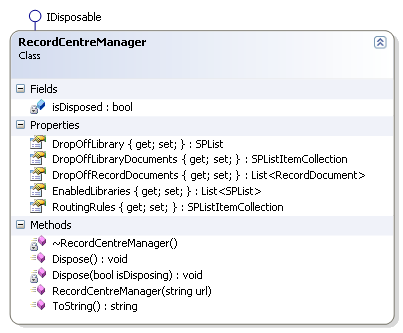
Record Centre Manager

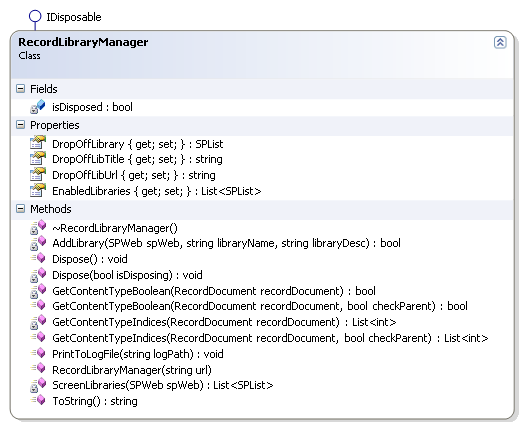


Initialized with the string url corresponding to the Record Centre’s site.

It uses a:

* Record Library Manager for the Enabled Libraries plus the Drop Off Library.
* Routing Rules Manager for the rules from the Drop Off Library.
* Record Document Manager for the Record Documents in the Drop Off Library.

# Record Library Manager



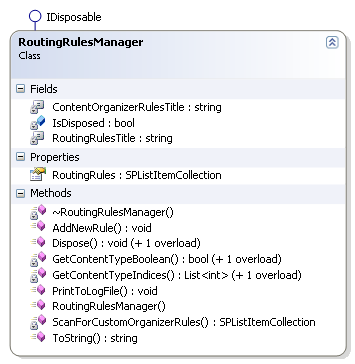
It is initialized with the string corresponding to the site.

It deduces the url for the Drop Off Library, its title, its list and the Enable Libraries.

It is capable of deducing if the Record Document’s content type (or parent’s) presence and also the libraries where they can fit.

It can create also the corresponding library for a missing content type.

# Routing Rules Manager



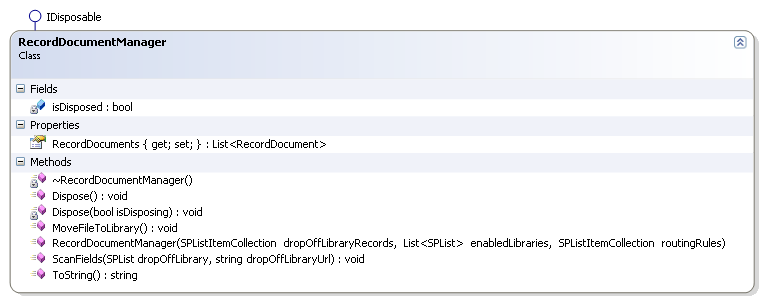
It is initialized with a string for the corresponding site.

It scans for the custom organizer rules.

It is capable of adding new rules.

It can also find if the content type (or parent’s) presence with the corresponding lists.

# Record Document Manager

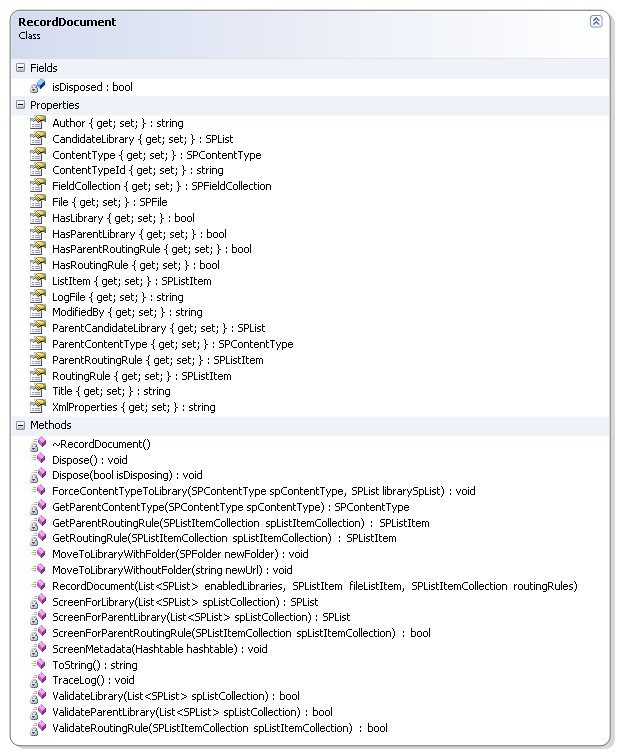


It manages Record Documents creating a list. It is created using the Drop Off Library Records (List Item Collection), a list of Enabled Libraries and Routing Rules (List Item Collection).

It can scan fields using a Drop Off Library list and a Drop Off Library Url. By doing so, it creates a Record Field Manager for each Record Document.

It move also files to its corresponding library based on each file’s content type (or parent’s).

## Record Document



It is initialized with a list of Enabled Libraries, a list item for the file and a list item collection for the routing rules.

It collects metadata from file such as title, Xml properties, and content type (or parent’s) and library related information.

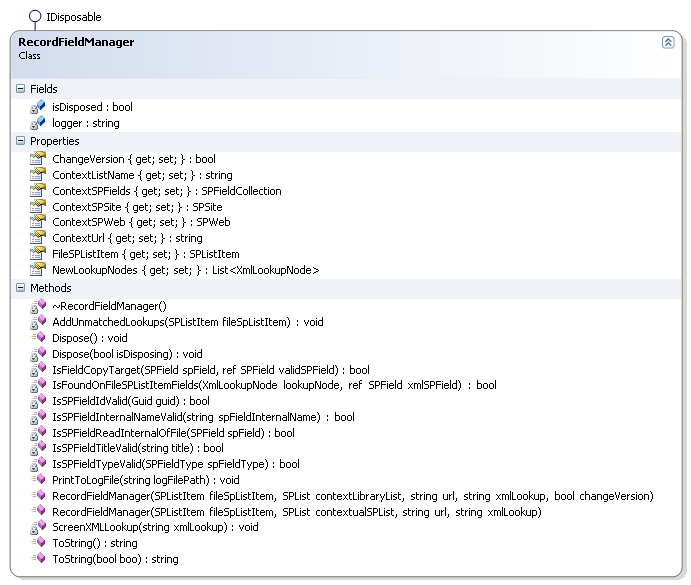
It can move the corresponding file based on the content type to the proper library.

It can force the moving of its corresponding file also.

It scans for the corresponding library (or parent’s).

It scans for the corresponding content type (or parent’s).

# Record Field Manager



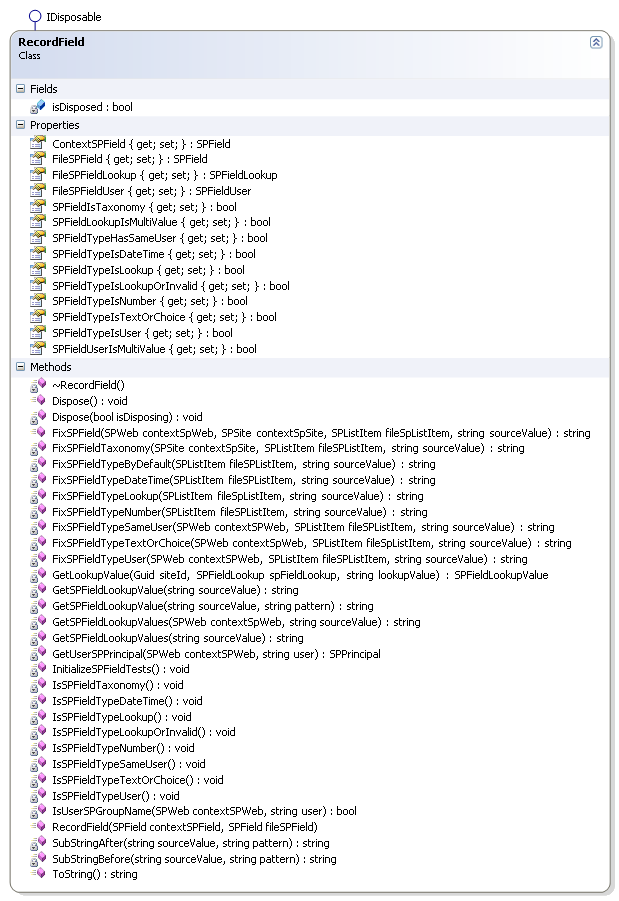
It manages information of the current context field information. Initialized with a list item for the file, a list for the current library, the current url, the file’ lookup information in Xml format.

It handles all the logic for analyzing the information of a field.

It screens also the lookup information store in Xml format:

* Reading done by a Xml Lookup Reader initiated with the Xml information
* Each property is stored in a Xml Lookup Node.
* Each field is updated using a Record Field.

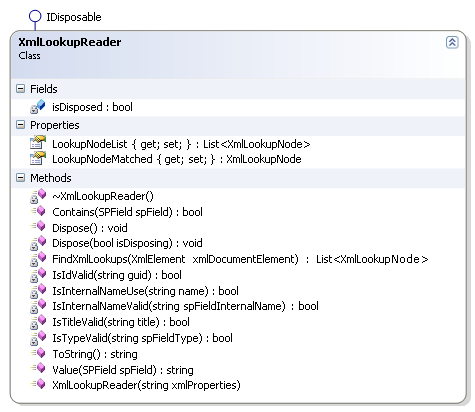
## Record Field



It contains all the business logic in order to update the field. It is initialized with the fields from the context and with the field from the file.

XML Lookup Information

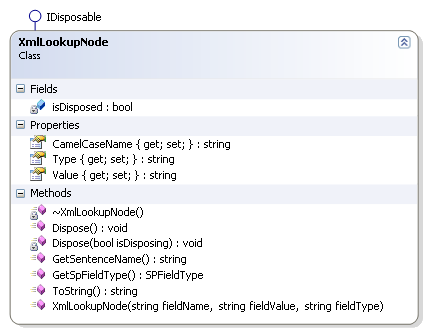
# XML Lookup Reader



It is initialized with an Xml format string containing all the lookup information by screening all the properties children and storing them in a list of Xml Lookup Nodes.

It has the business logic to skip a property without the proper information, adding to the list any property passing those tests.

## XML Lookup Node



It is initialized by a string field name stored in camel case, a string field value and a string field type.

It gives when needed a name transformed from camel case to sentence case, each word separated with a space; each word starts with capital case.

It also gives the corresponding field object screened from the structure fields corresponding to the string field type.

Functional Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| <http://dev2010/Scrum/Shared%20Documents> | Parent Content Type | Library | Library Content Type | Rule | Folder |
| MyContent | Document | MyContentLibrary | MyContent | MyContentRule | N |
| Xls | Document | XlsLibrary | Xls | XlsRule | Y |
| Csv | Xls |  |  |  | N |
| Doc | Document | DocLibrary | Doc | DocRule | N |
| Txt | Doc |  |  |  | N |
| Vsd | Document | VsdLibrary | Vsd | VsdRule | N |
| Pdf | Document | PdfLibrary | Pdf | PdfRule | N |
| Avi | Video | AviLibrary | Avi | AviRule | N |
| Jpg | Image | JpgLibrary | Jpg | JpgRule | N |
| Png | Image | PngLibrary | Png | PngRule | N |
| Mp3 | Audio | Mp3Library | Mp3 | Mp3Rule | N |

In this table is shown the set of tests done with certain content types invented to satisfy the extension of a file.

The Excel Library has a default Xls Folder/ and a rule for Excel files but not for a Comma Separated Value file. There is no library; therefore no rule for comma separated files. The comma separated parent content type is Excel; the solution matches to it by default and places the file under its folder by default.

The Word Library has no folder; it has a rule for Word files but not for a Text file. The text file parent content type is Word; the solution matches to it by default and places the file under the library.