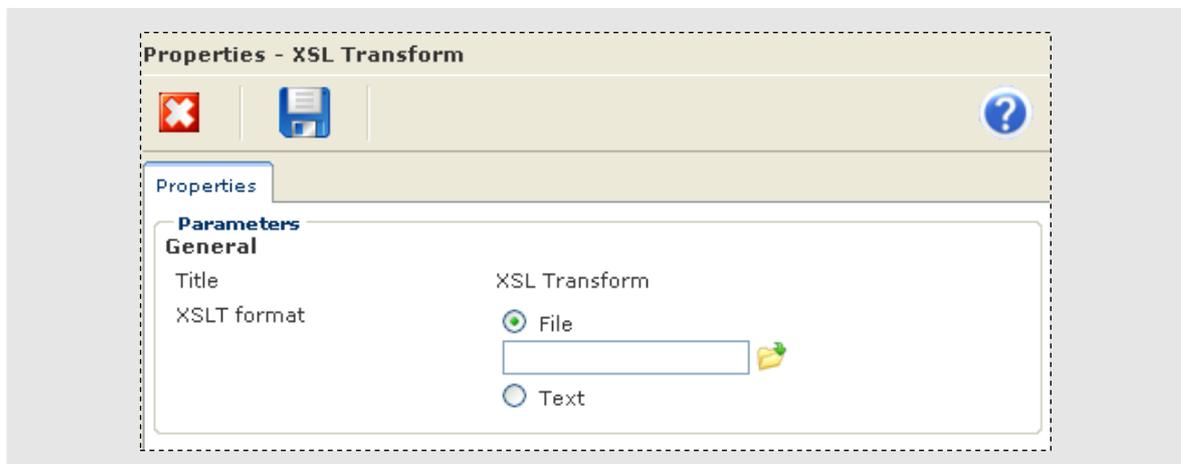


Figure 12. XSL transform properties.

General

- **File**
This option lets you choose an XSLT file from the *File Manager*.
- **Text**
Select this option to enter XSLT code directly in a separate text field.

1.3.4 Adding your own activities

If you want to add your own activities to the *Activities pane*, you can use the Dynamicweb eCommerce *Extensibility API* to develop custom activities (See 3 *Activity - Code sample*).

1.4 Run pipeline

After a successful configuration of a pipeline, you have to run the pipeline in order to export or import the eCommerce data. You can run the pipeline by clicking *Run* in the toolbar. This displays a report showing the different information about the data transfer. Initially the transfer history is displayed and underneath you can see the start and end date for the transfer.



Figure 13. Data transfer report.

1.5 Reuse pipeline

If you want to reuse a pipeline configuration, i.e. use the same combination of activities, you can use the *Save as...* function in the toolbar. You can for instance change the file reference or the product source and then save the pipeline with a new name in the *File Manager*.

As mentioned earlier, the *Import/Export* module comes with four standard pipelines that are locked for further reconfiguration. Instead you can use these pipeline "templates" to save your own copy of e.g. a product export and/or make small adjustments to fit your specific needs. You can read more about the four standard pipelines included in the *Import/Export* module under 2 *Sample pipelines*.

2 SAMPLE PIPELINES

The four sample pipelines allow you to perform basic imports and exports of products and product groups. You can use these samples as guides when creating your own pipelines or just reuse the pipeline after making minor adjustments to the activity properties. In the following you can read about the four pipelines and how they are put together. You will also learn how to reconfigure the basic import/export pipelines so that they fit your specific needs.

2.1 Import product groups from XML file to database

The following pipeline imports product groups from an XML file to the Dynamicweb eCommerce database. The pipeline consists of an "XML file source" activity and a "Product group destination" activity in the mentioned order. In this section you can read about how the pipeline is configured (and you can follow the same steps when you want to make your own product group import.)

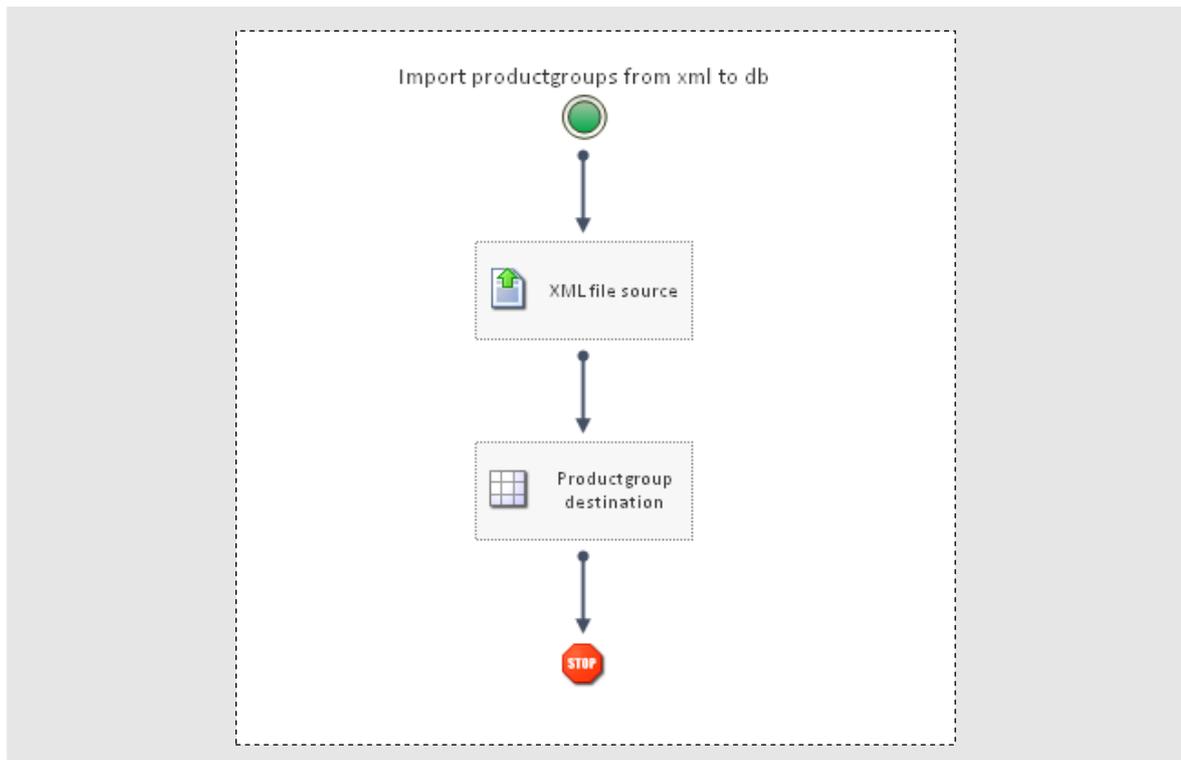


Figure 14. Graphical presentation of the import (product group) pipeline.



HOW TO CREATE A BASIC PRODUCT GROUP IMPORT

1. Open the pipeline editing interface.
2. Select an "XML file source" activity from the *Activities pane* and drag it to the *Workspace* just after the green icon.
3. Right-click the activity to open the *Properties* window.
4. Enter the root element of the XML file, e.g. *Groups*.
5. Enter the following XPath statement: `//Groups/Group`. (This filters the XML file so that only product groups are inserted)

- in the database.)
6. Choose the XML file containing the product groups from the *File manager*. (This could be XML data extracted from your financial system).
 7. Save the activity properties.
 8. Select a "Product group destination" activity from the *Activities pane* and place it after the XML source activity.
 9. Right-click the activity to open the *Properties* window.
 10. Click the "Choose folder" icon to open the group navigation.
 11. Select the shop that you want to insert the group into.
 12. Save the activity properties.

2.2 Export product groups from database to XML file

The following pipeline exports product groups from the Dynamicweb eCommerce database to an XML file in the *File Manager*. The pipeline consists of a "Product group source" activity and an "XML file destination" activity (in the mentioned order). In this section you can read about how the pipeline is configured.

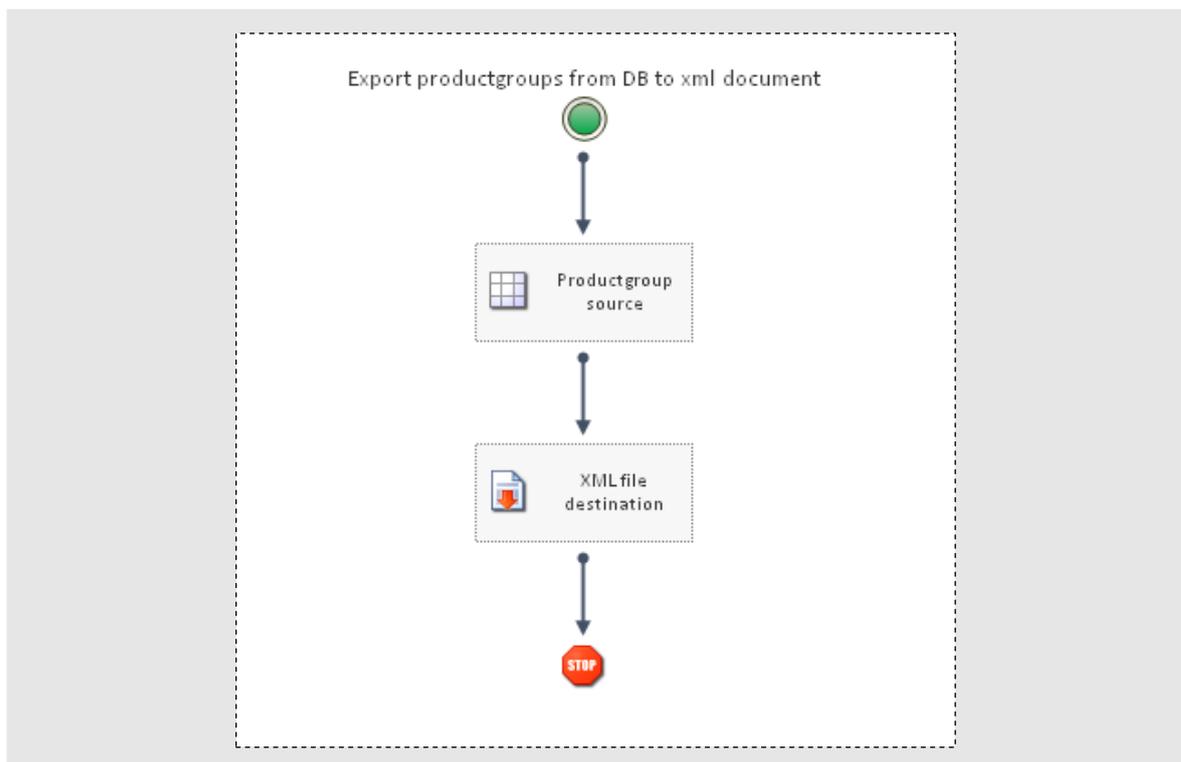


Figure 15. Graphical presentation of the export (product group) pipeline.



HOW TO CREATE A BASIC PRODUCT GROUP EXPORT

1. Open the pipeline editing interface.
2. Select a "Product group source" activity from the *Activities pane* and drag it to the *Workspace* just after the green icon.
3. Right-click the "Product group source" activity to open the *Properties* window.
4. Choose one or more groups to export data from.

5. Save the activity properties.
6. Select a "XML file destination" activity from the *Activities pane* and place it after the XML source activity.
7. Right-click the "XML file destination" activity.
8. Choose the folder where the destination XML file is placed.
9. Enter the XML file name, e.g. `ProductGroups.xml`.
10. Enter the root element of the XML file, e.g. `Group`.
11. Save the activity properties.

2.3 Import products from XML file to database

The following pipeline imports product groups from an XML file in the *File Manager* to the Dynamicweb eCommerce database. The pipeline consists of a "XML file source" activity and a "Product destination" activity (in the mentioned order). In the following you can read about how the pipeline is configured.

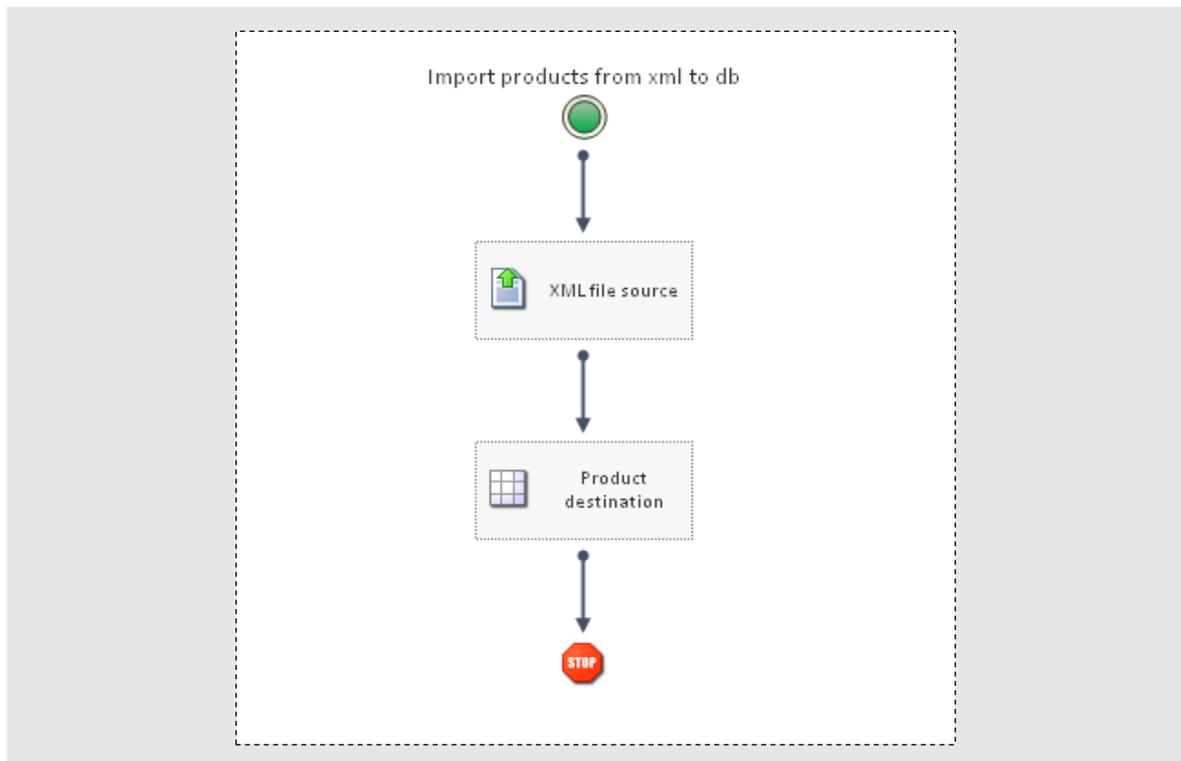


Figure 16. Graphical presentation of the import (product) pipeline.



HOW TO CREATE A BASIC PRODUCT IMPORT

1. Open the pipeline editing interface.
2. Select an "XML file source" activity from the *Activities pane* and drag it to the *Workspace* just after the green icon.
3. Right-click the "XML file source" activity to open the *Properties* window.
4. Enter the root element of the XML file, e.g. `Products`.
5. Enter an XPath statement, e.g. `//Products/Product`. (The proposed XPath statement filters the XML file so that only

- the actual products are inserted in the database.)
6. Choose the XML file containing the products from the *File manager*. (This could be XML data extracted from your financial system).
 7. Save the activity properties.
 8. Select a "Product destination" activity from the *Activities pane* and place it after the XML source activity.
 9. Right-click the "Product destination" activity to open the *Properties* window.
 10. Click the "Choose folder" icon to open the group navigation.
 11. Select the product group that you want to insert the products into.
 12. Save the activity properties.

2.4 Export products from database to XML file

The following pipeline exports products from the Dynamicweb eCommerce database to an XML file in the *File Manager*. The pipeline consists of a "Product source" activity and an "XML file destination" activity (in the mentioned order). In the following you can read about how the pipeline is configured.

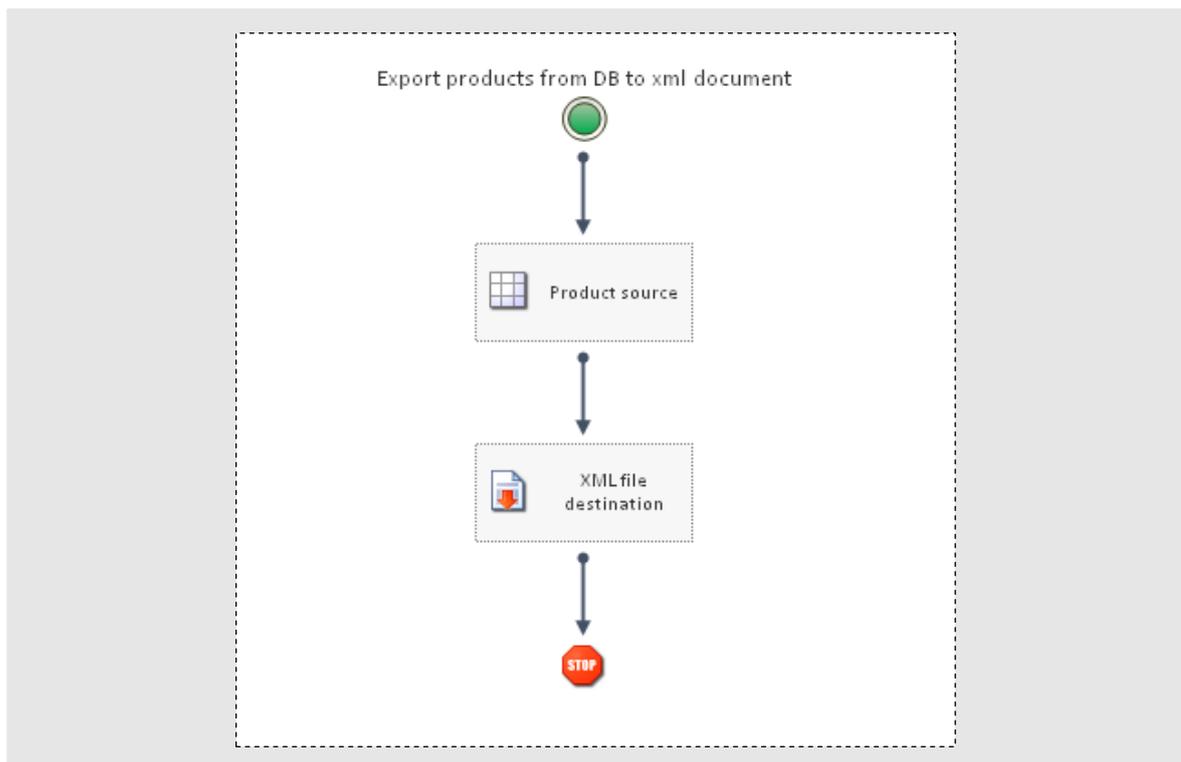


Figure 17. Graphical presentation of the export (product) pipeline.



HOW TO CREATE A BASIC PRODUCT EXPORT

1. Open the pipeline editing interface.
2. Select a "Product source" activity from the *Activities pane* and drag it to the *Workspace* just after the green icon.
3. Right-click the "Product source" activity to open the *Properties* window.

4. Choose one or more groups to export data from.
5. Save the activity properties.
6. Select an "XML file destination" activity from the *Activities pane* and place it after the XML source activity.
7. Right-click the "XML file destination" activity.
8. Choose the folder where the destination XML file is placed.
9. Enter the XML file name, e.g. `Products.xml`.
10. Enter the root element of the XML file, e.g. `Products`.
11. Save the activity properties.

3 ACTIVITY - CODE SAMPLE

Through the Extensibility API you can create properties for activities that are treated as settings for the activity when designing pipelines. By right-clicking on an activity you can see a window that contains a form that is rendered according to the set properties. This reduces the work considerably because no UI coding is needed, and you can concentrate on the functional code.

3.1 Datatype conditions.

An activity defines what type of data it inputs and what type of data it outputs. It does this through an *Attribute* called `AddInDataConstrain` that then contains the data conditions.

- "*", "XML" would for instance mean that it can input anything but always outputs XML.
- "", "XML" would mean that it has no input but returns XML.



What are the limitations?

The limits are your imagination. You can retrieve or send data from a local file, a remote server, a database, a web service or other data source/destinations.

3.2 How do I decide what my activity should do?

You do not need to load any settings as the *Extensibility API* should already have provided the activity properties with its correct values according to settings given when designing the pipeline (see 1.3 Activities.)

To achieve correct functionality there are several functions and properties that you must override:

1. The `Open` function that is run only when the activity is initialized first and allows you to do some preparation, like opening database connections or open a stream reader to a text file.
2. The `HasData` property that tells the next activity if your activity has more data to contribute to the pipeline.
3. The `Process` function which is run if the `HasData` returns true. This function would then contain the main code of your activity.
4. The `Close` function which is invoked as soon as the first activity in the pipeline has no more data.

These four are what most activities need to override in order to function correct. The `Open` and `Close` functions are optional and only needed in cases where something has to be invoked or disposed.

3.3 Code sample

This code sample shows how to make your own activity which you can use in the configuration of pipelines.

```
Imports System
Imports System.IO
Imports System.Collections.Generic
Imports System.Text

Imports Dynamicweb.Extensibility
Imports Dynamicweb.Extensibility.Editors

Namespace Integration

    < _
    AddInName("Textfile Reader"), _
    AddInGroup("Files"), _
    AddInImage("image.png"), _
    AddInDescription("Reads out of a textfile line by line."), _
    AddInActive(False), _
    AddInDataConstrain("", "XML") _
    > _
    Public Class TextFileReaderActivity
        Inherits Activity

        Private reader As StreamReader

        Private _filename As String = ""

        <
        AddInParameter("File"), _
        AddInParameterEditor(GetType(Editors.FileSelectEditor), "") _
        > _
        Public Property Filename() As String
            Get
                Return _filename
            End Get
            Set(ByVal Value As String)
                _filename = Value
            End Set
        End Property

        Public Overrides ReadOnly Property HasData() As Boolean
            Get
                Return Not reader.EndOfStream
            End Get
        End Property

        Public Overrides Sub Open()
            Dim path As String =
HttpContext.Current.Server.MapPath("\Files") & Filename
            Dim fi As New FileInfo(path)
            If fi.Exists Then
                reader = New StreamReader(Filename)
            Else
                Throw New Exception("File not found")
            End If
        End Sub

        Public Overrides Sub Close()
            reader.Close()
            reader.Dispose()
        End Sub

        Public Overrides Sub Process()
```

```
        Output = reader.ReadLine()  
    End Sub  
  
End Class  
End Namespace
```