

EiffelRSS

Martin Luder Michael Käser Thomas Weibel

31st January 2005

What is RSS?

- Acronym for Really Simple Syndication, Rich Site Summary or RDF Site Summary
- XML format for syndicating news, the content of news-like sites and pretty much anything that can be broken down into discrete items, e.g. the "recent changes" page of a Wiki, a changelog of SVN checkins, even the revision history of a book
- Once information about each item is in RSS format, an RSS-aware program can check the feed for changes and react to the changes in an appropriate way.
- There are 9 different and incompatible versions of RSS, EiffelRSS handles only RSS 2.0 at the moment, but can handle all of them through specialized reader and writer classes.

What is RSS?

- Acronym for Really Simple Syndication, Rich Site Summary or RDF Site Summary
- XML format for syndicating news, the content of news-like sites and pretty much anything that can be broken down into discrete items, e.g. the "recent changes" page of a Wiki, a changelog of SVN checkins, even the revision history of a book
- Once information about each item is in RSS format, an RSS-aware program can check the feed for changes and react to the changes in an appropriate way.
- There are 9 different and incompatible versions of RSS, EiffelRSS handles only RSS 2.0 at the moment, but can handle all of them through specialized reader and writer classes.

What is RSS?

- Acronym for Really Simple Syndication, Rich Site Summary or RDF Site Summary
- XML format for syndicating news, the content of news-like sites and pretty much anything that can be broken down into discrete items, e.g. the "recent changes" page of a Wiki, a changelog of SVN checkins, even the revision history of a book
- Once information about each item is in RSS format, an RSS-aware program can check the feed for changes and react to the changes in an appropriate way.
- There are 9 different and incompatible versions of RSS, EiffelRSS handles only RSS 2.0 at the moment, but can handle all of them through specialized reader and writer classes.

What is RSS?

- Acronym for Really Simple Syndication, Rich Site Summary or RDF Site Summary
- XML format for syndicating news, the content of news-like sites and pretty much anything that can be broken down into discrete items, e.g. the "recent changes" page of a Wiki, a changelog of SVN checkins, even the revision history of a book
- Once information about each item is in RSS format, an RSS-aware program can check the feed for changes and react to the changes in an appropriate way.
- There are 9 different and incompatible versions of RSS, EiffelRSS handles only RSS 2.0 at the moment, but can handle all of them through specialized reader and writer classes.

What is EiffelRSS?

- EiffelRSS is an Eiffel library to read and write RSS. The goal is to provide the Eiffel development community with an easy to use and well structured API for RSS.
- The distribution also contains a RSS newsfeed reader written with EiffelVision and EiffelRSS.

What is EiffelRSS?

- EiffelRSS is an Eiffel library to read and write RSS. The goal is to provide the Eiffel development community with an easy to use and well structured API for RSS.
- The distribution also contains a RSS newsfeed reader written with EiffelVision and EiffelRSS.

EiffelRSS Library: Helper Clusters

- ADT features classes to implement sortable structures.
- FETCH can fetch data from a source address to a local STRING using HTTP, FTP and file.
- LOGFILE represents a file which can be used for logging messages during the program execution.
- PROPERTIES represents a persistent set of properties.

EiffelRSS Library: Helper Clusters

- ADT features classes to implement sortable structures.
- **FETCH** can fetch data from a source address to a local **STRING** using HTTP, FTP and file.
- LOGFILE represents a file which can be used for logging messages during the program execution.
- PROPERTIES represents a persistent set of properties.

EiffelRSS Library: Helper Clusters

- ADT features classes to implement sortable structures.
- FETCH can fetch data from a source address to a local STRING using HTTP, FTP and file.
- LOGFILE represents a file which can be used for logging messages during the program execution.
- PROPERTIES represents a persistent set of properties.

EiffelRSS Library: Helper Clusters

- ADT features classes to implement sortable structures.
- FETCH can fetch data from a source address to a local STRING using HTTP, FTP and file.
- LOGFILE represents a file which can be used for logging messages during the program execution.
- PROPERTIES represents a persistent set of properties.

EiffelRSS Library: SYNDICATION

- **SYNDICATION** is the main cluster of EiffelRSS with a feed object model, and classes to load and write feeds. It has three subclusters:
 - **INTERFACE** contains all the classes a developer needs to use the library.
 - **FEED** is the central datastructure of EiffelRSS. It defines an abstract syndication feed.
 - **FORMATS** defines the different syndication formats. It is easy extensible with other formats.

EiffelRSS Library: SYNDICATION

- SYNDICATION is the main cluster of EiffelRSS with a feed object model, and classes to load and write feeds. It has three subclusters:
 - **INTERFACE** contains all the classes a developer needs to use the library.
 - FEED is the central datastructure of EiffelRSS. It defines an abstract syndication feed.
 - FORMATS defines the different syndication formats. It is easy extensible with other formats.

EiffelRSS Library: SYNDICATION

- SYNDICATION is the main cluster of EiffelRSS with a feed object model, and classes to load and write feeds. It has three subclusters:
 - INTERFACE contains all the classes a developer needs to use the library.
 - FEED is the central datastructure of EiffelRSS. It defines an abstract syndication feed.
 - FORMATS defines the different syndication formats. It is easy extensible with other formats.

EiffelRSS Library: SYNDICATION

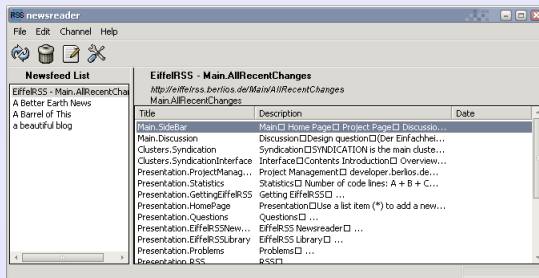
- SYNDICATION is the main cluster of EiffelRSS with a feed object model, and classes to load and write feeds. It has three subclusters:
 - INTERFACE contains all the classes a developer needs to use the library.
 - FEED is the central datastructure of EiffelRSS. It defines an abstract syndication feed.
 - FORMATS defines the different syndication formats. It is easy extensible with other formats.

[EIFFEL RSS SYNDICATION](#)
[EIFFEL RSS SYNDICATION JAR](#)



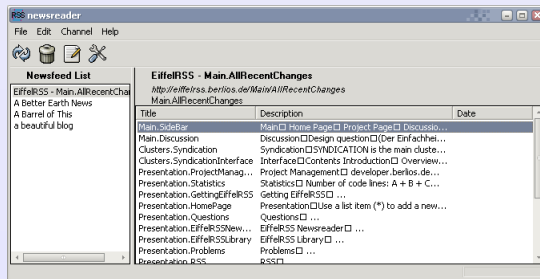
EiffelRSS newsreader

- Newsreader is a simple RSS-feed reader which shows the possibilities of the EiffelRSS library.
- You can add custom feeds and open news in your Internet browser.
- Features a graphical and command line user interface.



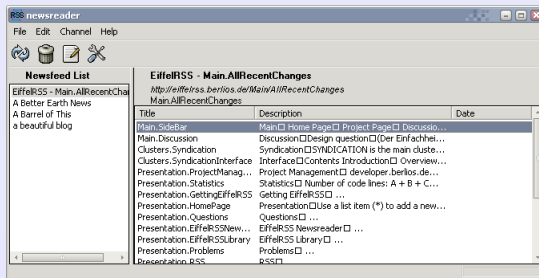
EiffelRSS newsreader

- Newsreader is a simple RSS-feed reader which shows the possibilities of the EiffelRSS library.
- You can add custom feeds and open news in your Internet browser.
- Features a graphical and command line user interface.



EiffelRSS newsreader

- Newsreader is a simple RSS-feed reader which shows the possibilities of the EiffelRSS library.
- You can add custom feeds and open news in your Internet browser.
- Features a graphical and command line user interface.



Getting EiffelRSS

- <http://eiffelrss.berlios.de>
- Subversion: `svn checkout svn://svn.berlios.de/eiffelrss`

Getting EiffelRSS

- `http://eiffelrss.berlios.de`
- Subversion: `svn checkout svn://svn.berlios.de/eiffelrss`

Project Management

- developer.berlios.de
- Subversion
- PmWiki
- Gobo Eiffel test (getest)

Project Management

- [developer.berlios.de](#)
- [Subversion](#)
- [PmWiki](#)
- [Gobo Eiffel test \(getest\)](#)

Project Management

- [developer.berlios.de](#)
- [Subversion](#)
- [PmWiki](#)
- [Gobo Eiffel test \(getest\)](#)

Project Management

- developer.berlios.de
- Subversion
- PmWiki
- Gobo Eiffel test (getest)

Problems

- EiffelNet has only support for HTTP 1.0 and is quite unstable
- The XML parser of gobo has problems with entites and special characters
- No sortable data structures
- No real stream concept, e.g. input and output are no files

Problems

- EiffelNet has only support for HTTP 1.0 and is quite unstable
- The XML parser of gobo has problems with entites and special characters
- No sortable data structures
- No real stream concept, e.g. input and output are no files

Problems

- EiffelNet has only support for HTTP 1.0 and is quite unstable
- The XML parser of gobo has problems with entites and special characters
- No sortable data structures
- No real stream concept, e.g. input and output are no files

Problems

- EiffelNet has only support for HTTP 1.0 and is quite unstable
- The XML parser of gobo has problems with entites and special characters
- No sortable data structures
- No real stream concept, e.g. input and output are no files

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 -
 -
 -

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 -
 -
 -

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z

Statistics

- Number of Eiffel code lines: $A + B + C$
 - Library: A
 - Newsreader: B
 - Examples and testing framework: C
- Number of Eiffel classes: $X + Y + Z$
 - Library: X
 - Newsreader: Y
 - Examples and testing framework: Z