Import Wizard Version 9

New to Import Wizard? See **Quick Start** (learn to use Import Wizard in 15 minutes)

Upgrading from a previous version? See **What's New**.

Introduction

The apparently simple task of importing a file into a database or spreadsheet can quickly get complex and confusing. Import Wizard has, thanks to the feedback from our over 2,000 customers, evolved into sophisticated tool for solving import problems.

Import Wizard is sold on a **Try-Before-You-Buy** basis. The unregistrered trial version is fully functional and identical to the registered version. The only exception is that the trial version imports up to 30 records into the output database, you can use the Preview function to see all records without actually storing them in the output database. This allows you to fully test the software: if your import works for 30 records in the trial version, we guarantee that it will work for 100 million records in the registered version.

Import Wizard can be used as **standalone program** and as **Add-In for Excel and Access** versions 2007, 2003, XP(2002), and 2000. The standalone program is started from the Windows Start menu. The Add-Ins are started from the Tools menu within Excel or Access, see screenshots below.

Benchmark: The screenshot below was taken after an import of a 3.5 GB source file with 12 million records. Import Wizard imported the file with a speed of over 1,000,000 records per minute.

Requirements: Microsoft .NET framework 2.0 or later. Import Wizard runs on 32 and 64 bit Windows 7, Vista, XP, Server 2003, 2000, NT4, ME and 98.

Imports from: complex fixed position text files, report files, printer spool files, delimited text files, XML files, HTML files, and Excel files.

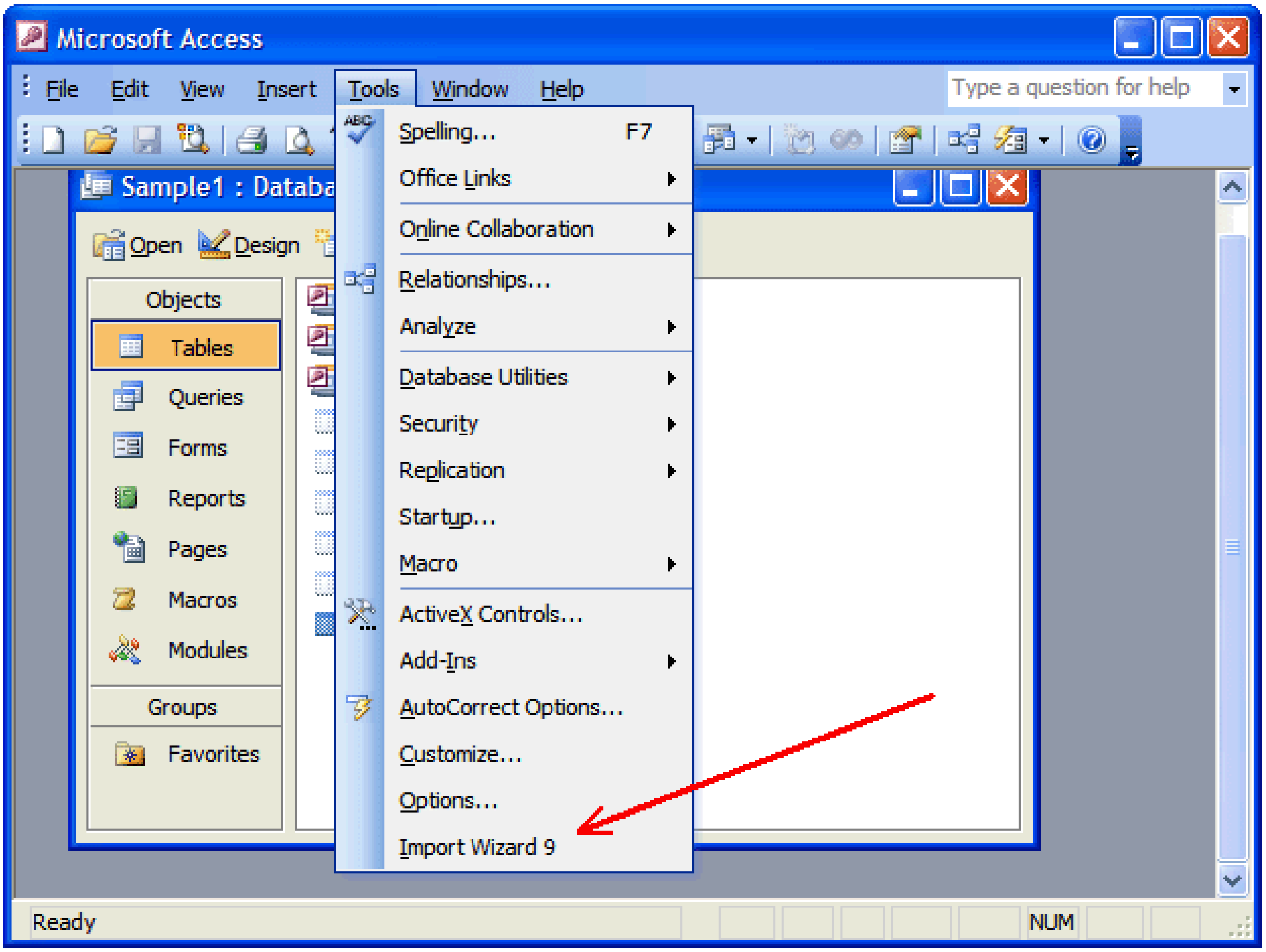
Outputs to: Access tables (does not require Access to be installed), Excel, SQL Server, MSDE, Oracle, MySQL, dBase, ODBC Databases, delimited files, XML files, HTML files and SQL script files.

Support: If you encounter any problems during your evaluation or after your purchase, please do not hesitate to contact us by email: impwiz@beside.com. In your email attach the .iwm file you are using and a small sample source file so that we can help you efficiently.

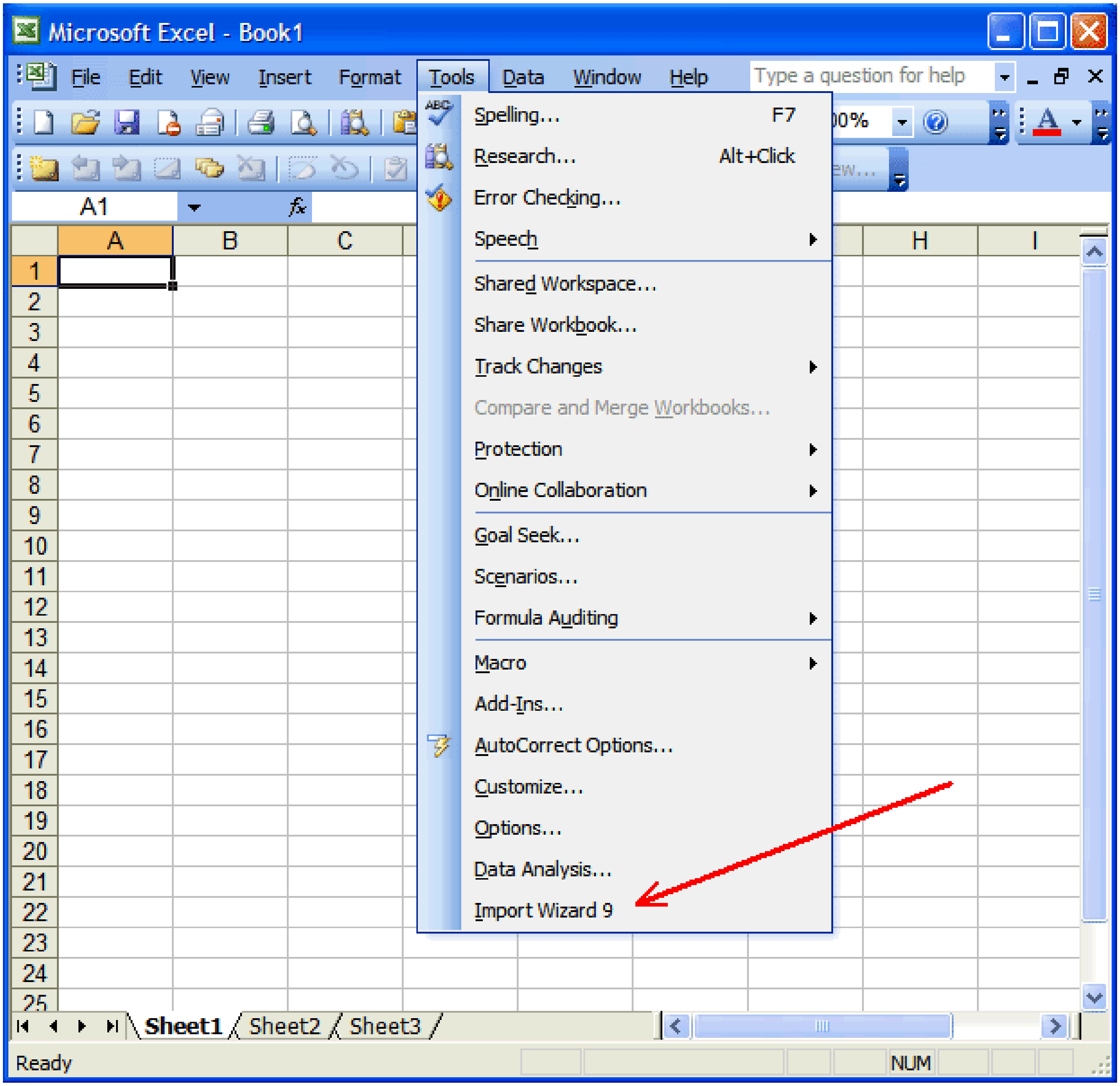
Free updates: Software updates for the same main version number as you purchased (9.x.x) are free of charge and can be downloaded from our website: www.beside.com

SDK: For developers the Import Wizard Software Developer Kit is available. With the SDK you can embed Import Wizard into your own application. Please visit www.beside.com for more information, licensing and downloads.

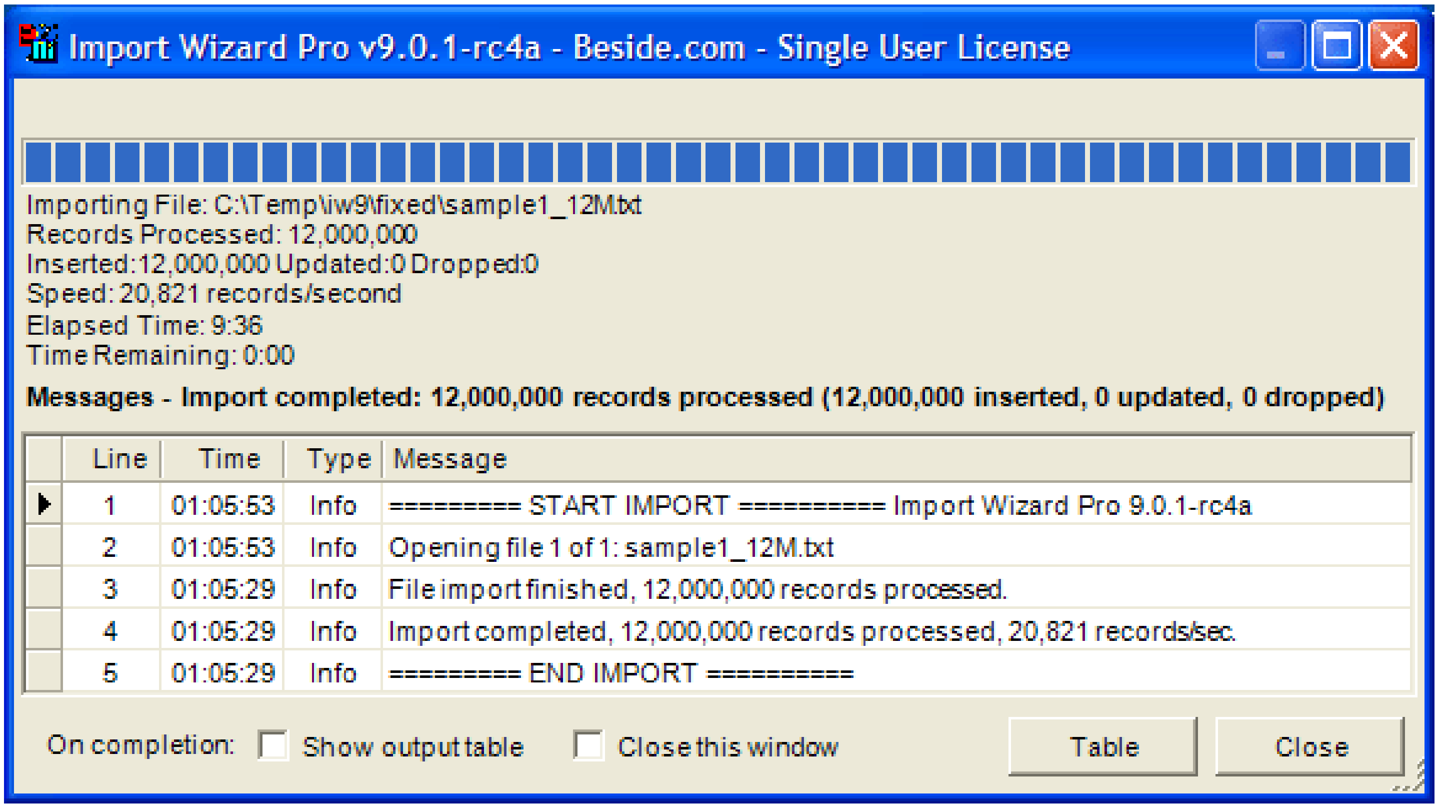
Screenshot: Starting Access Add-In



Screenshot: Starting Excel Add-In



Screenshot: Benchmark


Import Wizard Pro v9.0.1-rc4a - Beside.com - Single User License

Importing File: C:\Temp\iw9\fixed\sample1_12M.txt
Records Processed: 12,000,000
Inserted:12,000,000 Updated:0 Dropped:0
Speed: 20,821 records/second
Elapsed Time: 9:36
Time Remaining: 0:00

Messages - Import completed: 12,000,000 records processed (12,000,000 inserted, 0 updated, 0 dropped)

Line	Time	Type	Message
1	01:05:53	Info	========= START IMPORT ========== Import Wizard Pro 9.0.1-rc4a
2	01:05:53	Info	Opening file 1 of 1: sample1_12M.txt
3	01:05:29	Info	File import finished, 12,000,000 records processed.
4	01:05:29	Info	Import completed, 12,000,000 records processed, 20,821 records/sec.
5	01:05:29	Info	========= END IMPORT ==========

On completion: [ ] Show output table [ ] Close this window

Benchmark Details

- Source file size was 3.56 Gigabytes, the source file was created from 4 million copies of Sample1.txt
- Output to comma delimited file without text qualifier, output file size 433 Megabytes.
- Speed: 12 million records imported in 9 minutes 36 seconds, 20,821 records/second, 1,250,000 records/minute.
- File imported with sample1.iwm from the samples directory with the 6 fields set to Text/Trim.
- Import Wizard's memory usage was constant at 27MB during whole import as reported by Task Manager.
- Benchmark ran on a Windows XP machine with a 1.5 GHz Pentium M and 768 MB Ram

General Information

User's Manual

Reference Manual

Quick Start Tutorial

Introduction

In this turorial a Fixed Position file is imported, but the principles also apply to Delimited, HTML and Excel imports. XML imports work differently, see Importing XML Files.

Import Wizard uses so-called Import Models that define how a file is imported. These models are stored as regular files, with an **.iwm** extension. Models can be opened, saved and modified with the buttons on the main Import Wizard window.

The following example explains the most commonly used import features, advanced features are explained in the Reference section of this help file.

In the "C:\Program Files\Import Wizard 9\Samples" subdirectory you find the sample source files to be imported (*.txt), as well as the pre-build model files (*.iwm) for these source files.

The Task

In this example we import the source file "sample1.txt", this file has the following content:

```
==========================================================
Employee Hours Report                           Page:  1
Company: XYZ Company                    Date Printed:02/02/99
==========================================================
Employee           Date     Regular Hours/    Other Hours/
                            Overtime              Code
----------------------------------------------------------
Doe, John        01/23/99     23.3                 1.6
                              15.6                  AC
----------------------------------------------------------
Wacks, Gene      02/01/99      1.6                 0.0
                               0.0
----------------------------------------------------------
Peters, Kate     01/30/99     40.3                 3.4
                               1.1                  VA
----------------------------------------------------------
```

Our goal is to create the following table from this source file:

Employee	Date	Regular Hours	Other Hours	Overtime Hours	Code
Doe, John	1/23/1999	23.3	1.6	15.6	AC
Wacks, Gene	2/1/1999	1.6	0	0	
Peters, Kate	1/30/1999	40.3	3.4	1.1	VA

The Solution

Using standard fixed width text import program you will have a number of difficulties to overcome before you have the desired result. First you will have to get rid of the header lines and the dividing lines between the records, then you have to figure out a way to combine two lines per record into a single record. Import Wizard handles these tasks for you.

Follow these instructions to build a model for this import:

1. Create New Model

Start Import Wizard from the Windows Start menu: *Start->Programs->Import Wizard*. The Access and Excel Add-in versions are started from within Access/Excel: open any Access database or Excel spreadsheet, then select *Tools->Import Wizard* from the Access/Excel menu.

Press the New button on the Main Window to create a new model.

2. Select Source File

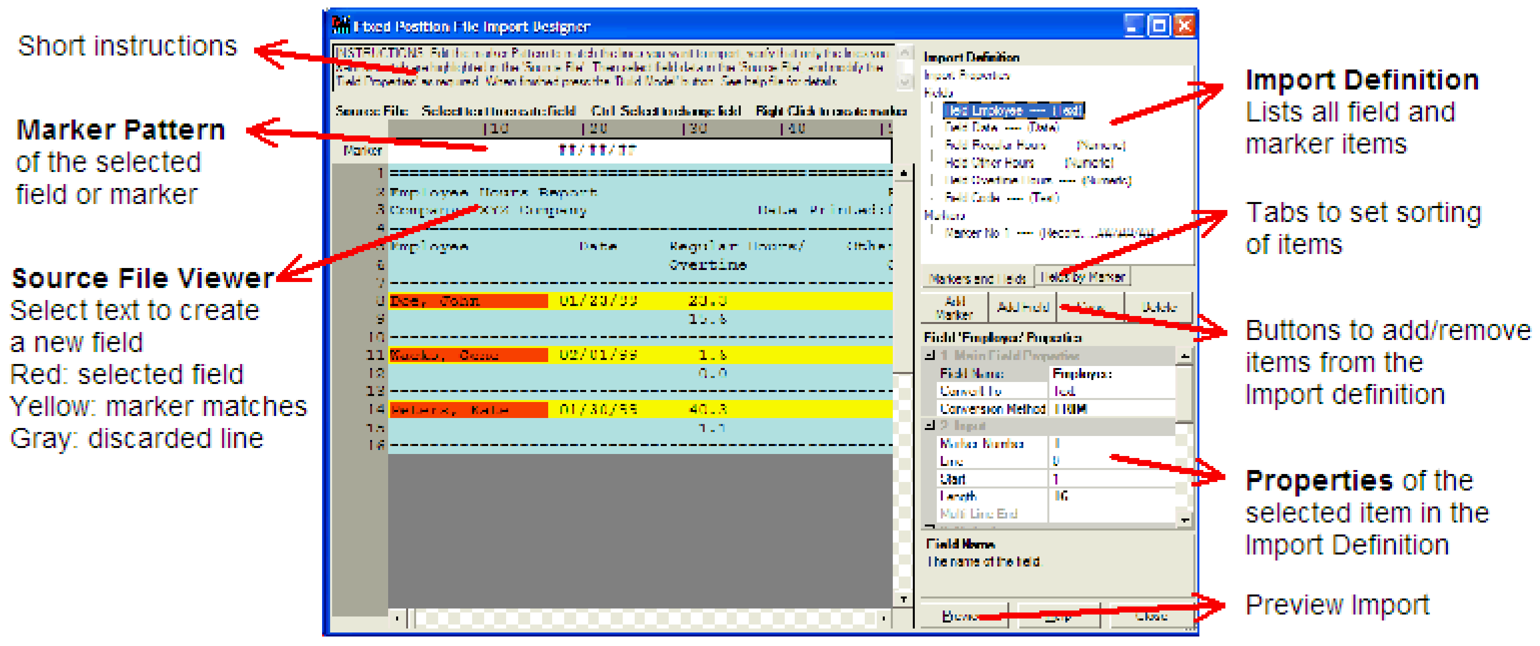
The Source File Dialog is now visible; select the file "sample1.txt" from the "Samples" subdirectory.

3. Select Import Type

In the Import Type Dialog select Fixed, and click the Next> button.

4. Design Window

You now see the Design Window. This window appears to be very complex at first sight, but once you understand the layout, this window allows you to define import models conveniently and quickly. The design window layout is as follows:


Short instructions
Marker Pattern of the selected field or marker
Source File Viewer Select text to create a new field Red: selected field Yellow: marker matches Gray: discarded line
Fixed Position File Import Designer
Import Definition Lists all field and marker items
Tabs to set sorting of items
Buttons to add/remove items from the Import definition
Properties of the selected item in the Import Definition
Preview Import

5. Markers

Markers are a central feature of Import Wizard. Markers allow you to mark (select) lines in the source file and to perform operations on these marked lines. You control which lines are marked by editing the **Marker Pattern** input box. A line in the source file is marked when the marker pattern matches the line.

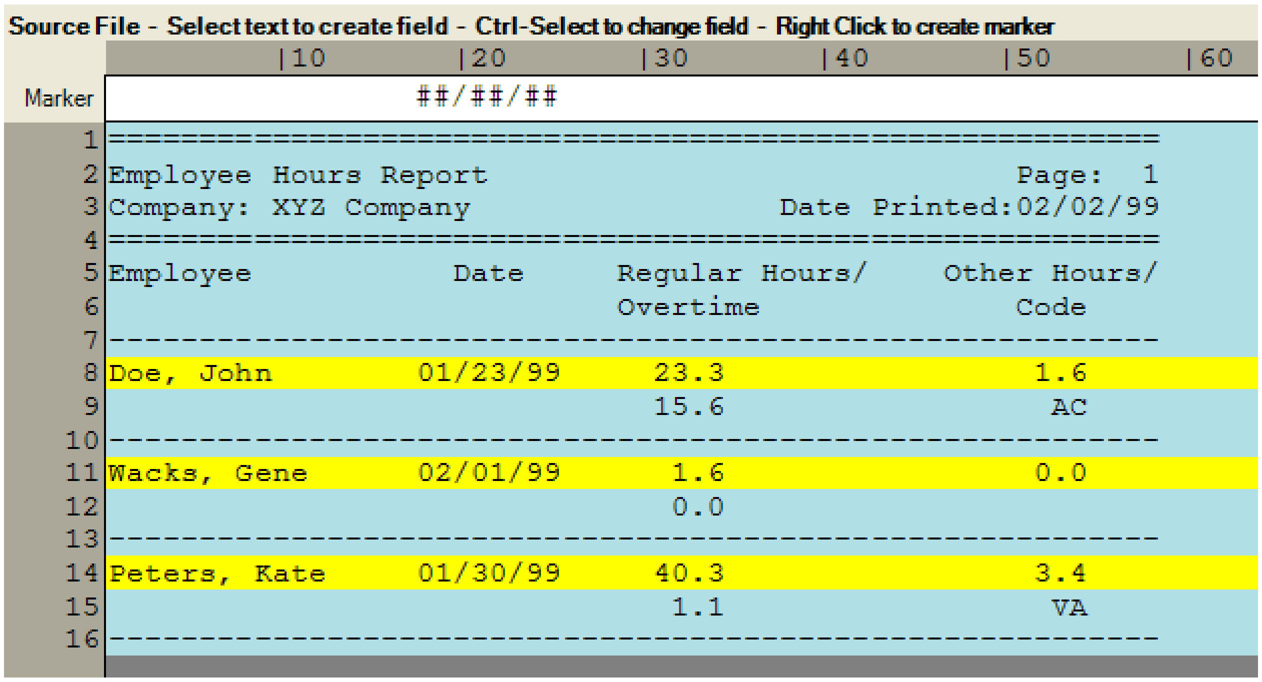
A marker pattern works similar to an asterix (*) in filenames. To list all the filenames that have an .asc extension you would enter ***.asc**. Here, the "*" is a wildcard character and stands for none or more characters. The other characters (".", "a", "s", and "c") have no special meaning and simply match the same character.

The only wildcard characters for filenames are the asterix (*) and the question mark (?), all other characters have no special meaning. For Marker Patterns a far more extensive set of wildcard characters is available, see following table:

Wildcard Character	Description
space	Matches any character.
?	Matches any character except space.
@	Matches an alpha character ("a" to "z" and "A" to "Z").
#	Matches a digit character ("0" to "9")
_	Matches a space character (" ").
other	Any character that is not a wildcard will match the same character (case sensitive).

In this example we want the marker to match the lines with the employee names, as these lines correspond with the records we want to import. Our task is to think of a Marker Pattern that matches only these lines in the file. Looking at the file, you see that the lines we want to match all contain a date and that this date does not appear at the same position in anywhere else in the file. Therefor this date is ideal as basis for our Marker Pattern.

A Marker Pattern for the date is **##/##/##**, in words: we look for lines with two digits (**##**), followed by a slash character (**/**), then two digits (**##**), a slash (**/**) and finally two digits (the last **##**). This pattern has to be entered in the Marker Pattern input box at the same character position as the dates we want to match. After entering the Marker Pattern the File Viewer looks like:

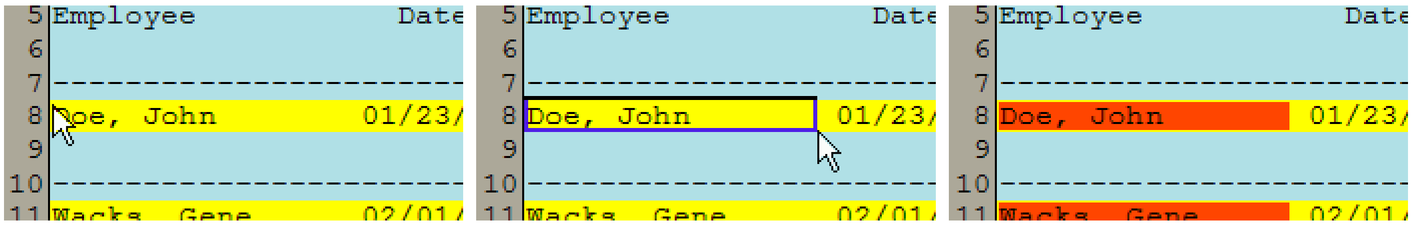


```
Source File - Select text to create field - Ctrl-Select to change field - Right Click to create marker
                 |10       |20       |30       |40       |50       |60
Marker                ##/##/##
==========================================================
Employee Hours Report                               Page:  1
Company: XYZ Company                      Date Printed:02/02/99
==========================================================
Employee             Date      Regular Hours/     Other Hours/
                               Overtime                Code
----------------------------------------------------------
Doe, John          01/23/99      23.3                   1.6
                                 15.6                    AC
----------------------------------------------------------
Wacks, Gene        02/01/99       1.6                   0.0
                                  0.0
----------------------------------------------------------
Peters, Kate       01/30/99      40.3                   3.4
                                  1.1                    VA
----------------------------------------------------------
```

For the marker to work correctly, the **##/##/##** has to be aligned with the dates in the file. This alignment is most easily done by scrolling the file until the line that needs to be matched is directly below the pattern input box. To further assist in entering the Marker Pattern the matched lines are highlighted as soon as you edit the Marker Pattern.

6. Fields

Next we create the fields. You can create new fields by selecting text in File window. Simply move the mouse pointer to the start of the text you want to import, press the left mouse button, drag the mouse to select the text you want to import into the field, then release the mouse button. The images below illustrate creating the field "Employee" by dragging the mousepointer over an employee name, here "Doe, John":



Make sure to include enough spaces after "Doe, John" to correctly import longer employee names. After you release the mouse button a new field is appended to the Model window, and the File windows highlights the field in red. Notice that in addition to "Doe, John" the other employee names are also highlighted. You can now change the Field Properties as desired, in our case we change the field name from "Field1" to "Employee".

Do the same with the "Date", "Regular Hours", "Overtime Hours", "Other Hours", and "Code" fields. If you make an error in dragging your mouse, you can delete the field by pressing the Delete button, then try again. Alternatively you can edit the field's Start, Length and Line properties.

7. Test your Import Model

Now that you defined the import model, press the "Import Preview" button to see the results. This will open a popup window, which contains the three records we wanted to import. If the result is different, you have to make the appropriate changes to your model and try again.

8. Finishing up

Once you verified that the model works correctly, press the "Ok" button. This brings you back to the Main Window and you see the model properties of the newly created model. You can modify these properties as required, for example you can enter a diffrent output table name. Save the model by pressing the "Save" button. Run the import by pressing the "Import" button.

If you have problems creating the model for this example; have a look at the pre-build model. Press the **Open** button and select the model file "sample1 Fixed.iwm" from the "Samples" directory. Then press the **Modify** button.

Further Reading

Example2 - Import with Headers
Other Examples
Importing Fixed Position Files
Importing Delimited Files
Importing HTML Files
Importing Excel Files
Importing XML Files
Marker Types

Import Example 2: Header Imports

See also: Import Example 1: Simple Import | Other Import Examples | The Main Import Wizard Window

In this example we will build a model for the "sample2.txt" report file. The file contains this report:

```
==========================================================
Product Sales Report                           Page:   1
Company: XYZ Company          For the month ended:09/30/98
==========================================================
Sales Person        Date         Qty          Price     Class
----------------------------------------------------------
            Product: Tape

Doe, John        09/23/98     1,120     11,225.60       AC
                 09/24/98    21,123    233,112.67       VA
                 09/27/98        23         32.22       CS
Wacks, Gene      09/01/98     3,766      3,455.55       AC
Peters, Kate     09/30/98     4,000      3,999.44       VA
                 09/02/98     1,653      6,775.55       OT

            Product: Stapler

Perrol, Barb     09/13/98     3,320     11,225.60       AC
Zonker, Pete     09/11/98       766        453.53       AC
                 09/20/98     1,455      2,543.43       VA
Perkins, Joe     09/12/98        23         23.55       OT
```

From this report you want to create a table that contains sales person, sales numbers, product and company, e.g. this table:

Company	Product	Sales Person	Date	Qty	Price	Class
XYZ Company	Tape	Doe, John	9/23/97	1120	11225.6	AC
XYZ Company	Tape	Doe, John	9/24/97	21123	233112.67	VA
XYZ Company	Tape	Doe, John	9/27/97	23	32.22	CS
XYZ Company	Tape	Wacks, Gene	9/1/97	3766	3455.55	AC
XYZ Company	Tape	Peters, Kate	9/30/97	4000	3999.44	VA
XYZ Company	Tape	Peters, Kate	9/2/97	1653	6775.55	OT
XYZ Company	Stapler	Perrol, Barb	9/13/97	3320	11225.6	AC
XYZ Company	Stapler	Zonker, Pete	9/11/97	766	453.53	AC
XYZ Company	Stapler	Zonker, Pete	9/20/97	1455	2543.43	VA
XYZ Company	Stapler	Perkins, Joe	9/12/97	23	23.55	OT

This import requires a model with three markers. The first marker is used to import the record information of Sales Person, Date, Qty, Price and Class fields. The second marker for the header Product, and the third to import the Company header. Follow these steps to create the model:

1.Select New from the File menu.

2.File to Import

In the file dialog select the report file “sample2.txt” from the Samples directory.

3.Select the type of file to import

Select Fixed, and click the Next> button.

4.Define Marker (first marker)

Edit the marker pattern and change it to:

" ##/##/##"

Make sure the “##/##/##” aligns with the dates in the file.

As you enter the marker, you will notice that lines the file are highlighted in yellow. These are the lines that match the marker pattern. Make sure that all the 10 detail lines are highlighted.

Note that the Marker Type is set to “Record”, this indicates that the matching this marker will create a new record (or row) in the import table each time it matches a line in the file.

5.Define Import Fields: (for the first marker)

Select text in the File window to define new fields for "Sales Person", "Date", "Qty", "Price", and "Class".

Make sure that the “Repeat” is set to Yes for the "Sales Person" field. This will repeat the sales person’s name for the records were the name is missing. E.g. "Doe, John", whose name occurs only in the first record, will be repeated in records 2 and 3 even though his name was not printed on the corresponding report lines.

For the “Qty” and “Price” fields select “Numeric” as field type.

6.Define New Marker Pattern: (second marker)

Click one of the lines with the product name in it, and press Yes to create a new marker. Alternatively, you can press the “New Marker” button to create a new marker.

Modify the marker pattern to:

" Product:"

Test that the pattern matches all the product lines by pressing the Refresh button.

For this marker pattern set the Record Type is to “Header”. This indicates that upon matching this marker Import Wizard will not create a new record (or row) in the import table. When a match is found for a header marker only the fields associated with the marker are updated, in this case the “Product” field.

7.Define Import Fields: (for the second marker)

Select text in the File window to define the “Product” field. In the Field Properties window set “Repeat” to Yes to repeat the entry for every record.

8.Define New Marker Pattern: (third marker)

Create a new marker for the company field by clicking the line with the company name in it..

Modify the marker pattern to:

"Company:"

Test that the marker matches the company line and set the Marker Type to “Header”.

9.Define Import Fields: (for the third marker)

Select text in the File window to define "Company" field, again with “Repeat” set to Yes, then click the “Ok” button.

Other Import Examples

See Also: Import Example 1: Simple Import | Import Example 2: Header Imports | The Main Import Wizard Window

The examples are stored in the "Samples" subdirectory of the directory where you installed Import Wizard, "C:\Program Files\Import Wizard 8\Samples" by default.

Example 3: Address Import

This example shows how to solve problems with optional fields, i.e. fields that are not present in each record. By defining separate sections and markers for each field it is possible to correctly import such files.

Example 4: Delimited Text File

This example shows how to selectively import fields from a delimited text file. The file used in this example is the log file of a web server.

Example 5: Footer Import

Sometimes the required information is found in footers below the actual record. This example shows how to use footers in an inventory report.

Example 6: Use of Delimited Field No and Formula properties

This example uses the SAMPLE1.TXT file to create a table with the first/last name of the employee together with the total number of hours.

Delimited Field No / Delimiter properties

The employee name is in the source file in the format Last, First (for example: “Doe, John”) From this data the first and last name are extracted using the “Delim Field No” property. To extract the last name “Delim Field No” is set to 1 and “Delimiter” to {COMMA}, this will extract the first field from the left. For the first name “Delim Field No” is set to –1, thus extracting the first field from the right. Note that “Delim Field No” set to 2 would have worked as well of course.

Formula properties

The first and last name obtained using the method described above is combined into a field named “FirstLast Name” using the formula: First_Name & “ “ & Last_Name

Note the underscore (“_”) in the fieldnames in the formula. As spaces and other special characters are not allowed in formula names these have been replaced with an underscore. The name of a field for use in formulas is given by the Formula Field property.

The “Hours” field has this formula: Regular_Hours+Other_Hours+Overtime_Hours, thus adding up all the hours. Note that the “Regular Hours”, “Other Hours”, and “Overtime Hours” fields are present in the model but not output to the destination table. This is because the Skip property is checked for these fields.

Example 7: Multi Line Fields

This example demonstrates the use of the MultiLine Field Type property.

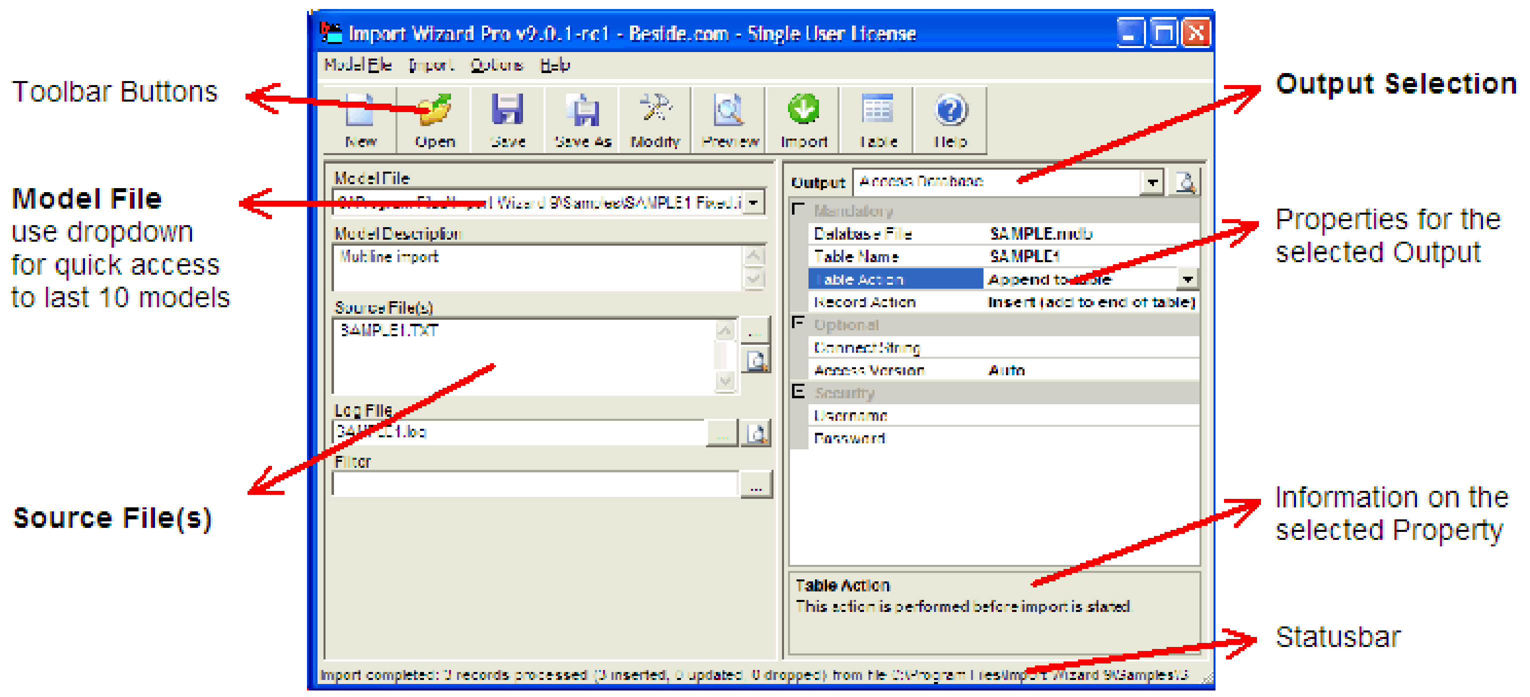
Example 8: Vertical Report

Sample report with records organized in columns instead of rows.

Example 9: Chinese Characters

Import of a file with Chinese characters. The Chinese characters are treated as double width characters in order to align the columns correctly. See the 'Chinese Chars' property in Fixed Import for details.

Main Window



Import Wizard Pro v9.0.1-rc1 - Beside.com - Single User License

Model File: C:\Program Files\Import Wizard 9\Samples\SAMPLE1 Fixed.i

Model Description: Multline import

Source File(s): SAMPLE1.TXT

Log File: SAMPLE1.log

Filter:

Output: Access Database

Mandatory
- Database File: SAMPLE.mdb
- Table Name: SAMPLE1
- Table Action: Append to table
- Record Action: Insert (add to end of table)

Optional
- Connect String:
- Access Version: Auto

Security
- Username:
- Password:

Table Action
This action is performed before import is started

Import completed: 3 records processed (3 inserted, 0 updated, 0 dropped) from file C:\Program Files\Import Wizard 9\Samples\S

The main window is the control center for the import. It allows direct access to the most often used properties: the source (input) files and the Output settings. Pressing the Modify toolbar button shows the design window for Fixed Position, Delimited, HTML, Excel, or XML import.

Model File

Read only property, the file name of the current model definition. Selecting one of the filenames from the dropdown list will open that model file.

All relative directory paths are relative to the path of the model file. For example: the log file "log\mdl.log" for model "c:\models\mdl.iwm" will be created in the directory "c:\models\log\" and source file "..\src.txt" is located in "c:\".

Model Description

You can use this property to add your notes to the model.

Source File(s)

The file(s) you want to import. You can use the **?** and ***** wildcards to select multiple files. It is also possible to supply multiple file names by entering each file name on a separate line. If no directory is specified, the previously entered directory will be used. Pressing the **...** button opens the browse-for-file(s) dialog. Pressing the Preview button will open the first source file in Notepad, or the editor given under Options.

The following example imports all files from the c:\dir1 directory and file1.txt and file2.txt from the c:\dir2 directory:

```
c:\dir1\*.*
c:\dir2\file1.txt
file2.txt
```

When a line starts with "-r " then files will be imported recursively from the specified directory and all its subdirectories. The following example imports html files from dir1 and all subdirectories of dir1:

```
-r c:\dir1\*.html
```

Log File

The filename where the import log is stored. The import log contains all import warnings, regardless of the “Show Warnings” setting. Leave the empty to skip log file creation. Pressing the ... button will open the Browse file dialog, pressing the preview button will open the log file in the text editor.

Filter

Leave blank to import all records, or enter a formula. The formula is evaluated just before a record is stored in the database. If the Filter formula evaluates to True (or a value not equal to 0) the record is stored in the database, otherwise the record is ignored. Pressing the ... button shows the Filter Builder dialog. See Formulas for details.

Ignore Type Conversion Warnings

When checked type conversion warning messages are not reported nor logged to the log file. Type conversion warnings occur when Import Wizard can not convert the loaded data into the desired type, for example converting '99/99/9999' to a date generates a type conversion warning.

Automation

Press the ... button to access the automation settings. The following settings are available:

Property	**Description**
On File Import OK	The selected action is performed when a file has been imported without any warning messages. The following actions are available: **Move File To** the file is moved to the entered directory. The directory name can be absolute (starting with \ or a drive specification such as c:), or relative to the model file path. For example: 'ImportOK' moves the imported files to *ModelFilePath\ImportOK\, and '\\computername\share\path' moves the imported file to the specified network path.* ***Delete File** delete the file.* ***Execute Commandline** executes the entered command line.*
On File Import with Warnings	The selected action is performed when a file has been imported and at least one warning occured while importing the file. The same actions as with 'On File Import OK' are available.
On Import Completed OK Execute	This command line is executed when the import completed without any warning messages.
On Completed with Warnings Execute	This command line is executed when the import completed with warning messages.
On Import Failed Execute	This commandline is executed when the import failed.

Note: To test your settings use 'Import Preview'. This logs the automation actions to the message list without actually executing the actions.

Output

This dropdown selects the type of output that is generated from the imported data. See Outputs for details on the specific properties for each output.

Toolbar Buttons

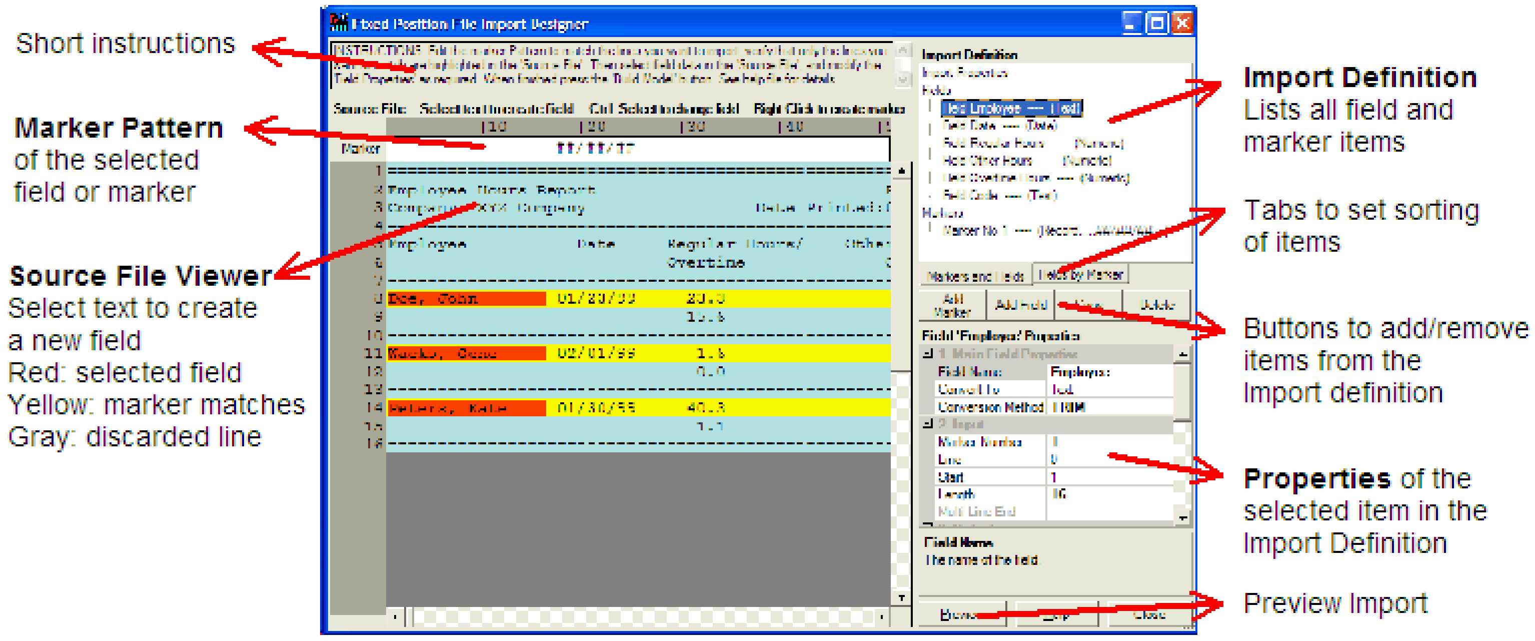
New Creates a new model.
Open Opens an existing model.
Save Saves the current model.
Save As Saves the current model under a different name.
Modify Opens the model designer of the current model.
Preview Runs the import without storing the data to the selected output.
Import Runs the import storing the data to the selected output.
Table View the imported table.
Help Show this help file.

Importing Fixed Position Files

Introduction

This help page contains reference information only, please read the **Tutorial** first.

Window Layout



The screen is split in three subwindows:

Model Definition Window gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties Window gives the property details for the selected marker or field in the Model window.

Source File Window shows the file content. Lines in the source file which match the selected marker are highlighted in yellow. If a field is selected then the field data is also highlighted in red. Discarded (ignored) lines are gray and any other lines are blue.

The File window supports two shortcuts. A new field is created when you select text in the File window (click and drag the mouse to select one or more characters). A new marker is created if you right click a line in the File window.

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Fields

To create a field: select a part of a line in the File window, this will setup the field with the currently selected marker and will try to guess (based on the selected text) the correct settings for the 'Convert To' and 'Data Type' properties. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source File

The following table only describes the Field Properties specific for Fixed imports, see **Field Properties** for a description of common Field Properties.

Field Property	Description
Marker Number	The marker number that will trigger the import of this field.
Start	The starting character position in the source file. The first position on a line is 1.
Length	The length in characters of the import field in the source file.
Line	The line offset of the import field relative to the line that matched the marker pattern. I.e. line 0 means the field is on the same line, -2 means the field is two lines above the matched line.

Markers

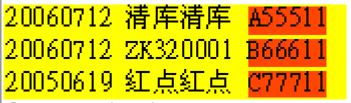
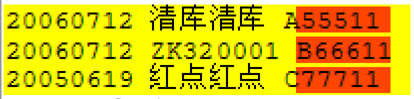
You can create new markers by pressing the **Add Marker** button, this will create a new blank (match all) marker. Alternatively, you can right click a line in the File window to create a new marker for the selected line, i.e. the selected line is copied into the marker pattern. See **Markers** for a detailed description of all Marker properties.

The marker pattern can be edited directly above File contents. Use the File vertical scroll bar to position the line you want to match directly under the pattern input box to help in aligning your pattern correctly.

The marker Pattern can contain multiple lines, a match is made when all the lines of the marker pattern match. To enter a multiline marker click the **v** button to the right of the marker input box.

Import Properties

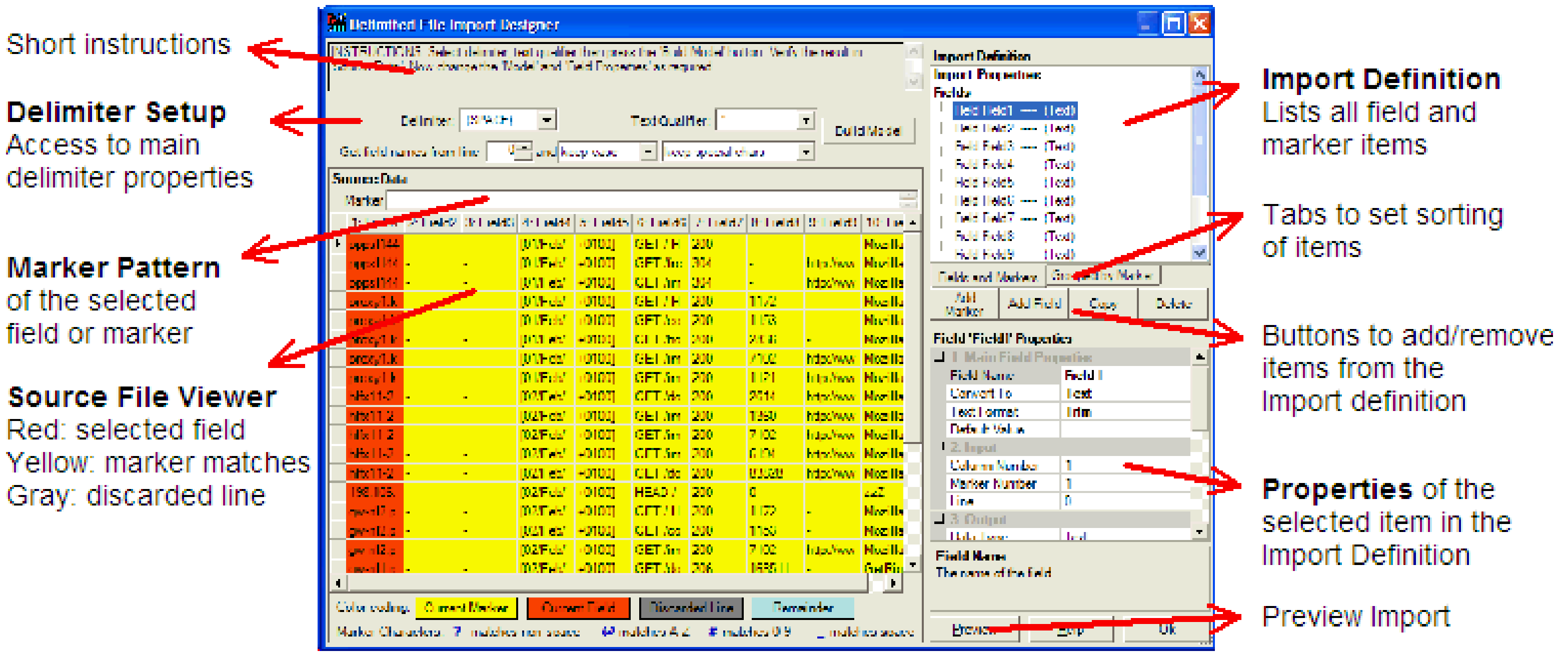
Selecting Import Properties in the Import Definition window allows you to set the following properties:

Property	Description
Year 2000 Cutoff	This is the first year for two digit year imports that will be treated as in the 20th century. This is set to 30 by default, i.e. 30-99 converts to 1930-1999, 00-29 to 2000-2029.
Code Page	Select the desired character set Code Page to read the source file. Import Wizard works internally with unicode (16bit) characters. If the required Code Page is not in the list the enter the Code Page number by hand. If the Code Page is invalid you will get a import warning upon importing.
New Line Delimiter	This parameter gives the character(s) that mark the end of a line. Default when left blank is to use {13}{10} or {10} which will import both Windows (CR+LF) and UNIX (LF) style files correctly.
Line Length	Enter a positive number to specify a fixed line length. Enter 0 for variable line length, the lines are delimited by the character(s) set in the Line Delimiter.
Skip Number of Lines	Skips this many lines from the start of the file.
First Char Offset	Enter a positive number to skip that many characters from the beginning of the file. Note that the length of a character can be 1 or 2 bytes or even a variable number of bytes, depending on the Code Page setting.
Chinese Chars	When set to true Chinese characters are treated as two positions wide, when set to false all characters take one position even if they are displayed with greater width. Open "SAMPLE9 Chinese Characters.iwm" in the Samples directory for an example. To display the character at the correct positions in the designer use a fixed font such as "MingLiU" or "MS Mincho", this font displays Chinese characters taking up exactly two positions. In many other fixed fonts such as "Courier New" the Chinese characters are displayed as less than two positions wide, and consequently the designer text layout will not be correct. You can set the Font via the main window: Options->Preferences.  20060712 清库清库 A55511 20060712 ZK320001 B66611 20050619 紅点紅点 C77711 *Correct: MingLiU*  20060712 清库清库 A55511 20060712 ZK320001 B66611 20050619 紅点紅点 C77711 *Incorrect: Courier New*
Pdf Enabled	Set to True to extract text from PDF files. Note that due to the nature of PDF files the results of the text extraction will vary from PDF file to PDF file. From some PDF files it will not be possible to extract text at all, for example protected files or files in which the text is stored as an image.
Pdf Character Spacing	Sets the character pitch for PDF files in 1/72 of an inch. Use 0 for automatic.

Introduction

Upon creating a new model for a delimited file, Import Wizard tries to find the best Delimiter Settings for the import and builds a model for these settings. If these settings are incorrect, you can change the Delimiter Settings and press the 'Build Model' button to rebuild the model. For an introduction in more advanced features such as Markers please read the **Tutorial**.

Window Layout



The screen is split in three subwindows:

Model Definition Window gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties Window gives the property details for the selected marker or field in the Model window.

Source File Window shows the file content split into fields according to the Delimiter Setup. Lines in the source file which match the selected marker are highlighted in yellow. If a field is selected then the field data is also highlighted in red. Discarded (ignored) lines are gray and any other lines are blue.

When you click anywhere in the Source File Window the first field for the selected marker and clicked column will be selected. If the field does not exist, a messagebox will ask you if you want to create a new field.

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Delimiter Setup

Build Model button
Pressing this button will create a new model with fields according to the selected settings. Changing **Delimiter** or **Text Qualifier** updates the File Window immediately, the fields and markers stay unchanged unless you press the Build Model button.

Delimiter
Sets the delimiter character(s) that separates fields in the import file, see Delimiter Characters Definitions.

Text Qualifier
Field starting with this character are expected to end at the next occurrence of this character. I.e. embedded delimiter characters are ignored until the closing text qualifier character is processed. Two consecutive Text Qualifier characters are treated as a single character upon import. See Delimiter Characters Definitions.

Get field names from line
This is the line number where the field names are taken from after you press the Build Model button. Line number 1 is the first line of the file. Set the line number to 0 if the file does not contain field names. The two dropdowns control the casing and filtering of non-alphanumeric characters of the fieldnames, changes are applied after the Build Model button is pressed.

Fields

To create a field: First select the marker for which you want to create a field, then click any cell in the column for which you want to create a field. If the field already exists, it will be selected and you can use the Copy button to create a duplicate. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source File

The following table only describes the Field Properties specific for Fixed imports, see **Field Properties** for a description of common Field Properties.

Field Property	Description
Column Number	The column number which is imported into this field. The first column is number 1. Column Number 0 will import the whole delimited line into the field.
Marker Number	The marker number that will trigger the import of this field.
Line	The line offset of the import field relative to the line that matched the marker pattern. I.e. line 0 means the field is on the same line, -2 means the field is two lines above the matched line.

Markers

You can create new markers by pressing the **Add Marker** button, this will create a new blank (match all) marker. See **Markers** for a detailed description of all Marker properties.

The **Column Number** property defines which column number is used for matching. The first column is number 1. Column Number 0 will match agains the whole delimited line.

The marker pattern can be edited directly above File contents. The marker Pattern can contain multiple lines. Press enter to create a new line. Notice that only a single line is shown in the .

Import Properties

Selecting Import Properties in the Import Definition window allows you to set the following properties:

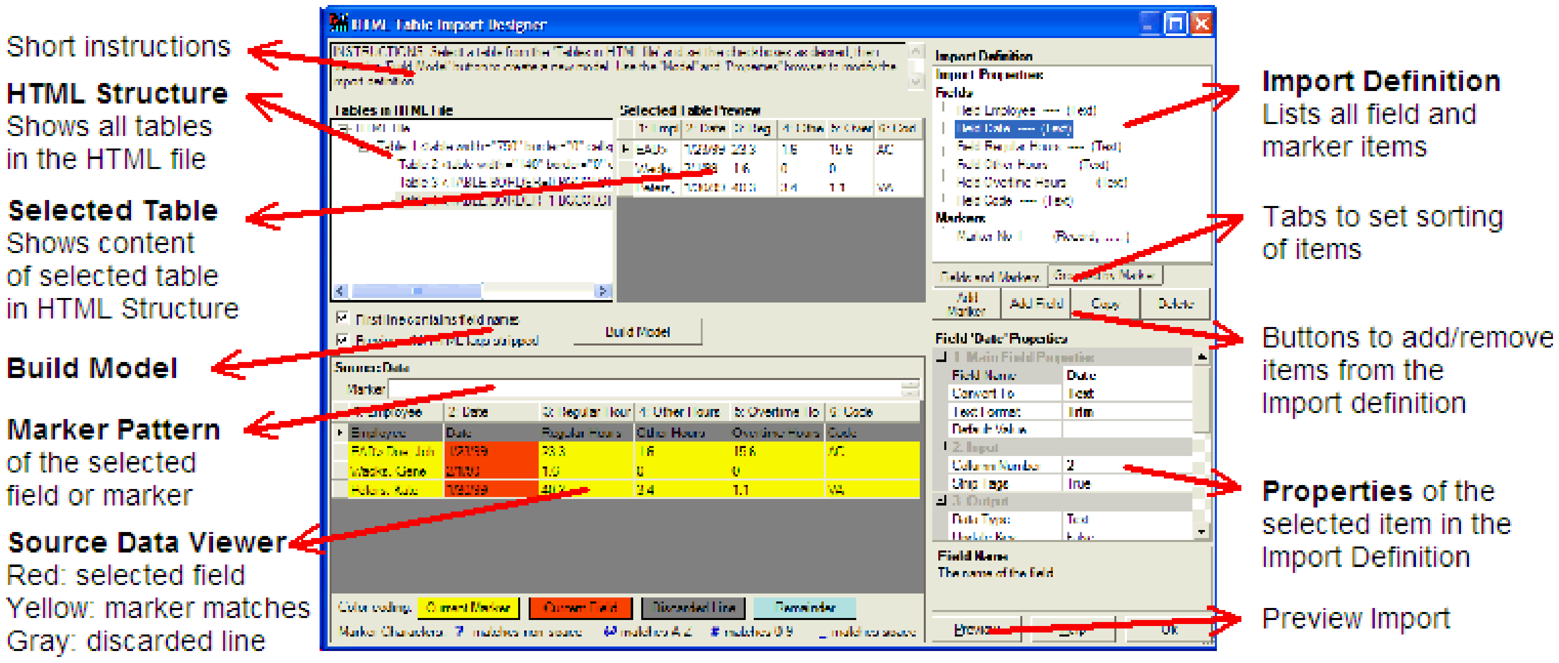
Property	Description
Year 2000 Cutoff	This is the first year for two digit year imports that will be treated as in the 20th century. This is set to 30 by default, i.e. 30-99 converts to 1930-1999, 00-29 to 2000-2029.
Code Page	Select the desired character set Code Page to read the source file. Import Wizard works internally with unicode (16bit) characters. If the required Code Page is not in the list the enter the Code Page number by hand. If the Code Page is invalid you will get a import warning upon importing.
New Line Delimiter	This parameter gives the character(s) that mark the end of a line. Default when left blank is to use {13}{10} or {10} which will import both Windows (CR+LF) and UNIX (LF) style files correctly.
Line Length	Enter a positive number to specify a fixed line length. Enter 0 for variable line length, the lines are delimited by the character(s) set in the Line Delimiter.
Skip Number of Lines	Skips this many lines from the start of the file.
First Char Offset	Enter a positive number to skip that many characters from the beginning of the file. Note that the length of a character can be 1 or 2 bytes or even variable number of bytes, depending on the Code Page setting.

Importing HTML Files

Introduction

With this designer you can import one or more HTML tables. The 'HTML Tables' window shows an overview of the tables in the HTML file, clicking on a table will show that table in the 'Table Preview' window. Pressing the 'Build Model' builds a new import model for this table. For an introduction in more advanced features such as Markers please read the **Tutorial**.

Window Layout


Short instructions
HTML Structure
Shows all tables in the HTML file
Selected Table
Shows content of selected table in HTML Structure
Build Model
Marker Pattern of the selected field or marker
Source Data Viewer
Red: selected field
Yellow: marker matches
Gray: discarded line
Import Definition
Lists all field and marker items
Tabs to set sorting of items
Buttons to add/remove items from the Import definition
Properties of the selected item in the Import Definition
Preview Import
HTML Table Import Designer
Tables in HTML File
Selected Table Preview
First line contains field names
Build Model
Source Data
Marker
Import Definition
Fields and Markers
Add Marker
Add Field
Copy
Delete
Field 'Date' Properties
Preview
OK

The screen is split the following parts:

HTML Tables shows the all the HTML Tables in the file

Selected Table show the selected table in the HTML Tables. The Selected Table might contain the following entries ###Table:x###, ###Colspan###, ###Rowspan### to indicate that at that position a embedded table exists, or that the cell is covered in a column or rowspan.

Build Model pressing the Build Model button will create a model for the selected table.

Source Data Viewer shows the data from the source file that will be used for the import, as specified with the Import Properties. Lines in the source file which match the selected marker are highlighted in yellow. If a field is selected then the field data is also highlighted in red. Discarded (ignored) lines are gray and any other lines are blue. The Source Data might contain the following entries ###Table:x###, ###Colspan###, ###Rowspan### to indicate that at that position a embedded table exists, or that the cell is covered in a column or rowspan. These entries will not show up in the actual imported table.

When you click anywhere in the Source File Window the first field for the selected marker and clicked column will be selected. If the field does not exist, a messagebox will ask you if you want to create a new field.

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Import Definition gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties gives the property details for the selected marker or field in the Import Definition window.

Build Model

Changing **Preview with HTML tags stripped** updates the Source Data and Selected Table Preview immediately. The model properties (fields and markers) stay unchanged unless you press the **Build Model** button.

Get field names from line
This is the line number where the field names are taken from after you press the Build Model button. Line number 1 is the first line of the file. Set the line number to 0 if the file does not contain field names. The two dropdowns control the casing and filtering of non-alphanumeric characters of the fieldnames, changes are applied after the Build Model button is pressed.

Preview with HTML tags stripped
This is a display setting only, it has no effect on the import. You can change the **Strip Tags** field property to control handling of HTML tags upon importing the file.

Build Model button
Pressing this button will update the model with new fields for the selected setting of 'First line contains fieldnames'.

Fields

To create a field: First select the marker for which you want to create a field, then click any cell in the column for which you want to create a field. If the field already exists, it will be selected and you can use the Copy button to create a duplicate. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source Data

The following table only describes the Field Properties specific for Fixed imports, see **Field Properties** for a description of common Field Properties.

Field Property	Description
Column Number	The column number which is imported into this field. The first column is number 1.
Marker Number	The marker number that will trigger the import of this field.
Line	The line offset of the import field relative to the line that matched the marker pattern. I.e. line 0 means the field is on the same line, -2 means the field is two lines above the matched line.
Strip Tags	When set to True all HTML tags are removed from the field.

Markers

You can create new markers by pressing the **Add Marker** button, this will create a new blank (match all) marker. See **Markers** for a detailed description of all Marker properties.

The **Column Number** property defines which column number is used for matching. The first column is number 1. NOTE: Column Number 0 will match against the first field as well, not the whole record.

The marker pattern can be edited directly above File contents. NOTE: the marker always matches against the data **with HTML tags** included, even if 'Strip Tags' has been checked. This allows for matching information inside HTML tags, such as the href="" attribute.

Import Properties

Selecting Import Properties in the Import Definition window allows you to set the following properties:

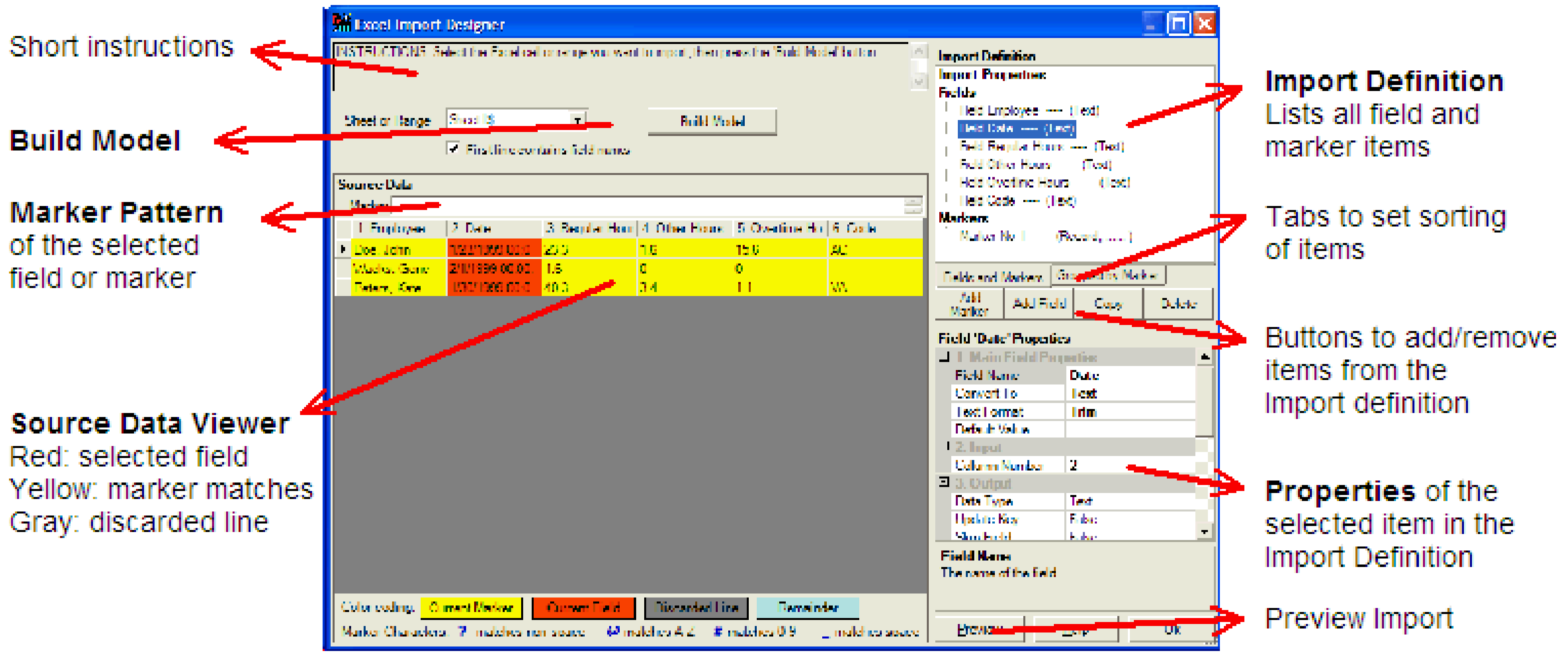
Property	Description
HTML Table(s)	The HTML table number(s) to import. Tables are numbered sequentially as they occur in the HTML file. You can use commas to separate table numbers, and dashes to indicate ranges of tables. For example '1,4-6' will import tables 1, 4, 5, and 6.
Skip Number of Lines	Skips this many lines from the start of the file.
Year 2000 Cutoff	This is the first year for two digit year imports that will be treated as in the 20th century. This is set to 30 by default, i.e. 30-99 converts to 1930-1999, 00-29 to 2000-2029.
Code Page	Select the desired character set Code Page to read the source file. Import Wizard works internally with unicode (16bit) characters. If the required Code Page is not in the list the enter the Code Page number by hand. If the Code Page is invalid you will get a import warning upon importing.

Importing Excel Files

Introduction

With this designer you can import a sheet or a named range from a Excel spreadsheet. The 'Sheet or Range' dropdown shows all sheets and ranges in the source file. Pressing the 'Build Model' builds a new import model for the selected sheet or range. For an introduction in more advanced features such as Markers please read the **Tutorial**.

Window Layout


Short instructions
Build Model
Marker Pattern of the selected field or marker
Source Data Viewer
Red: selected field
Yellow: marker matches
Gray: discarded line
Import Definition
Lists all field and marker items
Tabs to set sorting of items
Buttons to add/remove items from the Import definition
Properties of the selected item in the Import Definition
Preview Import
Excel Import Designer
Build Model
Add Field
Copy
Delete
Preview
OK

The screen is split the following parts:

Build Model pressing the Build Model button will create a model for the selected table.

Source Data Viewer shows the data for the selected sheet or range from 'Sheet or Range' dropdown.

When you click anywhere in the Source File Window the first field for the selected marker and clicked column will be selected. If the field does not exist, a messagebox will ask you if you want to create a new field.

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Import Definition gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties gives the property details for the selected marker or field in the Import Definition window.

Build Model

Sheet or Range
This dropdown is filled with all the Sheets (entries ending with $) and Named Ranges (entries without $) in the spreadsheet. Upon selecting an entry the Source Data updated. You can also enter a value directly, for example Sheet1$b10:c20, to import only part of a sheet.

If you enter a number then the corresponding sheet number will be imported. For example: '1' imports the first sheet from the Excel file.

Note: The the top-left cell and bottom-right cells of the table corresponds to the non-blank area of the selected sheet/range. For example, Sheet1 only contains data in cell B2 and C3, then Sheet1$ is a 2x2 table with top-left cell B2. This even holds true if an explicit range is specified: Sheet1$a1:d4 will still result in the same 2x2 table.

Get field names from line
This is the line number where the field names are taken from after you press the Build Model button. Line number 1 is the first line of the file. Set the line number to 0 if the file does not contain field names. The two dropdowns control the casing and filtering of non-alphanumeric characters of the fieldnames, changes are applied after the Build Model button is pressed.

Build Model button
Pressing this button will update the model with new fields for the selected setting of 'First line contains fieldnames'.

Fields

To create a field: First select the marker for which you want to create a field, then click any cell in the column for which you want to create a field. If the field already exists, it will be selected and you can use the Copy button to create a duplicate. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source Data

The following table only describes the Field Properties specific for Fixed imports, see **Field Properties** for a description of common Field Properties.

Field Property	**Description**
Column Number	The column number which is imported into this field. The first column is number 1.
Marker Number	The marker number that will trigger the import of this field.
Line	The line offset of the import field relative to the line that matched the marker pattern. I.e. line 0 means the field is on the same line, -2 means the field is two lines above the matched line.

Markers

You can create new markers by pressing the **Add Marker** button, this will create a new blank (match all) marker. See **Markers** for a detailed description of all Marker properties.

The **Column Number** property defines which column number is used for matching. The first column is number 1. NOTE: Column Number 0 will match against the first field as well, not the whole record.

The marker pattern can be edited directly above File contents. The marker Pattern can contain multiple lines. Press enter to create a new line. Note that only a single line is shown in the marker input box.

Import Properties

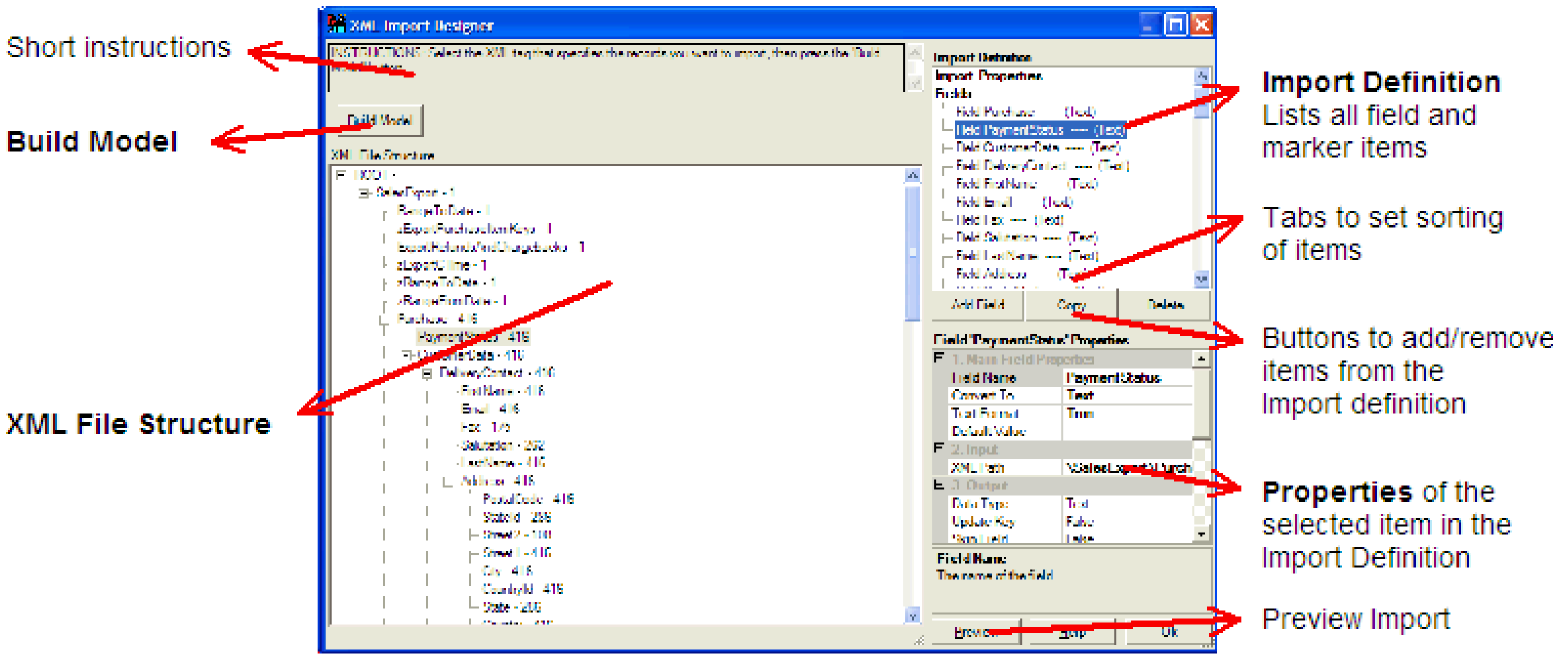
Selecting Import Properties in the Import Definition window allows you to set the following properties:

Property	**Description**
Skip Number of Lines	Skips this many lines from the start of the file. Note: if 'Get field names from line' is greater than 0 then the first line of the Excel range is filtered out by the Excel connection logic, setting 'Skip Number of Lines' to 1 in this case will skip an additional line from the Excel file, i.e. the first 2 lines are not imported.
Year 2000 Cutoff	This is the first year for two digit year imports that will be treated as in the 20th century. This is set to 30 by default, i.e. 30-99 converts to 1930-1999, 00-29 to 2000-2029.

Introduction

With this designer you can import XML files. The XML File Structure window shows the layout of the XML file, select the tag that you want to import as record then press the 'Build Model' button.

Window Layout


Short instructions
Build Model
XML File Structure
Import Definition Lists all field and marker items
Tabs to set sorting of items
Buttons to add/remove items from the Import definition
Properties of the selected item in the Import Definition
Preview Import
XML Import Designer
Build Model
XML File Structure
Import Definition
Add Field
Copy
Delete
Field 'PaymentStatus' Properties
Preview
Skip
Ok

The screen is split the following parts:

Build Model pressing the Build Model button will create a model for the selected table.

XML File Structure shows the tags and hierachy of the XML file. The numbers behind the tag names are indicate how many times the tag appears in the file.

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Import Definition gives an overview of the fields defined in the model.

Properties gives the property details for the selected marker or field in the Import Definition window.

Build Model

First select the tag that corresponds to the records you want to import. The number behind the tag indicates how many times the tag appears in the file, and also how many records will be created. Then press the **Build Model button** to create the model.

Fields

To create a field: First select the marker for which you want to create a field, then click any cell in the column for which you want to create a field. If the field already exists, it will be selected and you can use the Copy button to create a duplicate. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source Data

The following table only describes the Field Properties specific this type of imports, see **Field Properties** for a description of common Field Properties.

Field Property	**Description**
XML Path	The XML tag name including the path of all parent tags. The field data is loaded when this tag occurs in the file.

Import Properties

Selecting Import Properties in the Import Definition window allows you to set the following properties:

Property	**Description**
Record XML Path	The XML path for the Tag that creates records. Upon reading the end-tag of this path a new record is created.
Year 2000 Cutoff	This is the first year for two digit year imports that will be treated as in the 20th century. This is set to 30 by default, i.e. 30-99 converts to 1930-1999, 00-29 to 2000-2029.
Code Page	Select the desired character set Code Page to read the source file. Import Wizard works internally with unicode (16bit) characters. If the required Code Page is not in the list the enter the Code Page number by hand. If the Code Page is invalid you will get a import warning upon importing.

Importing Whole Files

Introduction

With this designer you can import whole files as single field values in a database table.

Window Layout

The screen is split the following parts:

Source File(s) are the currently selected source files. The list shows a complete listing of the individual files

The Copy and Delete buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Import Definition gives an overview of the fields defined in the model.

Properties gives the property details for the selected marker or field in the Import Definition window.

Fields

To create a field: First select the marker for which you want to create a field, then click any cell in the column for which you want to create a field. If the field already exists, it will be selected and you can use the Copy button to create a duplicate. Alternatively, you can press the **Add Field** button and change the Field Properties manually. The data which will be imported for the Field appears with red background color in the Source Data

See **Field Properties** for a description of the Field Properties.

Import Properties

Selecting Import Properties in the Import Definition window allows you to set the following properties:

Property	Description
Code Page	Select the desired character set Code Page to read the source file. Import Wizard works internally with unicode (16bit) characters. If the required Code Page is not in the list the enter the Code Page number by hand. If the Code Page is invalid you will get a import warning upon importing.

Output

Contents

Introduction

The Output dropdown on the Main Window defines the type of output that is generated by Import Wizard Pro. The dropdown is not available in the Import Wizard Access / Excel Add-Ins, the add-ins always output to the current database / active spreadsheet.

After selecting the output type from the dropdown, the output properties can be set. The properties are divided into a Mandatory and Optional properties. A mandatory property is for example the output file name is entered for file type outputs. The optional properties help to further specify the output, for example whether or not to output the fieldnames to the file.

Access Database

Requirements: Access is not required to be installed on the target PC. Required are the Microsoft MDAC and Jet 4.0 DAO database drivers which are installed as part of Access but can also be downloaded separately from www.microsoft.com, see Install.

Property	**Description**
Database File	The name of the Access .mdb file.
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.

Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Connect String	Optional DAO connect string.
Access Version	By specifying an explicit Access version, for example "Access 2000", Import Wizard will create the specified Access database version if the database file does not exist.
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.
Username	Username to use for database connection. Default is to use 'Admin' if left blank.
Password	Password to use for database connection. Default is no password.

dBase III File

The resulting dBase (.dbf) file can be read by dBase version III and higher.

Requirements: None.

Limitations: Text fields are limited to 254 characters. Field names are limited to 11 characters, longer names will be truncated. Only Create and Insert are supported as Table and Record actions.

Property	Description
Output File	The name of the dBase .dbf file.
Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped).
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table.

Delimited File

Outputs to a delimited text file.

Requirements: None.

Limitations: Only Create and Insert are supported as Table and Record actions.

Property	Description
Output File	The name of the output delimited text file.
Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped).
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table.
Output Field Names	If set to True the field names are stored in the first row in the output table.
Delimiter	The delimiter string used to delimited the fields. Default value is {comma}, see Delimiter Characters Definitions
Text Qualifier	A single character string. Text fields will be embedded between this character. Embedded text qualifiers in an output string will be replaced by two text qualifier characters. Default value is {dblquote}, see Delimiter Characters Definitions

Line Delimiter	The delimiter string used to delimited the output lines (records). Default value is {newline}, see Delimiter Characters Definitions
Date Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.
Numeric Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.

Excel Spreadsheet

Requirements: Excel needs to be installed for this output to work.

Property	**Description**
Output File	The name of the Excel .xls file.
Output Cell/Range	The name of the cell or range in the output Excel file where the imported data is stored. For example: **Sheet1!A1**
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Output Field Names	If set to True the field names are stored in the first row in the output table. Note that Field Names are required for the "Update" Table Actions.
Date Format	Sets formatting for numbers. Enter an Excel date format string, see Excel documentation for details.
Numeric Format	Sets formatting for numbers. Enter an Excel numeric format string, see Excel documentation for details.
Screen Updating	If set to True the Excel screen is updated while an import is running.

Excel Version 2 File

Unlike the "Excel Spreadsheet" output this ouput does not require Excel to be installed. The resulting Excel (.xls) file can be read by Excel version 2.1 and higher.

Requirements: None.

Limitations: Text fields are limited to 255 characters and the file can contain 65535 records maximum. Only Create and Insert are supported as Table and Record actions.

Property	**Description**
Output File	The name of the Excel .xls file. The file will contain a single sheet, the table is stored starting in cell A1.

Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped).
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table.
Output Field Names	If set to True the field names are stored in the first row in the output table.

HTML File

Creates a HTML file that contains the output table.

Requirements: None.

Limitations: Only Create/Append and Insert are supported as Table and Record actions.

Property	Description
Output File	The name of the output HTML file.
Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The output file is recreated on every import. An existing output file will be deleted. **Append to Table** If the output file exists the new table will be appended to the file, otherwise a new file is created.
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table.
Output Field Names	If set to True the field names are stored in the first row in the output table.
Date Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.
Numeric Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.

MySQL Database

Connects to MySQL 4.1 or later, no external drivers are required. For MySQL 4.0 or earlier use the ODBC output in combination with the MySQL ODBC driver that can be downloaded from www.mysql.com.

Requirements: None.

Property	Description
Database	The name of the MySQL database.
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.

Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Server	Domain name or IP address of the MySQL server. Default is Localhost (127.0.0.1).
Port	The port of the MySQL server. Default is 3306.
Username	Username to use for database connection. Default is 'root' if left blank.
Password	Password to use for database connection. Default is no password.
Connect String	Connect string properties in addition to the Server, Port, Username, Password and Database properties can be specified here. Available properties are: "connection timeout", "persist security info", "pooling", "min pool size", "max pool size" "connection lifetime". For example: pooling=yes;
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.

ODBC Database

To setup the ODBC connection, click the "..." button next to the "Connect String" property

Requirements: To use this output you need to have MDAC installed, which is available from www.microsoft.com, see Install. In addition you need an appropiate ODBC Driver to connect to the database.

Property	**Description**
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Connect String	The ADO Connect string required to connect to the database.
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.

Oracle

Requirements: Oracle client software version 8.1.7 or later.

Property	**Description**
Database	The name of the Oracle data source.
Table Name	The table where the imported data is stored.

Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Integrated Security	Sets the authentication method, True: use Integrated Security, False: use Username and Password
Username	Username, not used with Integrated Security.
Password	Password, not used with Integrated Security.
Connect String	Connect string properties in addition to the Data Source, Thrusted Connection, Username and Password properties can be specified here.
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.

Postgres Database

Requirements: None.

Property	**Description**
Database	The name of the Postgres database.
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Server	Domain name or IP address of the Postgres server. Default is Localhost (127.0.0.1).
Port	The port of the Postgres server.
Username	Username to use for database connection. Default is 'postgres'.
Password	Password to use for database connection. Default is no password.

Connect String	Connect string properties in addition to the Server, Port, Username, Password and Database properties can be specified here.
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.

SQL Script File

Output a SQL Script file that can be executed on the target database server. The Escape properties can be used to set the appropiate SQL-dialects.

Requirements: None.

Limitations: Only Create is supported as Table action.

Property	Description
Output File	The name of the output SQL script file.
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The output file is recreated on every import. An existing output file will be deleted.
Record Action	The type of SQL statements to generate: **Insert** Output "INSERT" statements. **Update** Output "UPDATE" statements based on the "Update Key" fields. **First Update, if fail then Insert** Output both "UPDATE" and "INSERT" statements.
Date Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.
Numeric Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.
Text Escape	The character to use to escape embedded single qoutes (') in text fields.
Name Escape	The characters to use to escape field names. For example **[]** (square brackets) or ` (backtick).

SQL Server / MSDE Database

Use this output for with SQL Server version 7.0 or later databases, use the ODBC output for earlier versions of SQL Server.

Requirements: None.

Property	Description
Server	The name of the SQL server.
Database	The name of the database.
Table Name	The table where the imported data is stored.
Table Action	The action to take before starting the import: **Append to Table** If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created. **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped). **Empty Existing Table** If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.

Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table. **Update** Existing records in the table are updated with the imported record based on the "Update Key" fields. **First Update, if fail then Insert** Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.
Thrusted Connection	Sets the authentication method, True: use Windows Authentication, False: use SQL Server Authentication (enter Username and Password)
Username	Username for SQL Server Authentication, not used for Windows Authentication.
Password	Password for SQL Server Authentication, not used for Windows Authentication.
Connect String	Connect string properties in addition to the Server, Thrusted Connection, Username, Password and Database properties can be specified here.
Pre Import SQL	SQL statement(s) that are executed before the import is started.
Post Import SQL	SQL statement(s) that are executed after the import has completed.

XML File

Outputs to a XML text file.

Requirements: None.

Limitations: Only Create and Insert are supported for Table and Record actions.

Property	Description
Output File	The name of the output XML file.
Root Tag	The name of the root element of the XML file. Default is "Table".
Record Tag	The name of the record elements of the XML file. Default is "Record".
Table Action	The action to take before starting the import: **Confirm Create New Table** The table is recreated on every import. If the table already exists the table will be deleted after confirmation by the user. **Create New Table** The table is recreated on every import. An existing table will be deleted (dropped).
Record Action	The action to take upon storing a record: **Insert** The record is appended to the end of the table.
Output Field Names	If set to True the field names are stored in the first row in the output table.
Date Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.
Numeric Format	Sets formatting for numbers. See Date and Numeric Format Strings for detailed specification.

What's New

Version 9 Upgrade Notes

- Import Wizard 9 uses a new model file format. Files saved with version 9 can not be opened in previous versions, we recommend to create a backup copy of your existing model files before using version 9.
- Version 9 can open unencrypted model files created with version 7 or 8. To upgrade encrypted model files and model files created with earlier versions: use the IwmConvert.exe utility in the c:\Program Files\Import Wizard 9 directory. Alternatively, you can open + save the iwm files with version 8, now Import Wizard 9 can open the files.
- The VBA Functions changed.
- The Command Line options changed.
- The add-ins for Access 97 and Excel 97 have been removed. If you still use Access/Excel 97 use Import Wizard version 8 or let us know and we will consider re-adding the add-ins for Office 97.
- The Filter and Formulas are now compiled VB.Net code for faster execution. In version 8 VBScript was used, see Formulas.

Version 9 Change Log

For a detailed change log, please see the ReadMe.txt file.

Conversion Field Properties

Field Properties

These field properties are present for all import types. Additional properties are available for individual import types.

Field Property	Description
Field Name	This is the field name in the destination table. Make sure that the name follows the naming conventions of the database you are using. For example: field names in Excel should not contain a period “.”, an exclamation mark “!”, an accent grave “`” or square brackets “[]”.
Convert To, **Text Format**, **Date Format**, **Fraction Mark**, **Multi Line** and **Variable Name**	These properties specify the conversion rules to be used, see Field Conversion Properties for details.
Data Type	If Import Wizard creates a new table, the field will be created with this data type. Import Wizard does not change the structure of existing tables unless "Create Table" is specified as Import Action in the main window.
Update Key	When the "Update" Import Action specified in the main window this property determines which records are updated (replaced).
Skip	If set to True the field will not be stored in the database, useful for creating calculated fields that hold intermediate results.
Repeat Empty	When set to True the last imported value will be repeated if the new field value is emtpy (blank).
Repeat No Import	When set to True on the program will "remember" the last imported value until a new value is loaded from the source file. This function is useful for importing headers in order to repeat the header field information for each record under the heading. The last imported value is repeated, even if this value was blank.
Delim Field No	The sub field number to import from the string extracted from the source file using the Length, Start, and Line properties. The delimited field number is counting from left to right for positive numbers, counting from right to left for negative numbers. I.e. –2 is the second sub field from the right, 1 is the left most field. For example the comma delimited string "aa,bb,cc" will return "cc" for delim field number 3 or –1.
Delimiter	The delimiter to use to subdivide the field. A delimiter can contain more than one character and any combination of characters and characters defined using the curly brackets. For example {SPACE}-{TAB} for fields delimited by a space,dash,tab.
Text Qualifier	If a field starts with this character then the next field is expected after a second occurrence of the Text Qualifier. Two consecutive Text Qualifier characters are treated as a single character upon import. The text qualifier is used to contain delimiter characters within a field. For example using a semicolon as delimiter for the string **a;"b;c"** and an empty text qualifier results in 3 subfields: **a**, **"b** and **c"**, with the double quote (") as text qualifier the string is split into 2 subfields: **a** and **b;c**.
Formula Name	This read only property represents the field name for use in formulas. All non alpha-numeric characters in the field name are replaced by an underscore ("_"). Also, if the fieldname does not start with an alpha character an "a" is inserted in front of the fieldname.
Formula	If a formula is entered, the result of the formula will be stored in the field. For detailed information on formulas see Formulas. The formula can also contain a regular expression.

Formula Evaluation	This property controls the moment when the formula is evaluated. Possible settings are: **On Append, After Properties** The formula is evaluated just before the record is appended to the output table, but after the other properties of the field (Convert To, Delimiter, etc.) are evaluated. **On Append, Before Properties** The formula is evaluated just before the record is appended to the output table. The formula is evaluated before the other properties are evaluated. **On Load, After Properties** The formula is evaluated directly after the field data is loaded from the source file, but after the other properties of the field are evaluated. **On Load, Before Properties** The formula is evaluated directly after the field data is loaded from the source file. The formula is evaluated before the other properties are evaluated. Note: The "On Append" fields are evaluated in a single pass in the order in which they appear in the Model Definition. For example: two fields fld1 and fld2 with loaded values 1 and 2 and with formulas **fld2+1** and **fld1+1** respectively, results in a record with fld1 = 3 and fld2 = 4 (and not 2 as expected) because fld1 is evaluated before fld2.

Processing Order

The field properties are processed in the following order:

Conversion Step	Description
1. Load Field	The field data is loaded from the source file based on the following field properties **Marker Number**, **Line**, **Start**, **Length**, **Column Number**, **Strip Tags**, and **XML Path** properties. The actual properties used depend on the type of import. The **Formula Evaluation** property determines whether conversion steps 2-7 are excuted immediately (Formula Evaluation is "On Load"), or just before storing the record in the database (Formula Evaluation is "On Append").
2. Variable	If **Convert To** is set to "Variable" then the field is loaded with data corresponding to **Variable Name**.
3. Formula	If **Formula Evaluation** is set to "Before Properties" then the **Formula** is evaluated an the result of the formula is stored as the field value.
4. Delimited	If **Delim Field No** is not "0" then the field data is split according to the **Delim Field No**, **Delimiter** and **Text Qualifier** properties.
5. Type Conversion	The data is converted according to the **Convert To**, **Text Format**, **Date Format**, **Fraction Mark** and **Multi Line** properties, see Type Conversion.
6. Formula	If **Formula Evaluation** is set to "After Properties" then the **Formula** is evaluated an the result of the formula is stored as the field value.
7. Repeat on Empty	If the field value is empty (null) at this point and the **Repeat on Empty** property is set then the field value is reset to the last field value.
	If the field value is empty (null) at this point then the field value is set to the **Default Value**.

Type Conversion

The **Convert To** property in combination with the **Text Format**, **Date Format**, **Fraction Mark**, **Multi Line** and **Variable Name** field properties specify how the imported data is converted into meaningful values. Convert To specifies what type of value the imported data is converted to, the other properties set further details for the selected Convert To type.

Convert To: Text

Converts the field to text. **Text Format** can have the following values:
Trim (default) stores the imported data without leading and trailing spaces.
Raw stores the import data without modification.

Convert To: Date

Converts the field to a date, time or date/time. **Date Format** specifies how to convert the imported data to a date. When Date Format is left blank (default) the imported data will be converted to a date according to your system locale settings. If you are importing dates that does not match your system locale setting you will have to enter the **Date Format** property to specify how to convert the imported data. There are two date formats, delimited dates and fixed position dates.

In a **delimited date** the individual parts of a date (month, day, year, hour, minute, second) are delimited (separated) by one or more non-alphanumeric characters. Examples of delimited dates are: "1/23/99", "DEC-2-2001", "January 3, 1999". Delimited dates can be imported by specifing a Conversion Method using any combination of **M**, **D**, **Y**, **H**, **N** or **S** to define the position of the month, day, year, hour, minute, and second parts within the delimited date. For example: the format **MDY** will correctly import all three dates prevously mentioned.

In a **fixed position date** the individual parts of the date have a fixed character position in the imported data. For fixed dates set Conversion Method using: **MM**, **MMM**, **DD**, **YY**, **YYYY**, **HH**, **NN** and **SS** to specify the positions of the date-parts. Any other character in Conversion Method will be ignored. For example: the Conversion Method **YYYYMMDDHH:NN:SS** imports "1999123123:59:12" correctly as "December 31, 1999 23:59:12".

Sometimes a delimited date string contains information such as the day of the week that should be ignored. Use the **X** in Conversion Method to ignore that part of a date. For example the string "Friday, November 26, 1999" will not import correctly using **MDY** because the program will attempt to convert the first date-part "Friday" into a month value. The format **XMDY** will work correctly, as it informs the program to ignore the first part of the date string.

Notes:
- In Conversion Method you can not mix delimited date formats with fixed width formats. If you do, the Conversion Method will be treated as a fixed date.
- Dates with a two digit years will be imported according to the "1900 Cutoff" setting.
- Missing date-parts are set to default values. Default values are 1 for day, January for month, the system year for year, 0 for hour, minute, and second. For example: Conversion Method **MY** imports "4/2003" as "May 1, 2003".
- Blank date values and dates with day or month equal to "0" are stored as Null.

Convert To: Numeric

Converts the field to a numeric value. The **Fraction Mark** defines the fraction (decimal) mark, default is a period (.). The program ignores all characters except digits "0" to "9" and the fraction mark. If the imported data contains a negative sign "-" or an opening round brackets "(" then the number is considered to be negative. Any other characters are ignored. For example: 12X34- is converted to -1234.

By leaving the **Fraction Mark** empty the field is converted to a number using the system locale settings. This is the fastest but least flexible conversion. The conversion stops upon finding the first non-numeric character For example: 12X34- is converted to 12.

Convert To: Multi Line

Converts multiple lines from the source file to a single field. **Multi Line End** defines at which line the multi line field ends, this can be when a marker matches or upon a blank line. The **Multi Line Clip**, **Multi Line Trim** and **Multi Line Line Break** further specify how the multi line field is converted.

Multi Line End	Description
Any Marker	The multi line field ends on the line where any of the defined markers matches.
Marker m,n,...	Where **m** and **n** stand for numbers. The multi line field ends on the line where one of the indicated markers matches. For example **Marker 1,5,6** will end the multiline field on the line where marker 1, 5, or 6 matches.
Blank Line	Ends the multiline field when a blank line is encountered.

Multi Line Property	Description

Multi Line Clip	**True**: Imports characters starting from the Start position up to the Start+Length position for each imported line. This is the default behaviour. **False**: Imports full lines starting from the Start character position on the first imported line, ending at the Start+Length character position on the last imported line.
Multi Line Trim	**True**: Each line is trimmed of leading and trailing spaces. This is the default behaviour. **False**: Leading and trailing spaces are not removed from the lines.
Multi Line Line Break	The line breaks are replaced with the string of this property value. The string can contain characters defined in Delimiter Characters Definitions. Default: {13}{10} (Carriage Return followed by Linefeed).

Example: The following settings terminates the multi line field on the line where marker 2 matches. The multi line field uses Clip mode, leading and trailing spaces are removed from each line and the line breaks between the lines are replaced with a single space. The multiline field is converted to a single paragraph.
Multi Line End=Marker 2
Multi Line Clip=True
Multi Line Trim=True
Multi Line Line Break={space}

Convert To: Formula

No type conversion takes place, the result of the formula or regular expression in the **Formula** property is stored in the field.

Convert To: Variable

The field will contain the value from the variable selected from the **Variable Name** dropdown:

Variable Name	**Description**
FileName	File name of the imported file.
FilePath	Path of the imported file.
FileFull	Full file name, path plus name, of the imported file.
FileDateTime	Create date/time of the imported file.
RecordNo	Record number within the current file (counting filtered records only).
RecordID	Record number within the current file (counting all records).
TotalRecordNo	Record number (counting filtered records only).
TotalRecordID	Record number (counting all records).
LineNo	Line number in the text file where a match was found for the record marker that imported the record.
FileContents	Loads the whole file contents as a string value.

Markers

Introduction

Markers are a central feature of Import Wizard. Markers allow you to mark (select) lines in the source file and to perform operations on these marked lines. You control which lines are marked by editing the **Marker Pattern** input box. A line in the source file is marked when the marker pattern matches the line.

A marker pattern works similar to an asterix (*) in filenames. To list all the filenames that have an .asc extension you would enter ***.asc**. Here, the "*" is a wildcard character and stands for none or more characters. The other characters (".", "a", "s", and "c") have no special meaning and simply match the same character.

The only wildcard characters for filenames are the asterix (*) and the question mark (?), all other characters have no special meaning. For Marker Patterns a far more extensive set of wildcard characters is available.

Marker Processing

The marker patterns of each marker defined in the model are compared to each line in the source file. If a match occurs between the marker pattern and a line from the source file the action set by the Marker Type is taken. The markers are processed in the order that they appear in the model. Most imports models can be defined with:

- A single Record Marker in the last (lowest) position in the Model Definition with a none or more Header Markers above the Record Marker
- Or, a single StartRecord Marker in the first (topmost) position in the Model Definition with a none or more Header Markers below the StartRecord Marker.

Marker Types

Marker Type	Match Action
Header	When a Header Marker matches then the fields attached to the marker are loaded with data from the source file.
Record	The Record Marker indicates the end of a record in the source file. When a Record Marker matches, first the fields attached to the marker are loaded with data from the source file, then the current record is appended to the output table.
Start Record	The Start Record Marker indicates the beginning of a record. When a Start Record Marker matches, first the current record is appended to the output table, then the fields attached to the marker are loaded with data from the source file. The first match of a Start Record Marker for a source file is ignored, this is to prevent storing empty or partial records.
Discard	When a Discard Marker matches the line is discarded from the source file. To the import routine it appears as if the line did not exist in the source file. Main use of this marker type is to remove headers from a file with records and headers intermixed in such a way that the records can not be correctly matched. Any fields defined for a Discard marker will never be imported.
Footer	The Footer marker works differently; before the import is started the source file is scanned for the first line that matches the Footer marker. The fields attached to the marker are loaded with data from the match location. After this the actual import is started by processing the Discard, Header and Record markers for each line in the source file. The Footer marker can be used to load fields from lines that are below multiple records, such as a summary or a footnote. The Footer marker is not intended to load data that occurs within every record, use a Record or Header marker for this.

Note: At the end of a source file an implicit Record Marker is placed which will import the current record to the output table if any field of the current record was loaded (has changed) since the last record was appended.

Marker Pattern

The marker pattern is compared against the input lines and when a match is found the approptiate action for the marker type is taken. The following characters and strings have special meaning when used in a marker pattern:

Pattern	Matches this in the source file
Space	Any character at this position.
?	A non-space character at this position.
@	An alpha (a-z and A-Z) character at this position.
#	A digit (0-9) character at this postion.
_	A space character at this postion.
<New Line>	An empty line or a line containing only spaces.
<New Page>	A line containing the form feed character (ASCII code 12).
<LineNo n,m>	Matches line number n and repeat every m lines. Parameter m is optional. Example: <LineNo 1,4> will match the 1st, 5th, 9th, 13th, etc. lines of the source file.
/ /	If the marker pattern starts with a forward slash (/) then the pattern is interpreted as a Regular Expression which allows for more powerful matching.

Markers can have more than one line. A multiline marker matches when each of the lines of the marker matches a corresponding line in the source file.

Discard marker patterns can only contain a single line, if a discard marker pattern contains more than one line only the first line of the pattern will be used.

Marker Pattern Examples

Hello_@@ Matches a line that starts with "Hello", a space and two alpha characters (case sensitive).

##.### Matches a line starts with two digits, a period, and another three digits.

/i/hello Matches a line that contains "hello" at any character position (case insensitive).

First line: **<New Line>** second line: **###** Matches a blank line followed by a line that start with three digits.

Marker Number

The Marker Number merely identifies the marker. It is allowed to define two or more markers with the same marker number. Upon importing the fields for the marker number are loaded when any of the markers with that marker number is matched. For example if you define two markers "#" and "A" both with marker number 1, then the fields for marker number 1 are loaded when the first character of a line is either numeric (0-9) or "A".

Line Offset

This property is only available for Discard Markers. The Line Offset defines the positive or negative number of lines from the line where the match occured to the line that is dicarded. For example, a Line Offset of -1 will discard the line directly above the line that is matched by the marker.

Formulas

Formulas are used in the Formula field property and in the Filter model property on the Main Window.

You can use any Visual Basic (VB.Net namespace Microsoft.VisualBasic) expression in your formula, such as Mid(), Left(), DateSerial(), see below for a summary of available operators and functions. Use the name specified in the read-only field property Formula Name to refer to the content of a field. The Formula Name property translates the Name of the field to a name without spaces or special characters so that it can be used in formulas.

Examples:

Filter

Left(Some_Field,3)<>"Doe"

This will import only records where the first 3 characters of the field "Some Field" are not equal to "Doe"

Field Formula

"S" & Right("000000" & Sales_Order_No,6)

This will convert field "Sales Order No" to a string starting with an "S" followed by 6 digits, i.e. 56 is converted to "S00056".

DateSerial(1999,12,31)

VBScript Operators and Functions Summary

This summary contains the for data conversion most useful operators and functions. Arguments between square brackets [] are optional.

Logical Test Function Summary

IIf(expression, truepart, falsepart)

Inline if function, if expression evaluates to true then truepart is returned, otherwise falsepart is returned. Note that this function fails if either expression, truepart, or falsepart contains an error.

Example: IIf(overtime_hours>0,"OVERTIME","NO OVERTIME")

IsNull(expression)

Returns True if expression is Null or empty, otherwise this function returns False.

IfNull(expression, nullvalue)

Returns nullvalue if expression is Null or empty, otherwise this function returns expression.

Example: IfNull(overtime_hours,0)
This will replace empty (null) overtime values with a 0 (zero).

Operator Summary

&

Used to force string concatenation of two expressions.

^ * / Mod + -

Mathematical operators: raise a number to the power of an exponent (^), multiply two numbers (*), divide two numbers (/), divide two numbers and return only the remainder (Mod), sum two numbers (+), find the difference between two numbers or to indicate the negative value of a numeric expression (-).

< <= > >= = <>

Comparison operators: less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (=), not equal to (<>).

And Or Not

Logical operators: logical conjunction on two expressions (And), logical disjunction on two expressions (Or), logical negation on an expression (Not).

AndAlso OrElse

Performs short-circuiting logical conjunction on two expressions. A logical operation is said to be short-circuiting if the compiled code can bypass the evaluation of one expression depending on the result of another expression. If the result of the first expression evaluated determines the final result of the operation, there is no need to evaluate the other expression, because it cannot change the final result. Short-circuiting can improve performance if the bypassed expression is complex, or if it involves procedure calls.

Example: **IsNull(Field1) AndAlso Field2=3**

If Field1 is empty, the result is True. The the second part 'Field2=3' is only evaluated when Field1 is not empty.

String Functions Summary

Lcase(string), Ucase(string)

Convert string to lowercase or uppercase.

Len(string)

Find length of a string.

Mid(string, start[, length]), Left(string, length), Right(string, length)

Return a substring of string.

InStr([start,] strHaystack, strNeedle)

Returns the position of the first occurrence of strNeedle in strHaystack. The optional argument start sets the starting position for each search. If omitted, search begins at the first character position. Returns 0 if no match was found.

LTrim(string), RTrim(string), Trim(string)

Returns a String containing a copy of a specified string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).

Asc(string)

Returns an Integer representing the character code corresponding to the first letter in a string.

Chr(charcode)

Returns a String containing the character associated with the specified character code.

Date/Time Functions Summary

DateSerial(year, month, day)

Returns a Date for a specified year, month, and day.

DateValue(date)

Returns a Date. The required date argument is normally a string expression representing a date from January 1, 100 through December 31, 9999. However, date can also be any expression that can represent a date, a time, or both a date and time, in that range.

Year(date), Month(date), Day(date), Hour(date), Minute(date), Second(date)

Returns an Integer specifying respectively the year, month, day, hour, minute, or second as of the specified date.

Now, Date, Time

Returns current date and time (Now), date (Date) or time (Time) according your computer’s system date and time.

Math Functions Summary

Int(number)

Returns a value of the type passed to it containing the integer portion of a number.

Abs(number), Atn(number), Cos(number), Exp(number), Log(number), Sin(number), Sqr(number), Tan(number)

Return result of the math function applied.

Rnd[(seed)]

The Rnd function returns a random number less than 1 but greater than or equal to zero.

Regular Expressions

Contents

Introduction

The regular expression language is designed and optimized to manipulate text. The language comprises two basic character types: literal (normal) text characters and metacharacters. The set of metacharacters gives regular expressions their processing power.

You are probably familiar with the ? and * metacharacters used with the DOS file system to represent any single character or group of characters. The DOS file command DIR *.DOC displays a directory listing of all files a .DOC file name extension. The metacharacter * stands in for any file name in front of the file name extension .DOC. Regular expressions extend this basic idea many times over, providing a large set of metacharacters that make it possible to describe very complex text-matching expressions with relatively few characters.

It is common when discussing regular expressions to analyze them based on text they would or would not match. There are three players in the regular expression interaction: the regular expression **pattern**, the **input** string, and any **matches** the pattern makes within the string.

Regular Expressions in Import Wizard

In Import Wizard regular expressions can be used in marker strings and in field formulas. To enter a regular expression enter a forward slash, followed by the options, followed by another forward slash and the regular expression itself:

/<options>/<regular expression pattern>

The two forward slashes are mandatory, even if no options are given. The first forward slash has to be at the first character position, otherwise the string will not be interpreted as a regular expression.

The **<options>** are:

Option	Description
i	Ignore Case - Causes the pattern to ignore case sensitivity when matching the search string. Matching is case sensitive by default.
0-9	Only used in field formulas, ignored in markers. The formula's value is set the n-th substring. Also, 1 will return the first substring, 2 the second, etc. 0 will return the full match of the regular expression. (Default is 0)
m	Multiline Mode - Has nothing to do with how many lines are in the input string. Rather, this changes the meaning of ^ and $ so that they match at the beginning and end, respectively, of any line, not just the beginning and end of the whole string.
s	Single-Line Mode - Has nothing to do with how many lines are in the input string. Rather, this changes the meaning of the period character (.) so that it matches every character (instead of every character except \n).

The **<regular expression pattern>** is the pattern to match, how to define the pattern is described in the following sections.

Simple Expressions

The simplest regular expression is one you're already familiar with—the literal string. A particular string can be described, literally, by itself, and thus a regular expression pattern like **foo** would match the input string **foo** exactly once. In this case, it would also match the input: **The food was quite tasty**, which might be not be desired if only a precise match is sought.

Of course, matching exact strings to themselves is a trivial implementation of regular expressions, and doesn't begin to reveal their power. What if instead of **foo** you wanted to find all words starting with the letter **f**, or all three letter words? Now you've gone beyond what literal strings can do (within reason)—it's time to learn some more about regular expressions. Below is a sample literal expression and some inputs it would match.

Pattern	**Inputs (Matches)**
foo	foo, food, foot, "There's evil afoot."

Quantifiers

Quantifiers provide a simple way to specify within a pattern how many times a particular character or set of characters is allowed to repeat itself.

The following table describes the metacharacters that affect matching quantity.

Quantifier	**Description**
*****	Specifies zero or more matches; for example, `\w*` or `(abc)*`. Equivalent to **{0,}**.
+	Specifies one or more matches; for example, `\w+` or `(abc)+`. Equivalent to **{1,}**.
?	Specifies zero or one matches; for example, `\w?` or `(abc)?`. Equivalent to **{0,1}**.
{*n*}	Specifies exactly *n* matches; for example, `(pizza){2}`.
{*n*,}	Specifies at least *n* matches; for example, `(abc){2,}`.
{*n*,*m*}	Specifies at least *n*, but no more than *m*, matches.

Quantifiers always refer to the pattern immediately preceding (to the left of) the quantifier, which is normally a single character unless parentheses are used to create a pattern group. Below are some sample patterns and inputs they would match.

Pattern	**Inputs (Matches)**
fo*	foo, foe, food, fooot, "forget it", funny, puffy
fo+	foo, foe, food, foot, "forget it"
fo?	foo, foe, food, foot, "forget it", funny, puffy
ab{2}c	abbc, aaabbccc
ab{,2}c	ac, abc, abbc, aabbcc
ab{2,3}c	abbc, abbbc, aabbcc, aabbbcc

Metacharacters

The constructs within regular expressions that have special meaning are referred to as metacharacters. You've already learned about several metacharacters, such as the *****, **?**, **+**, and **{ }** characters. Several other characters have special meaning within the language of regular expressions. These include the following: **$ ^ . [(|)]** and \.

The **.** (period or dot) metacharacter is one of the simplest and most used. It matches any single character. This can be useful for specifying that certain patterns can contain any combination of characters, but must fall within certain length ranges by using quantifiers. Also, we have seen that expressions will match any instance of the pattern they describe within a larger string, but what if you only want to match the pattern exactly? This is often the case for validation scenarios, such as ensuring the user entered something that is the proper format for a postal code or telephone number. The **^** metacharacter is used to designate the beginning of a string (or line), and the **$** metacharacter is used to designate the end of a string (or line). By adding these characters to the beginning and end of a pattern, you can force it to only match input strings that exactly match the pattern. The **^** metacharacter also has special meaning when used at the start of a character class, designated by hard braces **[ ]**. These are covered below.

The \ (backslash) metacharacter is used to "escape" characters from their special meaning, as well as to designate instances of predefined set metacharacters. These too are covered below. In order to include a literal version of a metacharacter in a regular expression, it must be "escaped" with a backslash. So for instance if you wanted to match strings that begin with "c:\" you might use this: **^c:\\** Note that we used the **^** metacharacter to indicate that the string must begin with this pattern, and we escaped our literal backslash with a backslash metacharacter.

The **|** (pipe) metacharacter is used for alternation, essentially to specify 'this OR that' within a pattern. So, the regular expression **ab|ac** would match anything with 'ab' or 'ac' in it. For performance reasons it is generally better to avoid the | metacharacter if possible, the given example can be rewritten as **a[bc]**.

Finally, the parentheses **( )** are used to define substrings within the pattern. These substrings can be extracted with the 0-9 option.

Some examples of metacharacter usage are listed below.

Pattern	**Inputs (Matches)**
.	a, b, c, 1, 2, 3
.*	Abc, 123, any string, even no characters would match
^c:\\	c:\windows, c:\\\\\, c:\foo.txt, c:\ followed by anything else
abc$	abc, 123abc, any string ending with abc
(abc){2,3}	abcabc, abcabcabc

Character Classes

Character classes are a mini-language within regular expressions, defined by the enclosing hard braces **[ ]**. The simplest character class is simply a list of characters within these braces, such as **[aeiou]**. When used in an expression, any one of these characters can be used at this position in the pattern (but only one unless quantifiers are used). It's important to note that character classes cannot be used to define words or patterns, only single characters.

To specify any numeric digit, the character class **[0123456789]** could be used. However, since this would quickly get cumbersome, ranges of characters can be defined within the braces by using the hyphen character, **-**. The hyphen character has special meaning within character classes, not within regular expressions (thus it doesn't qualify as a regular expression metacharacter, exactly), and it only has special meaning within a character class if it is not the first character. To specify any numeric digit using a hyphen, you would use **[0-9]**. Similarly for any lowercase letter, you could use **[a-z]**, or for any uppercase letter **[A-Z]**. The range defined by the hyphen depends on the character set being used, so the order in which the characters occur in the (for example) ASCII or Unicode table determines which characters are included in the range. If you need a hyphen to be included in your range, specify it as the first character. For example, **[-.?]** would match any one of those four characters (note the last character is a space). Also note, the regular expression metacharacters are not treated special within character classes, so they do not need escaped. Consider character classes to be a separate language from the rest of the regular expression world, with their own rules and syntax.

You can also match any character except a member of a character class by negating the class using the carat **^** as the first character in the character class. Thus, to match any non-vowel character, you could use a character class of **[^aAeEiIoOuU]**. Note that if you want to negate a hyphen, it should be the second character in the character class, as in **[^-]**. Remember that the **^** has a totally different meaning within a character class than it has at the start of a regular expression pattern.

The following table summarizes character matching syntax.

Character class	**Description**
.	Matches any character except **\n**. If modified by the Singleline option, a period character matches any character. For more information, see Regular Expression Options.
[*aeiou***]**	Matches any single character included in the specified set of characters.

[^*aeiou*]	Matches any single character not in the specified set of characters.
[0-9a-fA-F]	Use of a hyphen (–) allows specification of contiguous character ranges.
\p{name}	Matches any character in the named character class 'name'. Supported names are Unicode categories. For example Ll, Nd, Z, and Sc (currency).
\P{name}	Matches text not included in the named character class 'name'.
\w	Matches any word character. For non-Unicode implementations, this is the same as **[a-zA-Z_0-9]**. In Unicode categories, this is the same as **[\p{Ll}\p{Lu}\p{Lt}\p{Lo}\p{Nd}\p{Pc}]**.
\W	Matches any non-word character, i.e. the negation of \w. For non-Unicode implementations, this is the same as **[^a-zA-Z_0-9]**. In Unicode categories, this is the same as **[^\p{Ll}\p{Lu}\p{Lt}\p{Lo}\p{Nd}\p{Pc}]**.
\s	Matches any white-space character. Equivalent to the Unicode character categories **[\f\n\r\t\v\x85\p{Z}]**. For non-Unicode implementations, \s is equivalent to **[\f\n\r\t\v]** (note leading space).
\S	Matches any non-white-space character. Equivalent to the Unicode character categories **[^\f\n\r\t\v\x85\p{Z}]**. For non-Unicode implementations, \S is equivalent to **[^ \f\n\r\t\v]** (note space after ^).
\d	Matches any decimal digit. Equivalent to **[\p{Nd}]** for Unicode and **[0-9]** for non-Unicode.
\D	Matches any non-decimal digit. Equivalent to **[\P{Nd}]** for Unicode and **[^0-9]** for non-Unicode.

Some examples of character classes in action are listed below.

Pattern	**Inputs (Matches)**
^b[aeiou]t$	Bat, bet, bit, bot, but
^[0-9]{5}$	11111, 12345, 99999
^c:\\	c:\windows, c:\\\\\, c:\foo.txt, c:\ followed by anything else
abc$	abc, 123abc, any string ending with abc
(abc){2,3}	abcabc, abcabcabc
^[^-][0-9]$	0, 1, 2, ... (will not match -0, -1, -2, etc.)

Character Escapes

Most of the important regular expression language operators are unescaped single characters. The escape character \ (a single backslash) signals to the regular expression parser that the character following the backslash is not an operator. For example, the parser treats an asterisk (*) as a repeating quantifier and a backslash followed by an asterisk (*) as the Unicode character 002A.

Escaped character	**Description**
ordinary characters	Characters other than . $ ^ { [(|) * + ? \ match themselves.
\a	Matches a bell (alarm) \u0007.
\b	The escaped character **\b** is a special case. In a regular expression, **\b** denotes a word boundary (between **\w** and **\W** characters) except within a **[]** character class, where **\b** refers to the backspace \u0008 character.
\t	Matches a tab \u0009.
\r	Matches a carriage return \u000D.
\v	Matches a vertical tab \u000B.
\f	Matches a form feed \u000C.
\n	Matches a new line \u000A.
\e	Matches an escape \u001B.
\040	Matches an ASCII character as octal (up to three digits); numbers with no leading zero are backreferences if they have only one digit or if they correspond to a capturing group number. (For more information, see Advanced Topics below.) For example, the character \040 represents a space (Decimal 32).

\x20	Matches an ASCII character using hexadecimal representation (exactly two digits). For example, the character `\x20` represents a space (Decimal 32).
\cC	Matches an ASCII control character; for example, `\cC` is control-C.
\u0020	Matches a Unicode character using hexadecimal representation (exactly four digits). For example, the character `\u0020` represents a space (Decimal 32).
\	When followed by a character that is not recognized as an escaped character, matches that character. For example, **\*** is the same as **\x2A**.

Atomic Zero-Width Assertions

The metacharacters described in the following table do not cause the engine to advance through the string or consume characters. They simply cause a match to succeed or fail depending on the current position in the string. For instance, ^ specifies that the current position is at the beginning of a line or string. Thus, the regular expression `^FTP` returns only those occurrences of the character string "FTP" that occur at the beginning of a line.

Assertion	Description
^	Specifies that the match must occur at the beginning of the string or the beginning of the line. For more information, see the **Multiline** option in Regular Expression Options.
$	Specifies that the match must occur at the end of the string, before **\n** at the end of the string, or at the end of the line. For more information, see the **Multiline** option in Regular Expression Options.
\A	Specifies that the match must occur at the beginning of the string (ignores the **Multiline** option).
\Z	Specifies that the match must occur at the end of the string or before **\n** at the end of the string (ignores the **Multiline** option).
\z	Specifies that the match must occur at the end of the string (ignores the **Multiline** option).
\b	Specifies that the match must occur on a boundary between **\w** (alphanumeric) and **\W** (nonalphanumeric) characters. The match must occur on word boundaries — that is, at the first or last characters in words separated by any nonalphanumeric characters.
\B	Specifies that the match must not occur on a **\b** boundary.

Sample Expressions

Marker String: "//[0-9]+/[0-9]+/[0-9]+"

Matches: "12/31/2001" or "The last date was 2/7/99" but not "1-31-01"

Field Formula: "/i/([0-9]+):([0-9]+) *[ap]m"

Returns: "9:45 PM" from “The time is 9:45 PM”

Field Formula: "/i1/([0-9]+):([0-9]+)"

Returns: "9" from “The time is 9:45 PM”

Field Formula: "/i2/([0-9]+):([0-9]+)"

Returns: "45" from “The time is 9:45 PM”

Pattern	Description
^\d{5}$	5 numeric digits, such as a US ZIP code.
^(\d{5})|(\d{5}-\d{4}$	5 numeric digits, or 5 digits-dash-4 digits. This matches a US ZIP or US ZIP+4 format.
^(\d{5}(-\d{4})?$	Same as previous, but more efficient. Uses ? to make the -4 digits portion of the pattern optional, rather than requiring two separate patterns to be compared individually (via alternation).

^[+-]?\d+(\.\d+)?$	Matches any real number with optional sign.
^[+-]?\d*\.?\d*$	Same as above, but also matches empty string.
^(20|21|22|23|[01]\d)[0-5]\d$	Matches any 24-hour time value.
/*.**/	Matches the contents of a C-style comment /* ... */

Advanced Topics

Three advanced regular expression features are lazy qualifiers, backreferences and lookaround processing. Since these advanced features are only needed on rare occasions, they are only briefly discussed here.

Lazy Quantifiers

In normal operation Quantifiers always the maximum number of characters. For example **b+** matches **abbbc**. With Lazy Qualifiers you can instruct the regular expression engine to match the minimum number of characters. For example **b+?** matches **abbbc**. The following table describes the lazy metacharacters that affect matching quantity.

Quantifier	Description
***?**	Specifies the first match that consumes as few repeats as possible (equivalent to lazy *).
+?	Specifies as few repeats as possible, but at least one (equivalent to lazy +).
??	Specifies zero repeats if possible, or one (lazy ?).
{*n*}?	Equivalent to **{n}** (lazy **{n}**).
{*n*,}?	Specifies as few repeats as possible, but at least *n* (lazy **{*n*,}**).
{*n*,*m*}?	Specifies as few repeats as possible between *n* and *m* (lazy **{*n*,*m*}**).

Backreferences

Backreferences are references to groups inside a regular expression. A common use of backreferences is within matching expressions themselves, such as this expression for finding repeated letters: **([a-z])\1**. This will match 'aa', 'bb', 'cc' and is not the same as **[a-z]{2}** or **[a-z][a-z]** which are equivalent and would allow 'ab' or 'ac' or any other two-letter combination. Backreferences allow an expression to remember things about parts of the input string it has already parsed and matched.

Lookaround processing

Lookaround processing refers to positive and negative lookahead and lookbehind capabilities. These constructs do not consume characters even though they may match them. Some patterns are impossible to describe without lookaround processing, especially ones in which the existence of one part of the pattern depends on the existence of a separate part. The syntax for each flavor of lookaround is described below.

Syntax	Description
(?=...)	Positive Lookahead
(?!...)	Negative Lookahead
(?<=...)	Positive Lookbehind
(?<!...)	Negative Lookbehind

One example of where lookaround processing is necessary is password validation. Consider a password restriction where the password must be between 4 and 8 characters long, and must contain at least one digit. You could do this by just testing **\d** for a match and using string operations to test the length, but to do the whole thing in a regular expression requires lookahead. Specifically positive lookahead, as this expression demonstrates: **^(?=.*\d).{4,8}$**

Ambiguity

If a regular expression could match two different parts of the input string, it will match the one which begins earliest. If both begin in the same place but match different lengths, or match the same length in different ways, life gets messier, as follows.

In general, the possibilities in a list of branches are considered in left-to-right order, the possibilities for "*", "+", and "?" are considered longest-first, nested constructs are considered from the outermost in, and concatenated constructs are considered leftmost-first. The match that will be chosen is the one that uses the earliest possibility in the first choice that has to be made. If there is more than one choice, the next will be made in the same manner (earliest possibility) subject to the decision on the first choice. And so forth.

For example, "(ab|a)b*c" could match "abc" in one of two ways. The first choice is between "ab" and "a"; since "ab" is earlier, and does lead to a successful overall match, it is chosen. Since the "b" is already spoken for, the "b*" must match its last possibility--the empty string--since it must respect the earlier choice.

In the particular case where the regular expression does not use "|" and does not apply "*", "+", or "?" to parenthesized subexpressions, the net effect is that the longest possible match will be chosen. So "ab*", presented with "xabbbby", will match "abbbb". Note that if "ab*" is tried against "xabyabbbz", it will match "ab" just after "x", due to the begins-earliest rule. (In effect, the decision on where to start the match is the first choice to be made, hence subsequent choices must respect it even if this leads them to less-preferred alternatives.)

Unicode Character Categories

The following Unicode Character Categories can be used with the \p{name} and \P{name} metacharacters. For example **\p{Cc}** matches a control character.

Unicode Category	Description
"Cc" (other, control)	Indicates that the character is a control code, whose Unicode value is U+007F or in the range U+0000 through U+001F or U+0080 through U+009F.
"Cf" (other, format)	Indicates that the character is a format character, which is not normally rendered but affects the layout of text or the operation of text processes.
"Cn" (other, not assigned)	Indicates that the character is not assigned to any Unicode category.
"Co" (other, private use)	Indicates that the character is a private-use character, whose Unicode value is in the range U+E000 through U+F8FF.
"Cs" (other, surrogate)	Indicates that the character is a high-surrogate or a low-surrogate. Surrogate code values are in the range U+D800 through U+DFFF.
"Ll" (letter, lowercase)	Indicates that the character is a lowercase letter.
"Lm" (letter, modifier)	Indicates that the character is a modifier letter, which is free-standing spacing character that indicates modifications of a preceding letter.
"Lo" (letter, other)	Indicates that the character is a letter that is not an uppercase letter, a lowercase letter, a titlecase letter, or a modifier letter.
"Lt" (letter, titlecase)	Indicates that the character is a titlecase letter.
"Lu" (letter, uppercase)	Indicates that the character is an uppercase letter.
"Mc" (mark, spacing combining)	Indicates that the character is a spacing character, which indicates modifications of a base character and affects the width of the glyph for that base character.
"Me" (mark, enclosing)	Indicates that the character is an enclosing mark, which is a nonspacing combining character that surrounds all previous characters up to and including a base character.
"Mn" (mark, nonspacing)	Indicates that the character is a nonspacing character, which indicates modifications of a base character.
"Nd" (number, decimal digit)	Indicates that the character is a decimal digit; that is, in the range 0 through 9.
"Nl" (number, letter)	Indicates that the character is a number represented by a letter, instead of a decimal digit; for example, the Roman numeral for five, which is 'V'.
"No" (number, other)	Indicates that the character is a number that is neither a decimal digit nor a letter number; for example, the fraction 1/2.
"Pc" (punctuation, connector)	Indicates that the character is a connector punctuation, which connects two characters.
"Pd" (punctuation, dash)	Indicates that the character is a dash or a hyphen.
"Pe" (punctuation, close)	Indicates that the character is the closing character of one of the paired punctuation marks, such as parentheses, square brackets, and braces.

"Pf" (punctuation, final quote)	Indicates that the character is a closing or final quotation mark.
"Pi" (punctuation, initial quote)	Indicates that the character is an opening or initial quotation mark.
"Po" (punctuation, other)	Indicates that the character is a punctuation that is not a connector punctuation, a dash punctuation, an open punctuation, a close punctuation, an initial quote punctuation, or a final quote punctuation.
"Ps" (punctuation, open)	Indicates that the character is the opening character of one of the paired punctuation marks, such as parentheses, square brackets, and braces.
"Sc" (symbol, currency)	Indicates that the character is a currency symbol.
"Sk" (symbol, modifier)	Indicates that the character is a modifier symbol, which indicates modifications of surrounding characters; for example, the fraction slash indicates that the number to the left is the numerator and the number to the right is the denominator.
"Sm" (symbol, math)	Indicates that the character is a mathematical symbol, such as '+' or '= '.
"So" (symbol, other)	Indicates that the character is a symbol that is not a mathematical symbol, a currency symbol or a modifier symbol.
"Zl" (separator, line)	Indicates that the character is used to separate lines of text.
"Zp" (separator, paragraph)	Indicates that the character is used to separate paragraphs.
"Zs" (separator, space)	Indicates that the character is a space character, which has no glyph but is not a control or format character.

Date and Numeric Format Strings

Date and Numeric format strings are used in the diverse File Outputs to set the output format of dates and numbers.

Date/Time Format Strings

Character	Description
(\\)	Display the next character in the format string. To display a character that has special meaning as a literal character, precede it with a backslash (\\). The backslash itself isn't displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\\\\). Examples of characters that can't be displayed as literal characters are the date-formatting and time-formatting characters (a, c, d, h, m, n, p, q, s, t, w, y, / and :), the numeric-formatting characters (#, 0, %, E, e, comma, and period), and the string-formatting characters (@, &, <, >, and !).
(:)	Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
c	Display the date as ddddd and display the time as ttttt, in that order. Display only date information if there is no fractional part to the date serial number; display only time information if there is no integer portion.
d	Display the day as a number without a leading zero (1 – 31).
dd	Display the day as a number with a leading zero (01 – 31).
ddd	Display the day as an abbreviation (Sun – Sat).
dddd	Display the day as a full name (Sunday – Saturday).
ddddd	Display the date as a complete date (including day, month, and year), formatted according to your system's short date format setting. For Microsoft Windows, the default short date format is m/d/yy.
dddddd	Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. For Microsoft Windows, the default long date format is mmmm dd, yyyy.
w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1-54).
m	Display the month as a number without a leading zero (1-12). If m immediately follows h or hh, the minute rather than the month is displayed.
mm	Display the month as a number with a leading zero (01-12). If m immediately follows h or hh, the minute rather than the month is displayed.
mmm	Display the month as an abbreviation (Jan-Dec).
mmmm	Display the month as a full month name (January-December).
q	Display the quarter of the year as a number (1-4).
y	Display the day of the year as a number (1-366).
yy	Display the year as a 2-digit number (00-99).
yyyy	Display the year as a 4-digit number (100-9999).
h	Display the hour as a number without leading zeros (0-23).
hh	Display the hour as a number with leading zeros (00-23).
n	Display the minute as a number without leading zeros (0-59).
nn	Display the minute as a number with leading zeros (00-59).
s	Display the second as a number without leading zeros (0-59).
ss	Display the second as a number with leading zeros (00-59).

ttttt	Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. For Microsoft Windows, the default time format is h:mm:ss.
AM/PM	Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Use the 12-hour clock and display a lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. For Microsoft Windows, the default format is AM/PM.

Examples:

The following are examples of user-defined date and time formats for December 7, 1958 8:50:35 PM

Format	**Display**
m/d/yy	12/7/58
d-mmm	7-Dec
d-mmmm-yy	7-December-58
d mmmm	7 December
mmmm yy	December 58
hh:mm AM/PM	08:50 PM
h:mm:ss a/p	8:50:35 p
h:mm	20:50
h:mm:ss	20:50:35
m/d/yy h:mm	12/7/58 20:50
\mmm\ddd	m12d07

Numeric Format Strings

The following table identifies characters you can use to create user-defined numeric formats. The default date format is "", i.e. no formatting is done.

CharacterDescription

Character	**Description**
(0)	Digit placeholder. Display a digit or a zero. If the expression has a digit in the position where the 0 appears in the format string, display it; otherwise, display a zero in that position. If the number has fewer digits than there are zeros (on either side of the decimal) in the format expression, display leading or trailing zeros. If the number has more digits to the right of the decimal separator than there are zeros to the right of the decimal separator in the format expression, round the number to as many decimal places as there are zeros. If the number has more digits to the left of the decimal separator than there are zeros to the left of the decimal separator in the format expression, display the extra digits without modification.
(#)	Digit placeholder. Display a digit or nothing. If the expression has a digit in the position where the # appears in the format string, display it; otherwise, display nothing in that position. This symbol works like the 0 digit placeholder, except that leading and trailing zeros aren't displayed if the number has the same or fewer digits than there are # characters on either side of the decimal separator in the format expression.

(.)	Decimal placeholder. In some locales, a comma is used as the decimal separator. The decimal placeholder determines how many digits are displayed to the left and right of the decimal separator. If the format expression contains only number signs to the left of this symbol, numbers smaller than 1 begin with a decimal separator. To display a leading zero displayed with fractional numbers, use 0 as the first digit placeholder to the left of the decimal separator. The actual character used as a decimal placeholder in the formatted output depends on the Number Format recognized by your system.
(%)	Percentage placeholder. The expression is multiplied by 100. The percent character (%) is inserted in the position where it appears in the format string.
(,)	Thousand separator. In some locales, a period is used as a thousand separator. The thousand separator separates thousands from hundreds within a number that has four or more places to the left of the decimal separator. Standard use of the thousand separator is specified if the format contains a thousand separator surrounded by digit placeholders (0 or #). Two adjacent thousand separators or a thousand separator immediately to the left of the decimal separator (whether or not a decimal is specified) means "scale the number by dividing it by 1000, rounding as needed." For example, you can use the format string "##0,," to represent 100 million as 100. Numbers smaller than 1 million are displayed as 0. Two adjacent thousand separators in any position other than immediately to the left of the decimal separator are treated simply as specifying the use of a thousand separator. The actual character used as the thousand separator in the formatted output depends on the Number Format recognized by your system.
(:)	Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
(E- E+ e- e+)	Scientific format. If the format expression contains at least one digit placeholder (0 or #) to the right of E-, E+, e-, or e+, the number is displayed in scientific format and E or e is inserted between the number and its exponent. The number of digit placeholders to the right determines the number of digits in the exponent. Use E- or e- to place a minus sign next to negative exponents. Use E+ or e+ to place a minus sign next to negative exponents and a plus sign next to positive exponents.
- + $ ( )	Display a literal character. To display a character other than one of those listed, precede it with a backslash (\) or enclose it in double quotation marks (" ").
(\\)	Display the next character in the format string. To display a character that has special meaning as a literal character, precede it with a backslash (\). The backslash itself isn't displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\\). Examples of characters that can't be displayed as literal characters are the date-formatting and time-formatting characters (a, c, d, h, m, n, p, q, s, t, w, y, / and :), the numeric-formatting characters (#, 0, %, E, e, comma, and period), and the string-formatting characters (@, &, <, >, and !).
("ABC")	Display the string inside the double quotation marks (" "). To include a string in format from within code, you must use Chr(34) to enclose the text (34 is the character code for a quotation mark ("")).

Delimiter Character Definitions

The following formats can be used to define a Delimiter character, a Text Qualifier string, and the LineBreak multiline Format property:

Format	Description
{NONE}	Nothing (not valid as delimiter character)
{TAB}	Tab (ASCII code 9)
{SPACE}	Space
{NULL}	Null character (ASCII code 0)
{COMMA}	Comma (,)
{DBLQUOTE}	Double quote (“)
{SGLQUOTE}	Single quote (“)
{COLON}	Colon (:)
{SEMICOLON}	Semicolon (;)
{x}	With x a decimal number between 0 and 255, defines a character with ASCII code x. For example {32} is a space.

Visual Basic Functions (Add-in Versions Only)

Introduction

Import Wizard exposes two Visual Basic for Applications (VBA) functions that can be used to automate your application. The functions are: ImportWizardGo() to start an import, and ImportWizardStart() to show the main Import Wizard window. The function are accessed as methods of the **Application.COMAddIns("IW9DLL.ImportWizardAddIn").Object** object. It is probably the easiest to define a new object variable and use this variable to access the functions:

```
Dim iw As Object
Set iw = Application.COMAddIns("IW9DLL.ImportWizardAddIn").Object
iw.ImportWizardStart
```

ImportWizardGo Function

Function ImportWizardGo(ModelFile as String, Optional SourceFile as String, Optional DestTable as String, Optional TableAction as String, Optional RecordAction as String) as Long

This function runs the import defined in ModelFile. Calling the function with a value for any or all of the four optional arguments (SourceFile, DestTable, TableAction, RecordAction) will override the setting in the model. **TableAction** and **RecordAction** can be shortened to a single character, for example "F" is the same as "First Update, then Insert".

TableAction	Description
"Append"	If the table exists new records will be appended to it, the table structure is not modified not even if the import model defines other fields than exist in the table. If the table does not exist it is created.
"Drop"	The table is recreated on every import. An existing table will be deleted (dropped).
"Empty"	If the table exists it all records in the table will be deleted before the import starts. If the table does not exist it will be created.

RecordAction	Description
"Insert"	The record is appended to the end of the table.
"Update"	Existing records in the table are updated with the imported record based on the "Update Key" fields.
"First Update, then Insert"	Combination of "Update" and "Insert". First the existing records are updated with the imported record (based on the "Update Key" Fields). If no existing record was updated then the imported is appended to the end of the table.

Return values:
0: Import completed without warnings or errors
1: Import completed with warnings.
2: Import failed, an fatal error occurred.

Example: import the sales2005.txt file using the Sales.iwm model and display a message if import had warnings and/or errors.

```
returnvalue = ImportWizardGo("Sales.iwm", "sales2005.txt")
If returnvalue > 0 then Msgbox "Import problems…"
```

ImportWizardStart Function

Function ImportWizardStart(Optional (ModelFile as String, Optional SourceFile as String, Optional DestTable as String, Optional TableAction as String, Optional RecordAction as String) as Long

This function opens the main Import Wizard window. If the ModelFile argument is set, the model will be opened but the import will not be started. Calling the function with a value for any or all of the three remaining optional arguments (SourceFile, DestTable, Append) will override the setting in the model.

Return values:
0: Ok
2: Failed, an fatal error occurred.

Command Line Arguments (Pro Version Only)

Command line arguments can be used to steer the import process with an external program. The command line arguments are:

impwiz modelfile[.iwm] [/O] [/L logfile] [/S sourcefile(s)] [/D database(file)] [/T tablename] [/A Drop|Empty|Append] [/A Insert|Update|First update then insert] [/Q]

The text between square brackets [] is optional. The command line argument are not case sensitive and a space between a switch and the following argument is optional. When an argument is specified on the command line it overrides the corresponding settings in the modelfile, if an argument is ommitted then the settings as defined in the modelfile are used. Note that the modelfile has to be the first argument.

Command Line Arguments

Argument	Description
/?	Displays command syntax in a popup window.
/O	Open modelfile, do not import.
/L	Sets the logfile, or switches logging off when /L alone is used.
/S	Sets the sourcefile(s) to import.
/D	Sets the output database name or output filename. Note: If this switch is used on a .iwm model file that does not have an output database the import is cancelled with exit code 2 and a popup window shows an error message.
/T	Sets the output table name. Note: If this switch is used on a .iwm model file that does not have an output table the import is cancelled with exit code 2 and a popup window shows an error message.
/A	Sets Table and Record Actions. The /A switch can be specified twice to set both Table Action and Record Action. The names of the actions can be abbreviated to a single character: /AC Confirm Create Table /AD Drop (Delete) Table /AA Append to Table /AE Empty Table /AI Insert Records /AU Update Records /AF First Update then Insert Records. Note that not all outputs support all table and record actions.
/Q	Quiet mode: no popup messages are generated on errors and during an import the progress dialog is not shown. In Quiet mode the only indication of an error is an exit code 2.

Examples

impwiz /? Show command line syntax.

impwiz somemodel /O opens modelfile "somemodel.iwm" but does not run it.

impwiz mdl /O /Sc:\import\*.* opens modelfile "mdl.iwm" with source files "c:\import\*.*" but does not run the model.

impwiz c:\models\model.mdl performs an import as defined in model file "c:\models\model.mdl", note the non-standard file extension.

impwiz test.iwm /s myfile.txt /t mytable /aa /ai /q runs model "test.iwm", the file "myfile.txt" is imported to "mytable" using table action "Append" and record action "Insert" without showing any windows.

Exit Codes

Exit Code	Description
0	Import completed without warnings or errors.
1	Import completed with warnings. The warnings are logged in the logfile.
2	Import failed. If you used the /Q flag, run impwiz again without the /Q flag to get an error message for invalid arguments. For import errors open the log file and look for 'E' error codes.

The exit code can be used in your application or in a batch file for error handling. For example, the following batch file runs an import and prints out "Import OK", "Import Warnings", or "Import Failed" based on the result of the import (see "iwdemo.bat" in the Samples directory).

```
@echo off
echo Starting Import Wizard ...
"c:\Program Files\Import Wizard\impwiz.exe" "sample1 fixed.iwm"
IF ERRORLEVEL 2 GOTO fail
IF ERRORLEVEL 1 GOTO warn
echo Import OK
GOTO end
:warn
echo Import Warnings
GOTO end
:fail
echo Import Failed
:end
pause
```

Import Algorithm and Limitations

Import Algorithm

Import Wizard will use the model definition in the following way to parse the source file into a table.

Step 1: Read next line from the source file. If the line matches a Discard marker then repeat Step 1, else continue with Step 2.

Step 2: Match line with markers. The line from Step 1 is matched against the first marker. If the marker matches the line then the fields defined for that marker are read into a temporary import record. If the marker does not match then the next marker is processed. Upon finding a match for a Record marker the following three steps occur:

i. The fields for the record marker are read into the temporary import record.
ii. The formulas are evaluated. Evaluation occurs in a single pass so that circular references are not possible.
iii. The temporary import record is appended to the import table.

When all markers are processed the import continues with Step 1, the import is finished when all lines from the import file have been processed.

Note: markers are processed in the order that they appear in the model.

Performance Tips

The following performance tips are intended to optimize large imports that take many minutes to run. For smaller imports the time gained on the import will probably not pay off against the time spent optimizing the import.

1. Output to delimited files, not to the database directly. For large imports into databases consider importing into a delimited file and then use the bulk insert function of the database to load the delimited file into the database. Examples of bulk insert functions are the "BULK INSERT" command or the bcp program for SQL-Server and the "LOAD DATA INFILE" command for MySQL.

2. Use Insert, not Update. If you need to update a table with a large dataset, import this large dataset into a separate table with insert mode. Then use a SQL UPDATE statement to update the final table with the imported table. This is faster because instead of running an UPDATE statement for each row a single UPDATE statement is used. This requires of course that your database supports this.

3. Use indexes with Update. Make sure that the 'Update Key' columns are indexed in the database. This will give a performance boost if the table to be updated contains many records.

4. Use Field Properties wisely. Field Properties such as 'Convert To', 'Delimiter' and 'Formula' are highly optimized for speed but still consume processing time. Remove unused properties, for example on a 1.5GHz Pentium M the 'Convert To Date' function requires 10 to 20 seconds for 1 million conversions, if the date is already correctly formatted you can gain these 10-20 seconds by setting 'Convert To Text'.

5. Minimize the number of markers. Don't create new markers for fields that always occur at a fixed number of rows below or above an existing marker.

Limitations

Number of records: Not limited by Import Wizard, limited by the database system used. For example, imports into Excel are limited by the maximum number of rows allowed in a spreadsheet.

Source file size: Unlimited for fixed, delimited, excel, and xml imports. For these import types only a small part of the file is read into memory at any time, thus allowing imports of files over 2GB in size. Only the HTML import type loads the whole file into memory, and is therefore limited by system memory size.

Source file line length: Up to 32767 characters for fixed and delimited imports. Unlimited for HTML imports.

Number of fields: Unlimited.

Number of markers: Unlimited.

Multiline fields: Up to 255 lines.

Field Line offset: Line offsets allowed between -255 and +255 lines.

Model Design: The first 10000 lines of the file are displayed.

Discard Markers: Limited to single line markers. If the discard marker contains more than one line, only the first line is used.

Discard Marker Line offset: Line offsets allowed between -255 and +255 lines.

Output Limitations: See Outputs for limitations specific to an output type.

Import Error and Warning Messages

During an imports problems might occur, for example when a field can not be converted into a specific data type. These import problems are listed in the import window and are logged to the log file (if specified). There are two types of import problems: **Errors** and **Warnings**. Errors cause the import to fail, they are identified by a code starting with an **E**. Warning indicate a problem with the import, but do not cause the whole import to fail, warnings are identified by a code starting with a **W**.

Below all Errors and Warning messages are listed in numeral order.

Source File Errors

E101: Source file name not supplied.
The source file was not supplied in the model. Supply a source file name.

Output Errors

E121: Database connection failed. <Error Description>
Import Wizard was unable to connect to the destination database.
Actions:
1) Check that all required components are installed, see Installation.
2) Check that the entries in the main screen are correct.
3) Check that the database is not exclusively locked by another user, or read only.
4) When using a ODBC database check that the connect string is correct, common mistakes include incorrect passwords, or user accounts with insufficient rights. See Database Types (Pro Version Only).

E122: Output table connection failed. <Error Description>
The output database could be opened but the output table couldn’t.

E123: Table “<tablename>“ not updateable.
The table could not be created because it is marked as read-only.

E124: Cell or range '<Rangename>' not found in output Excel file.
The Excel spreadsheet output cell/range was not specified or incorrect, in doubt use "A1" as Rangename as this should always work. Correct the entry and try again.

E125: Unable to create output table '<tablename>', please check model or create table manually.
There was a general error trying to create the table. Check for illegal characters in tablename and fieldnames. For ODBC connections check the output Datatype properties.

E126: Could not open/create output file '<Filename>'. <Error Description>
The output file defined in the model could not be opened/created. Check that the name is valid and that you have sufficient rights to open/create the file.

E127: Output Excel file '<Filename>' is readonly, or currently opened in Excel.
Could not open the output Excel for writing. If the file is opened in Excel then close the file in Excel.

E128: Error while saving output Excel file '<Filename>'. <Error Description>
An error occured while saving the output Excel spreadsheet, see error description for further details.

E129: Error building SQL statement. <Error Description>
An error occures while building the SQL insert or update statements, see error description for further details.

E130: Delete of existing table not confirmed, import aborted. Change Table Action setting and try again.
The Table Action setting is set to "Confirm Create New Table" and the table exists. The user did not confirm deleting and recreation of the table.

Field Errors

E141: The import model has no fields.
No fields are defined in the model. Add at least one field to the model and try again.

E142: The output table exists but does not contain any of fields defined in the model, nothing imported.
Import Wizard does not modify exiting tables. If none of the fields defined in the model are present in the output table then the import fails. Rename the fields in the model or set the Set Table Action to 'create table' to create the table.

E143: Record Action Error: 'Update' specified but without any Update Key fields, either select Insert as import action or set the Update Key field property for at least one field.
When the 'Update' or 'First Update then Insert' record action is used you should set the Update Key property for at least one field to True.

E144: Update Record Action specified but update key field '%1%' is missing in the output table. Either select Insert as import action or unset the Update Key field property for this field.
All Update Key fields are required to exist in the output table to guarantee correct updates. Either remove the indicated field from the model or add it to the output table.

Source File Warnings

W201: Source file '<Filename>' not found.
Upon trying to import using the model Import Wizard couldn't find the source file. Supply the correct file name.

W202: Error while opening file '<Filename>'. <Error Description>
An error occured upon opening the file. See error description for details

W203: Could not open import range '<Rangename>' in input Excel file '<Filename>'.
The Range could not be found in the Excel source file.

Output Warnings

W221: Error while appending record. Field Name='<Fieldname>' Value='<Field Value>' Error=<Error Description>
A general error occurred while attempting to append (insert or update) a record to the destination table, the record is not appended.

W222: Record insert failed. <Error Description>
An error occured while inserting a record in the output table. See error description for details.

W223: Record update failed. <Error Description>
An error occured while updating a record in the output table. See error description for details.

W224: Failed to delete existing records from output table '<Table Name>'. <Error Description>
The output table contains records and you specified 'Empty' as Table Action, Import Wizard failed to delete the records from the table.
Action: in the output set the Table Action to 'Append' or 'Create'.

W225: Table Action Error: 'Create' specified but unable to drop (delete) the existing output table failed, using existing table.
The Import Action was set to 'Create' but the existing table could not be dropped (deleted). The existing table is used for the import.

W226: Error while closing output Excel file. <Error Description>
An error occured upon closing the output Excel file. See error description for details.

W227: Error executing SQL statement. <Error Description> SQL=<SQL>
An error occured executing the pre- or post-import SQL statement(s). See error description for details.

Field Warnings

W241: Field '<Field Name>' not found in output table, field ignored.
The output table exists but does not contain the <Field Name> field. Imports to this field will be ignored.
Action: Import Wizard does not modify existing tables, either modify the destination table in your database to include the field or set Table Action to 'Create' to create a new table.

W242: Unable to create field '<fieldname>'.
The field could not be created. Check for illegal characters in the fieldname.

W243: Numeric conversion error in field '<Field Name>' with fractional mark '<Method String>', the imported data to convert was '<Import Data>'.
Import Wizard is unable to convert <Import Data> to a number using the conversion method specified for the field. Correct the conversion method property in the model and try again.

W244: Date/time conversion error in field '<Field Name>' with date format '<Method String>', the imported data to convert was '<Import Data>'.
Import Wizard is unable to convert <Import Data> to a date using the conversion method specified for the field. Correct the conversion method property in the model and try again.

W245: Filter compile error. Ignoring Filter, importing all records. <ErrorDescription>
The Filter contains an error. The Filter will be ignored and all records will be imported. Edit the Filter to fix this problem.

W246: Formula compile error, ignoring Formula. <ErrorDescription>
The formula contains an error. The formula will be ignored. Edit the Formula to fix this problem.

W247: Error in filter. Record not imported. <ErrorDescription>
Upon executing the filter expression an error occured. The record is not imported. Edit the Formula to fix this problem.

W248: Error in formula '<Formula>' of field '<FieldName>'. <ErrorDescription>
Upon executing the formula defined for this field contains an error occured. The formula will be ignored. Edit the Formula to fix this problem.

W249: Error in regex formula '<Formula>' of field '<FieldName>'. <ErrorDescription>
Upon executing regular expression defined for this field contains an error occured. The regex formula will be ignored. Edit the regex Formula to fix this problem.

W250: Undefined marker number '<MarkerNo>' in field '<FieldName>'.
The field references an undefined marker number, the field will not import data from the source file. Either create a new marker for the missing marker number or change the Marker No property of the field.

W251: Invalid marker number. Defined in multline field: '<FieldName>', Marker no: '<MarkerNo>', Multi Line End: '<MultiLineEnd>'
The marker referenced in the multiline end property of field fieldname is invalid. Edit the Multiline End property to correct this problem.

W252: Multiline field '<FieldName>' truncated to <n> lines.
The multiline end condition for the field was not met before reaching n lines. The only the first n lines of the multiline field will be imported. Check the multiline end condition, see also Limitations.

W253: No fields defined for Marker '<Marker Number>'.
The indicated marker has no effect because no fields are defined for this marker. Delete the marker or create at least one field for this marker

Other Warnings

W281: Import cancelled by user.
The import was aborted by pressing the cancel button in the progress dialog.

W282: This software is unregistered, ONLY PART OF THE FILE(S) WILL BE IMPORTED.
You are using the unregistered demo version of Import Wizard. The demo version imports up to 30 records to demostrate that Import Wizard is capable of importing into the selected output database, spreadsheet or file. You can use the 'Preview' option to view the full imported table without actually importing it to the output database. If both the import of the first 30 records and the preview work correctly, then the full import will work correctly in the registered Import Wizard.
Imports of unlimited size are possible with the registered version, see Ordering Software, Registration.

W284: More messages received: see log file for details.
Additonal warning/error messages were generated but are not shown here, see log file for full details.

W285: Automation action failed: '<ActionDescription>' <ErrorDescription>
Could not perform the requested automation action, read the ErrorDescription for details.

Install / Uninstall

Install

1) It is optional to uninstall previous versions of Import Wizard before installing an update. To update a previous version 9 to the current version 9 simply download the latest version from www.beside.com and run the setup program without unstalling the previous version 9, this will keep your settings including registration key.

2) Run the iw9setup.exe program.

3) Follow the prompts to install the software in a directory.

4) To start the program: use the Start menu: *Start->Programs->Import Wizard*. The Access and Excel add-in versions can be started from within Access/Excel: open any database or spreadsheet and select from the Access/Excel menu *Tools->Import Wizard*.

Note: Import Wizard requires the Microsoft .NET Framework 2.0 or later. You can install the framework after you installed Import Wizard.

Note for Import Wizard Pro only: Import Wizard Pro uses Microsoft Data Access Components (MDAC) in combination with the Jet Database Engine version 4.0. If Office 2000 or later (including Access) is installed on your system, then you should have these components installed already. If not, or if you want to update MDAC to the latest version, you can download them directly from Microsoft:
MDAC information: http://www.microsoft.com/data
MDAC download: http://msdn.microsoft.com/data/downloads/updates/default.aspx#MDAC
Jet 4.0: http://support.microsoft.com/support/kb/articles/q239/1/14.asp

Uninstall

Note: Uninstall removes the registration code from your computer, make sure that you have a copy of the registration code in order to re-install the software at a later time.

To uninstall Import Wizard: Run *Start->Programs->Import Wizard->Uninstall* from the Windows Start menu. Alternatively you can select menu *Start->Settings->Control Panel->Add/Remove Programs* and remove the "Import Wizard" entry from the list by clicking the "Add/Remove" button.

After running uninstall you can delete the directory where you installed the software to remove files that were created after the software was installed.

Ordering Software, Registration

This is not free software. You may not use this software beyond the initial 30 day evaluation period unless you purchase a registration code for the software.

Upon receipt of your registration fee you will be sent a registration code which can be recorded in your copy of the software. The registration code need to be entered into the Import Wizard software via the registration screen. Press the Enter Registration Code button or use menu Options->Registration Code to get to the registration screen. **Once the registration code is entered the software will indicate that the copy is registered and imports of any size are possible. (Only limited by the capabilities of the database system you are using)**

Registered users can obtain, when available, version 9.x.x updates of the software free of charge from our web site: www.beside.com. Technical support is provided via email to impwiz@beside.com for the period of one year after the purchase date.

How to order?

Credit Card

You will have your registration code within minutes! Go to www.beside.com/order.html and place your secure online Visa, MasterCard, Amex, or Diners transaction. Ordering by credit card is a fully automated process, the registration code is send out immediately after the credit card transaction has been accepted.

Wire Transfer, Check or International Money Order

Go to www.beside.com/order.html and enter the number of licenses you wish to order. On the next page, indicate the payment method you wish to use. You will receive an invoice and payment instructions. The registration code is send out as soon as payment is received for this invoice.

Fax

Go to www.beside.com/order.html and enter the number of licenses you wish to order. On the next page, indicate that you wish to order by Fax. You will be given a preformatted fax sheet to print out and the fax number to send your order to. After you send the fax order you will receive the invoice for your order. The registration code is send out as soon as payment is received for this invoice.

Purchase Order

Please place your order by Fax as described above. You will receive an invoice for use with your Purchase Order. The registration code is send out as soon as payment is received for this invoice.

Phone

Please note that online orders are processed automatically and therefore more quickly than orders placed by phone, mail or fax because processing is not dependent on our order processing center's business hours.

If you would still prefer to place an order using one of these options, please include the following information:

- **Product ID: 300 065 359**
- Product Name: Import Wizard 9
- The quantity you wish to order
- The name the product license will be issued to
- Your billing address and your delivery address, if different
- Your phone number and your fax number, if available
- The e-mail address to which the order confirmation and invoice should be sent, and your e-mail delivery address, if different
- Your selected payment option and currency

If you include all of the required information, your order can be processed immediately by our order processing team.

The addresses below are for ordering only and do NOT provide technical support.
For technical support: email to impwiz@beside.com

Order Processing USA
share-it!
9625 West 76th Street, Suite 150
Eden Prairie, MN 55344, USA

Phone: 1-800-903-4152 or 1-952-646-5747 (English)
Fax: 1-952-646-4552

Order Processing Germany
share-it! - element 5
Vogelsanger Strasse 78
50823 Cologne, Germany

Phone: +49 221 31088-20 (German, English, French, Italian, Spanish, Portuguese)
Fax: +49 221 31088-29

Registration Fees

The registration fee for Import Wizard 9 covers Import Wizard Pro and all add-in versions.

Single User License: US$ 149

For Multi-User, Site, SDK, or Source Code licenses pricing please visit www.beside.com

See Copyrights, Warranty, and License Agreement for a full description of the licensing conditions.

Services

We can help you build a data transformation solution by programming an Import Wizard Model for you. Please email a sample source file and a description of the requirements to impwiz@beside.com and we will make you a qoute. We usually have a solution ready within 1 business day, please visit www.beside.com for up to date information.

Copyrights, Warranty, License Agreement

You should carefully read the following terms and conditions before using the software.

COPYRIGHTS

Import Wizard version 9, including the Import Wizard Pro, Import Wizard Access Add-In, and Import Wizard Excel Add-In, (the "Software") is owned by Beside Software and is protected by copyright laws and international treaties.

LICENSE AGREEMENT

Use of Unregistered Shareware Version

You are hereby licensed to:

- Use the unregistered shareware version of the Software for a 30 day evaluation period,
- Make as many copies of the unregistered shareware version of the Software as you wish,
- Give exact copies of the unregistered shareware version of the Software to anyone, and
- Distribute the shareware version in its unmodified form via electronic or other means.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made.

Evaluation and Registration

This is not free Software. You may not use this software beyond the initial 30 day evaluation period unless you register the Software. See Ordering Software, Registration

Use of Registered Version

Registration of the Software grants you a non-exclusive right to load and use the Software. The Software is "loaded" onto a computer when it is either copied to the permanent memory of the computer (the hard disk) or loaded into the temporary memory (RAM), such as when a computer runs the software from a network server, or from removable disk media.

Single User License

You may use the Software only on a single computer, or on up to two computers used exclusively by the same individual user.

Multi User License

The Software can be loaded in such a way that the number of users is less than or equal to the number of licenses owned. A user is any person that has access to the licensed software, regardless whether this person is actually using the software.

Other Restrictions

You may not transfer your rights under this agreement. You must not rent or lease this Software. You may not decompile or disassemble the Software. You may not distribute the registered version of the Software.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR
