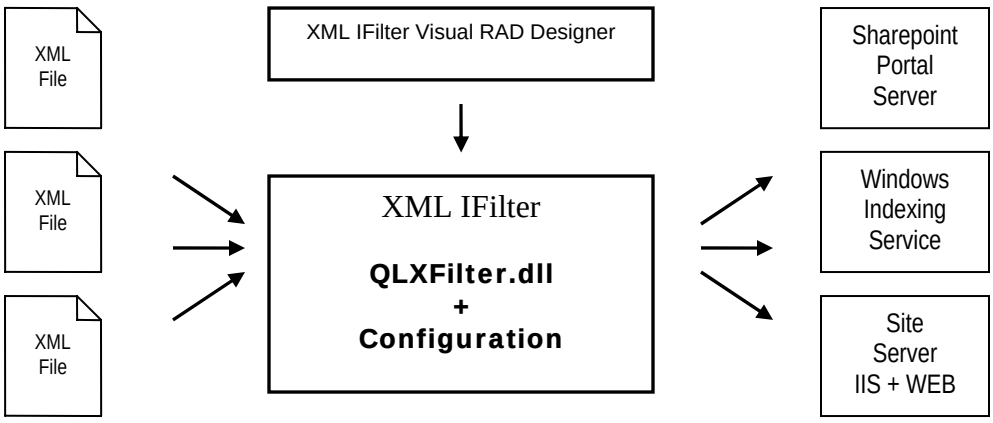
XML IFilter

for easy XML file indexing

White Paper

www.quilogic.cc



QuiLogic's XML IFilter enables crawling of documents containing xml based data. For the first time ever, our filter technology enables users to search and index arbitrary structured xml files based on content. Likewise VISIO® or Excel® files can be saved as xml files and indexed with the help of QuiLogic's XML IFilter!

XML IFilter extends the standard functionality of Microsoft Internet Information Server (IIS), Microsoft Share Point Portal Server (SPS), and all other products based on top of Microsoft Indexing Service Technology.

XML IFilter contains a Rapid Application Development (**RAD) tool** which let you **visually design and test** your indexing application. No coding is required. The RAD tool takes care to generate all necessary registry entries and definition files for you.

Supported Systems:

XML IFilter can be used with the following products:

- Microsoft Share Point Portal Server®
- Microsoft Share Point Team Services®
- Microsoft Windows® Indexing Service
- Microsoft Site Server®
- All other products based on Microsoft Indexing Technology

Technical Requirements

XML IFilter is self-contained and requires no other additional products to install.

The following information is extracted from an xml file:

- Name and text content of any element.
- Name and text content of any attribute.
- Date, time, numeric and boolean content from any element and attribute.
- Recognizes all usual character sets used by xml files (UTF8, UTF16 …).

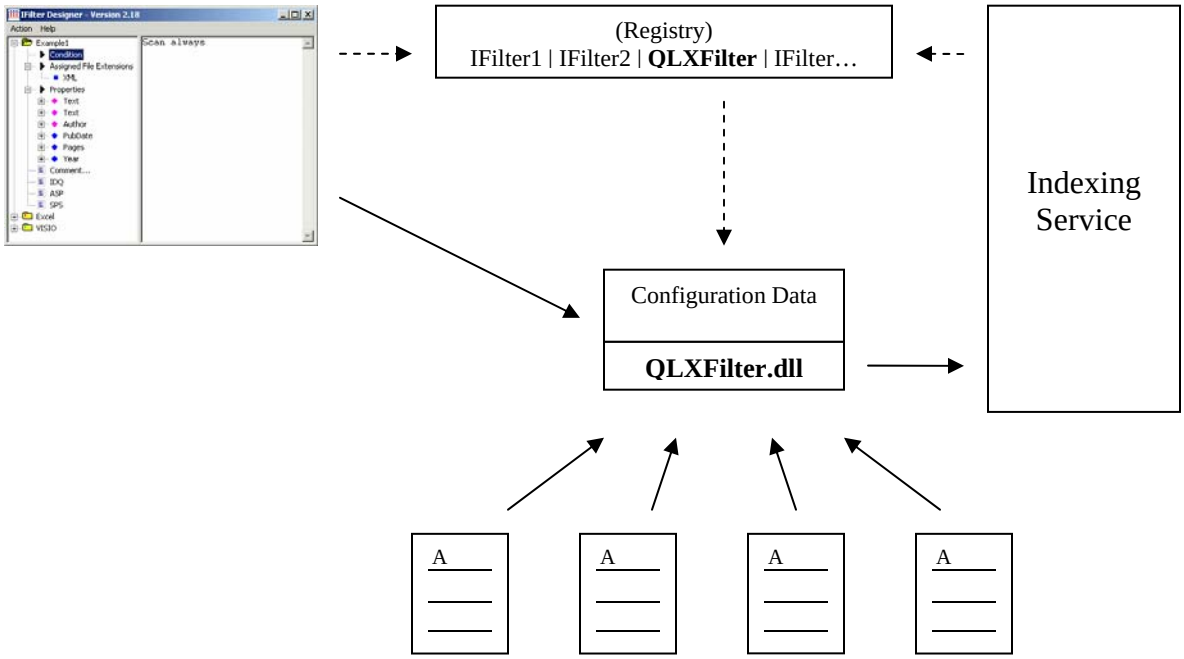
The included RAD tool let you visually design and specify which data to extract for each returned property. It might be either the content of a single element or any combination of attribute and element values including the names of elements in any order.

Features

- Build in “Probing Technology”. Enables crawling and indexing of multiple, arbitrary structured xml files with totally different content but having the same file name extension.
- Build in facility to “normalize” the returned property text to a common string. A typical example might be the indexing of telephone numbers as shown below.
- **343-6790-555** or **#343-6790-555** or ‘**343 6790 555**’ can be normalized to: **3436790555**
- Assign more than one file extension to “common structured” XML files.
- Indexing can be fine tuned by parameters for controlling exactly what data to index from your XML files.
- Option Switch to ignore <XML> or <HTML> tags during indexing.
- Visual RAD Designer included for developing indexing solutions without coding.
- Designer takes care to set the required registry entries as needed.
- Designer creates required property definition files for SPS, ASP and WEB search applications.
- Build in facility to create “Abstracts” from totally different locations in the XML Data.
- Precisely specify and fine tune the returned content for “full text” indexing.
- Test-Mode available to check the data outcome of your indexed properties.
- Installation program included for quick and easy setup.
- Enterprise edition contains full C++ Source code for the IFilter implementation.
- Debugging Aid available for better troubleshooting your indexing application.
- Ultrahigh performance, makes the indexing of even hundreds of thousands XML documents a snap.

Operation Principle

XML IFilter is a dynamic link library (dll) file that provides a bridge between any Microsoft indexing client and files containing the xml data. When an indexing client needs to index content from documents it will look in the registry for an appropriate filter dll based on the type of the document file name extension (.doc, .xml …). By using the included RAD tool, users can visually specify the file extensions assigned as well as the content and properties to extract. That information is written into a configuration file to drive the filter dll at indexing time and to update the registry based association between document extensions and indexing filters.



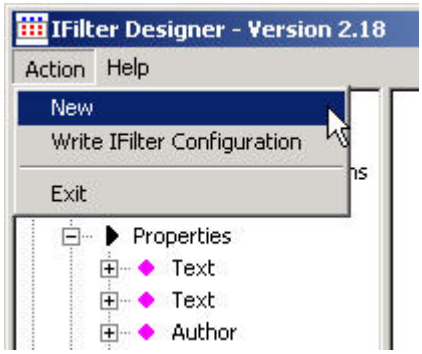
The Design tool let you create multiple filter descriptions. Each description is assigned an individual indexing condition. The condition is used by the filter dll to "probe" an xml file during the file scan. The file is indexed with a particular filter description only when the probing passed successfully the condition. In this respect, it is possible to scan xml files having the **same extension**, but totally **different xml content** and structure !

The user assigns each filter description a set of individual properties to be retrieved during the xml document scan. The data content of the returned property is individually laid down by the user, taking into account the structure of the underlying xml document.

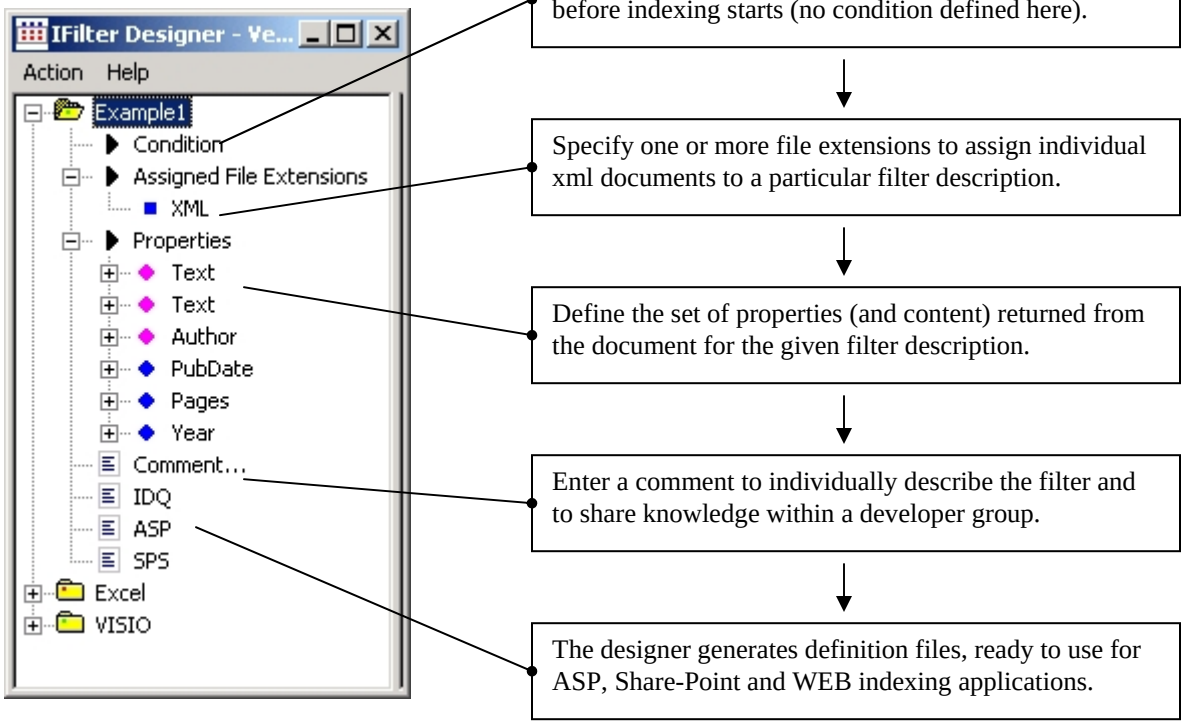
With the above described methods it is possible to fine tune the indexing of *any* xml based document with individual internal structure and file name extension.

4 Simple Steps to Setup XML Indexing

1. Create a new filter description

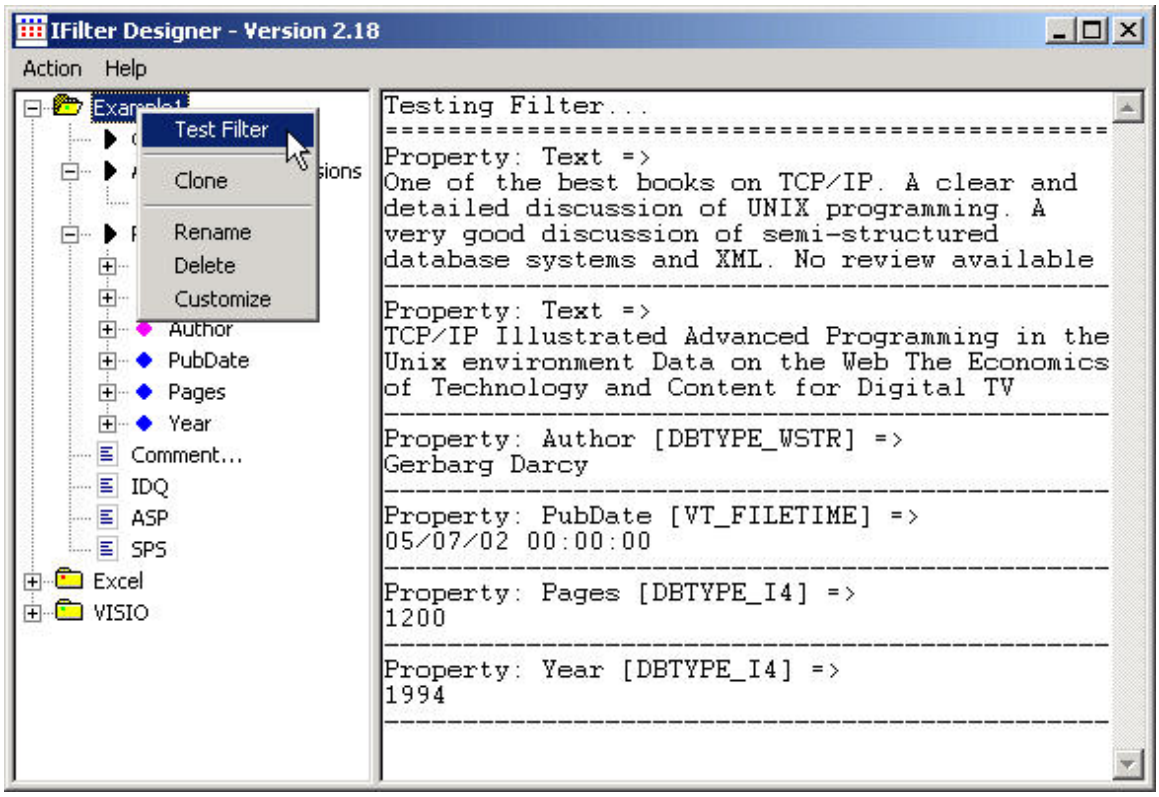


2. Edit filter description

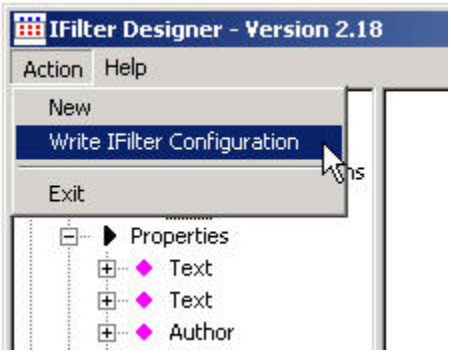


During indexing, the XML IFilter (QLXFilter.dll) is called file by file from the Indexing Service. The filter applies *all* defined filter-descriptions one by one, probing individually and returning the property data for which a match was found. Each property again has a query condition assigned which must be true to return any content for that property. Properties can be selected from a set of predefined standard properties (content, author, title …) or individually defined to match the user requirements.

3. Test filter



4. Deploy



Within the last step, the filter configuration is written to the configuration file, the registry updated and the indexing service stopped and restarted. At this point, full-text queries and SPS dashboard site simple search (‘contains’ and ‘freetext’) will work.

Query Technology Inside

Behind the scene QuiLogic's XML Database and Query Technology provides the heart beat to drive the QLXFilter indexing engine. Even multi megabyte sized xml files with arbitrary complexity and tree depth can be indexed fast and easy.

QuiLogic provides a visual RAD tool which enables the user to specify exact the individual content to be retrieved for each property. To do so, the user formulates simple queries which describe the property data that should be returned. A simplified version of XQuery (a W3C draft specification for querying xml data; see references) is used.

For example, to return the content of *all* xml attribute tags, use the following simple XQuery statement for the content property:

```
For $X in FILE//@* Return $X/text();
```

… or to query an Author property use:

```
For $X in FILE/abc/author WHERE … Return $X/text();
```

XQuery includes the well known XPath language (see References) to select sub trees out of a larger xml tree. In the return clause, you specify what to return. To return data types other then text (number, bool, date, time), a simple data type specification is appended to the referenced variable in the return statement (`/bool(), /text(), /date()`…).

The design tool let you assign individual queries for all defined properties and in this way you specify the returned content. At run time the indexing engine applies these queries to retrieve the associated content. This works very fast, because during file load a special index is build, taking into account the predefined queries.

XML content of any complexity can be indexed by this method. Even numeric data, build up from the content of several elements, can be indexed due to the possibility to apply aggregate functions like SUM, AVG, … on the XQuery statement.

QuiLogic has developed another product, **SQL/XML-IMDB** a universal in-memory database engine which (to our knowledge) is the only software product available on the marked for managing sql and xml based data within one component. SQL/XML-IMDB is a combined native SQL and XML database and part of QuiLogic's information integration strategy to unify structured and unstructured data from sources such as relational databases, xml documents, flat files and Web services (SOAP).

The engine is available for NET, VB, Delphi, C++ and Perl from QuiLogic. A white paper can be downloaded from our website, which describes the database in more detail. (see next section).

Availability

XML IFilter is available as:

- 1 Machine license.
- 4 Machine license.
- Enterprise edition for an arbitrary number of machines.

The Enterprise edition contains the source code of the IFilter implementation either as Visual Studio 6.0 c++ or as VC8.0 c++ project. This enables any developer in the Enterprise to make custom modifications for the IFilter dll to satisfy special requirements not available in the original product.

To order please visit www.quilogic.cc

QuiLogic, Inc. is an IT company headquartered in central EU, Austria. Founded in 1995, today QuiLogic creates innovative products and offers exceptional expertise in all sort of data management projects, high performance demanding applications and xml based solutions.

References

XQuery:
www.w3c.org/xml/query.html

XPath:
www.w3c.org/TR/xpath

SQL/XML-IMDB:
www.quilogic.cc/whitep.pdf