

# API Documentation

API Documentation

November 12, 2012

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package changemonitor</b>	<b>2</b>
1.1 Modules . . . . .	2
1.2 Variables . . . . .	2
1.3 Class MonitoredResource . . . . .	2
1.3.1 Methods . . . . .	4
1.3.2 Properties . . . . .	5
1.4 Class Monitor . . . . .	6
1.4.1 Methods . . . . .	7
1.4.2 Properties . . . . .	9
<b>2 Module changemonitor._http</b>	<b>10</b>
2.1 Variables . . . . .	10
2.2 Class HTTPDateTime . . . . .	10
2.2.1 Methods . . . . .	10
2.2.2 Properties . . . . .	12
<b>3 Module changemonitor.diff</b>	<b>14</b>
3.1 Variables . . . . .	14
3.2 Class DocumentDiff . . . . .	14
3.2.1 Methods . . . . .	14
3.2.2 Properties . . . . .	14
3.3 Class PlainTextDiff . . . . .	15
3.3.1 Methods . . . . .	15
3.3.2 Properties . . . . .	15
3.4 Class BinaryDiff . . . . .	16
3.4.1 Methods . . . . .	16
3.4.2 Properties . . . . .	16
3.5 Class HtmlDiff . . . . .	16
3.5.1 Methods . . . . .	16
3.5.2 Properties . . . . .	17
<b>4 Module changemonitor.errors</b>	<b>18</b>
4.1 Variables . . . . .	18
4.2 Class ChangeMonitorError . . . . .	18
4.2.1 Methods . . . . .	18

4.2.2	Properties . . . . .	18
4.3	Class DocumentTooLarge . . . . .	19
4.3.1	Methods . . . . .	19
4.3.2	Properties . . . . .	19
4.4	Class DocumentNotAvailable . . . . .	20
4.4.1	Methods . . . . .	20
4.4.2	Properties . . . . .	20
4.5	Class DocumentHistoryNotAvaliable . . . . .	21
4.5.1	Methods . . . . .	21
4.5.2	Properties . . . . .	21
4.6	Class NotSupportedYet . . . . .	22
4.6.1	Methods . . . . .	22
4.6.2	Properties . . . . .	22
4.7	Class UidError . . . . .	23
4.7.1	Methods . . . . .	23
4.7.2	Properties . . . . .	23
<b>5</b>	<b>Module changemonitor.model</b>	<b>24</b>
5.1	Variables . . . . .	24
5.2	Class BaseMongoModel . . . . .	24
5.2.1	Methods . . . . .	24
5.2.2	Properties . . . . .	24
5.3	Class Storage . . . . .	25
5.3.1	Methods . . . . .	26
5.3.2	Properties . . . . .	26
5.4	Class File . . . . .	27
5.4.1	Methods . . . . .	28
5.4.2	Properties . . . . .	29
5.5	Class Diffable . . . . .	29
5.5.1	Methods . . . . .	30
5.5.2	Properties . . . . .	30
5.6	Class Content . . . . .	30
5.6.1	Methods . . . . .	31
5.6.2	Properties . . . . .	32
5.7	Class HttpHeadersMeta . . . . .	32
5.7.1	Methods . . . . .	32
5.7.2	Properties . . . . .	34
<b>6</b>	<b>Module changemonitor.resolver</b>	<b>35</b>
6.1	Variables . . . . .	35
6.2	Class Resolver . . . . .	35
6.2.1	Methods . . . . .	35
6.2.2	Properties . . . . .	35
6.3	Class Rule . . . . .	36
6.3.1	Methods . . . . .	36
6.3.2	Properties . . . . .	36

# 1 Package changemonitor

Changemonitor – monitoring changes on web

TODO: docstring. 1) koncepcie, casti (checking, availability, versioning, differ) 2) user-view vs global-view 3) usage

The very basic and most probable usage:

```
>>> from rrslib.web.changemonitor import Monitor
>>> monitor = Monitor(user_id="rrs_university")
>>> resource = monitor.get("http://www.google.com")
>>> # if the page changed
>>> if resource.check():
>>>     print res.get_diff(start='last', end='now')
```

Date: \$21.6.2012 16:08:11\$

Author: Stanislav Heller

## 1.1 Modules

- **\_http**: Maly privatni modul zajistujici HTTP pozadavky pro changemonitor.  
(Section 2, p. 10)
- **diff** (Section 3, p. 14)
- **errors**: Exceptions raised by rrslib.web.changemonitor package  
(Section 4, p. 18)
- **model**: Model for changemonitor – interface for accessing the data  
(Section 5, p. 24)
- **resolver** (Section 6, p. 35)

## 1.2 Variables

Name	Description
HTTPDateTime	

## 1.3 Class MonitoredResource

object —  
    **changemonitor.MonitoredResource**

Monitored resource (URL). The resource is generally any document in any format, but most often it will be HTML code.

This class wraps the URL content and metadata.

The contents can be manipulated within the time so it can provide information about how the content changed in different versions of the document.

Warning: Application developers should generally not need to instantiate this class directly. The only correct way how to get this object is through `Monitor.get()` method.

Design pattern: Active Record

Example of checking new version of document:

```
>>> from rrslib.web.changemonitor import Monitor
>>> monitor = Monitor(user_id="myuid")
>>> monitor
Monitor(conn=Connection('localhost', 27017), dbname='webarchive', uid='myuid')
>>> resource = monitor.get("http://www.myusefulpage.com/index.html")
>>> resource
<MonitoredResource(url='http://www.myusefulpage.com/index.html', uid='myuid') at 0xb7398accL>
>>> resource.check()
True
>>> # the resource has changed
```

Checking availability of the document on the URL

```
>>> from rrslib.web.changemonitor import HTTPDateTime
>>> resource = monitor.get("http://www.nonexistentpage.com")
>>> resource.available()
False
>>> resource = monitor.get("http://www.myusefulpage.com/index.html")
>>> resource.available(HTTPDateTime(2012, 6, 30, 15, 34))
True
```

Example of getting last available version of the document on the URL

```
>>> resource = monitor.get("http://www.myusefulpage.com/index.html")
>>> content = resource.get_last_version()
>>> print content.data
<html><head>
...
>>> resource = monitor.get("http://www.crazynonexistentpage.com")
>>> content = resource.get_last_version()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
DocumentNotAvailable: The content of this URL is not available and it
is not in the storage.
```

Example of getting version of the document in exact time

```
>>> resource = monitor.get("http://www.myusefulpage.com/index.html")
>>> content = resource.get_version(HTTPDateTime(2012, 6, 30, 15, 34))
```

Getting the last time when the document was checked:

```
>>> resource = monitor.get("http://www.crazynotexistentpage.com")
>>> resource.last_checked()
HTTPDateTime(Thu, 01 Jan 1970 00:00:00 GMT)
```

### 1.3.1 Methods

**\_\_init\_\_**(*self*, *url*, *uid*, *storage*)

x.**\_\_init\_\_**(...) initializes x; see help(type(x)) for signature

**Parameters**

**url:** monitored URL  
*(type=basestring (str or unicode))*

**uid:** user identifier. This ID has to be unique for each one, who is using changemonitor.  
*(type=str)*

**storage:** storage of monitored-resource data  
*(type=model.Storage)*

Overrides: object.**\_\_init\_\_**

**check**(*self*)

Check the resource URL and load the most recent version into database.

TODO: consider using @lazy decorator. Most of use cases use this method so we have to insure that it will be called only once.

**Return Value**

True if the document has changed since last check.

**get\_last\_version**(*self*)

Get last available content of the document. If the document is available at this time, returns most recent version which is on the web server.

**Return Value**

Last available content of this resource.

*(type=Content)*

**get\_version**(*self*, *time\_or\_version*)

Get content of this document in specified time or version. If the document was not available in given time, returns last available content. If there is no available content until given time, raises exception.

**Parameters**

**time\_or\_version:** Time or version of the content we want to retrieve. Version numbering is a convenience atop the GridFS API provided by MongoDB. version “-1” will be the most recently uploaded matching file, “-2” the second most recently uploaded, etc. Version “0” will be the first version uploaded, “1” the second version, etc. So if three versions have been uploaded, then version “0” is the same as version “-3”, version “1” is the same as version “-2”, and version “2” is the same as version “-1”.

*(type=HTTPDateTime or int)*

**get\_diff**(*self*, *start*, *end*)

**Parameters**

**start:** start time or version to be diffed

(*type=HTTPDateTime or int*)

**end:** end time or version to be diffed

(*type=HTTPDateTime or int*)

**Return Value**

either textual or binary diff of the file (if available). If contents are equal (document did not change within this time range) returns None.

(*type=unicode*)

**available**(*self*, *httptime=None*)

**last\_checked**(*self*)

Get information about the time of last check of this resource.

**Return Value**

time of last check or None if the resource was never checked (or the HTTP requests timed out)

(*type=HTTPDateTime or None*)

**\_\_repr\_\_**(*self*)

str(x)

Overrides: object.\_\_repr\_\_ extit(inherited documentation)

**\_\_str\_\_**(*self*)

str(x)

Overrides: object.\_\_str\_\_ extit(inherited documentation)

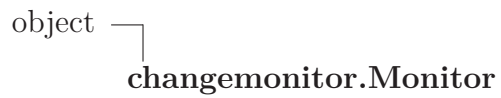
**Inherited from object**

**\_\_delattr\_\_**(*self*, *name*), **\_\_format\_\_**(*self*, *format\_spec*), **\_\_getattr\_\_**(*self*, *name*), **\_\_hash\_\_**(*self*), **\_\_new\_\_**(*cls*, *args*, *kwargs*), **\_\_reduce\_\_**(*self*), **\_\_reduce\_ex\_\_**(*self*, *proto*), **\_\_setattr\_\_**(*self*, *name*, *value*), **\_\_sizeof\_\_**(*self*), **\_\_subclasshook\_\_**(*self*, *subclass*)

**1.3.2 Properties**

Name	Description
<i>Inherited from object</i>	
<b>__class__</b>	

## 1.4 Class Monitor



Monitor is main class representing web change monitor. It serves as factory for creating MonitoredResource objects.

Usage:

```
>>> from rrslib.web.changemonitor import Monitor
>>> monitor = Monitor(user_id="rrs_university")
>>> resource = monitor.get("http://www.google.com")
>>> # if the page changed
>>> if resource.check():
>>>     print res.get_diff(start='last', end='now')
```

## 1.4.1 Methods

---

```
__init__(self, user_id, db_host="localhost", db_port=27017,
db_name="webarchive", http_proxy=None)
```

---

Create a new monitor connected to MongoDB at *\*db\_host:db\_port\** using database *db\_name*.

**Parameters**

- user\_id:** identification string of user/module who uses monitor. If *user\_id* is given *None*, the monitor switches to ‘global-view’ mode and all requests to storage don’t care about >>who checked this resource<<. On the other hand, if *user\_id* is given a string, the monitor switches to ‘user-view’ mode and all operations are oriented to the user. Most of the reasonable use cases are using *user\_id*, because a user/module almost everytime ask about >>what changed since I have been here for the last time<<, not >>what changed since somebody has been here for the last time<<...  
(*type=*str or None)
- db\_host:** (optional) hostname or IP address of the instance to connect to, or a mongodb URI, or a list of hostnames / mongodb URIs. If *db\_host* is an IPv6 literal it must be enclosed in ‘[’ and ‘]’ characters following the RFC2732 URL syntax (e.g. ‘[::1]’ for localhost)
- db\_port:** (optional) port number on which to connect  
(*type=*int)
- db\_name:** name of database which is used to store information about monitored documents and their versions.  
(*type=*str)
- http\_proxy:** (FUTURE USE) proxy server where to send requests  
(*type=*unknown)

Overrides: object.\_\_init\_\_

---



---

**get**(*self*, *url*)

---

Creates new MonitoredResource instance which represents document on \*url\*.

**Parameters**

**url**: URL of monitored resource  
*(type=str)*

**Return Value**

monitored resource object bound to URL \*url\*.  
*(type=MonitoredResource)*  
*Design pattern: factory method.)*

---

**allow\_large\_documents**(*self*)

---

Allow large objects to be stored in the storage. Large document is defined as file larger than 4096KB. This constant is defined in this module named as LARGE\_DOCUMENT\_SIZE representing size of the file in kilobytes.

---

**check\_uid**(*self*)

---

Check if user id given in constructor is a valid user id within the Monitor storage system. If the UID is occupied, returns False, True otherwise.

If user\_id is None, an exception UidError is raised.

**Return Value**

True if the UID is free  
*(type=bool)*

---

**check\_multi**(*self*, *urls*=[])

---

Check list of urls, start new thread for each one.

**Parameters**

**urls**: *(type=list)*

**Return Value**

list of MonitoredResource objects, each with actual data  
*(type=list<MonitoredResource>)*

---

**\_\_repr\_\_**(*self*)

---

str(x)

Overrides: object.\_\_repr\_\_ extit(inherited documentation)

```

__str__(self)

str(x)

Overrides: object.__str__ extit(inherited documentation)

```

### *Inherited from object*

```

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

```

### 1.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 2 Module changemonitor.\_\_http

Maly privatni modul zajistujici HTTP pozadavky pro changemonitor. Obsahuje maly middleware, který slouží k odstínění changemonitoru od nižší vrstvy HTTP a který poskytné možnost pohodlnějšího testování nebo cachování.

**Date:** \$22.6.2012 13:01:57\$

**Author:** Stanislav Heller

### 2.1 Variables

Name	Description
<code>__modulename__</code>	<b>Value:</b> <code>"_http"</code>
<code>__email__</code>	<b>Value:</b> <code>"xhelle03@stud.fit.vutbr.cz"</code>

### 2.2 Class HTTPDateTime

object └─  
           **changemonitor.\_\_http.HTTPDateTime**

Datetime class for manipulating time within HTTP environment.

Usage: `>>> h = HTTPDateTime() >>> h HTTPDateTime(Thu, 01 Jan 1970 00:00:00 GMT) >>> h.now() >>> h HTTPDateTime(Sat, 30 Jun 2012 16:09:43 GMT) >>> h.to_httpheader_format() 'Sat, 30 Jun 2012 16:09:43 GMT'`

FIXME: solve the problem with GMT: `>>> h = HTTPDateTime() >>> h.to_timestamp() -7200.0 # WTF?`

#### 2.2.1 Methods

`__init__(self, year=1970, month=1, day=1, hour=0, minute=0, second=0, microsecond=0)`

`x.__init__(...)` initializes x; see `help(type(x))` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

---

**to\_httpheader\_format**(*self*)

Converts this object into date and time in HTTP-header format.

**Return Value**

date and time in HTTP format, i.e. 'Wed, 31 Aug 2011 16:45:03 GMT'.

(*type=*str)

---

**from\_httpheader\_format**(*self*, *timestr*)

Parse http header datetime format and save into this object.

**Parameters**

**timestr:** date and time in format which is used by HTTP protocol  
(*type=*str)

**Return Value**

HTTPDateTime object equivalent to date and time of the timestr  
(*type=HTTPDateTime*)

---

**to\_timestamp**(*self*)

Convert into UNIX timestamp. (seconds since start of the UNIX epoch).

**Return Value**

unix timestamp  
(*type=*float)

---

**from\_timestamp**(*self*, *timestamp*)

Set date and time from timestamp.

**Parameters**

**timestamp:** time since start of the unix epoch  
(*type=*float)

**Return Value**

HTTPDateTime object representing date and time of the timestamp  
(*type=HTTPDateTime*)

---

**to\_datetime**(*self*)

---

 Convert the date and time from this object into python's datetime.datetime.
**Return Value**

datetime object equivalent to date and time of this object

*(type=datetime.datetime)*


---

**from\_datetime**(*self*, *datetimeobj*)

---

**now**(*self*)

---

 Set the time of this object as current time (time.time())
**Return Value**

HTTPDateTime object representing current date and time.

*(type=HTTPDateTime)*


---

**\_\_repr\_\_**(*self*)

repr(x)

Overrides: object.\_\_repr\_\_ exitit(inherited documentation)

---

**\_\_lt\_\_**(*self*, *other*)

---

**\_\_le\_\_**(*self*, *other*)

---

**\_\_eq\_\_**(*self*, *other*)

---

**\_\_ne\_\_**(*self*, *other*)

---

**\_\_gt\_\_**(*self*, *other*)

---

**\_\_ge\_\_**(*self*, *other*)
**Inherited from object**
 \_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(),  
 \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(),  
 \_\_subclasshook\_\_()
**2.2.2 Properties**

---

Name	Description
<i>Inherited from object</i> __class__	

### 3 Module changemonitor.diff

**Date:** \$25.6.2012 12:12:44\$

**Author:** Stanislav Heller

#### 3.1 Variables

Name	Description
<code>__modulename__</code>	<b>Value:</b> "diff"
<code>__email__</code>	<b>Value:</b> "xhelle03@stud.fit.vutbr.cz"

#### 3.2 Class DocumentDiff

object —  
**changemonitor.diff.DocumentDiff**

Zakladni interface sjednocujici pristup k diffovani dokumentu.

Zdedena trida musi implementovat tridu diff, ktera se stara o diffnuti dvou dokumentu stejnych typu.

##### 3.2.1 Methods

<b>diff</b> (cls, obj1, obj2)
-------------------------------

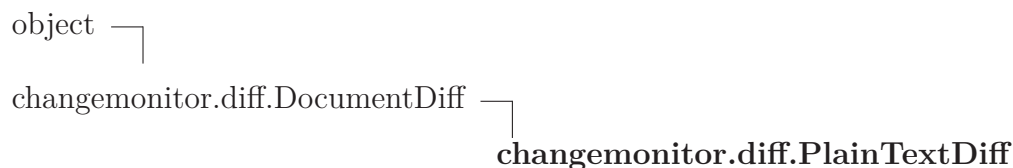
*Inherited from object*

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`,  
`__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`,  
`__sizeof__()`, `__str__()`, `__subclasshook__()`

##### 3.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

### 3.3 Class PlainTextDiff



This class is a diff wrapper around classical gnu diff. Using this differ we can process every text documents (plaintext, html, css etc.)

#### 3.3.1 Methods

**diff**(cls, obj1, obj2)

**Parameters**

obj1: first text to be diffed  
(type=string or unicode)

obj2: first text to be diffed  
(type=string or unicode)

**Return Value**

unicode diff  
(type=unicode)

Overrides: changemonitor.diff.DocumentDiff.diff

#### *Inherited from object*

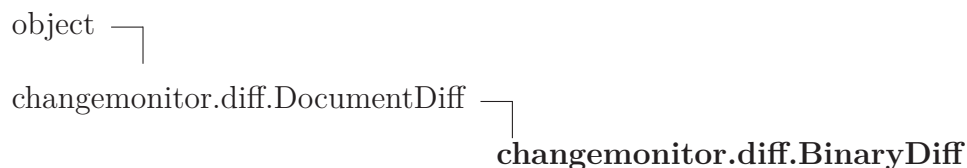
\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_init\_\_(),  
\_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(),  
\_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 3.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	



### 3.4 Class BinaryDiff



Tato trida bude diffovat binarni dokumenty - prevazne pdf, odt, doc atp.

#### 3.4.1 Methods

**diff**(cls, obj1, obj2)

Overrides: changemonitor.diff.DocumentDiff.diff

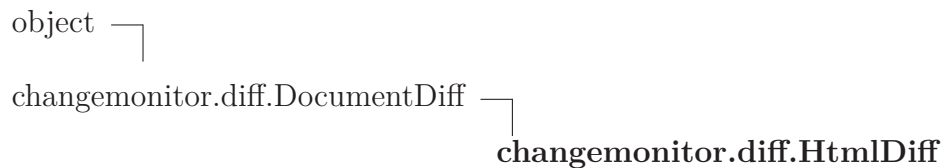
#### *Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_init\_\_(),  
 \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(),  
 \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 3.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

### 3.5 Class HtmlDiff



Html diff, which shows pieces of code, which was added to the page. Uses output of GNU diff and its interface PlainTextDiff.

#### 3.5.1 Methods

**htmldiff**(cls, raw\_diff)

**diff**(cls, obj1, obj2)

Overrides: changemonitor.diff.DocumentDiff.diff

***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`,  
`__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`,  
`__sizeof__()`, `__str__()`, `__subclasshook__()`

**3.5.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 4 Module *changemonitor.errors*

Exceptions raised by `rrslib.web.changemonitor` package

**Date:** \$22.6.2012 13:01:57\$

**Author:** Stanislav Heller

### 4.1 Variables

Name	Description
<code>__modulename__</code>	<b>Value:</b> "errors"
<code>__email__</code>	<b>Value:</b> "xhelle03@stud.fit.vutbr.cz"

### 4.2 Class *ChangeMonitorError*



Base error class for all exceptions within this package.

#### 4.2.1 Methods

*Inherited from `exceptions.Exception`*

`__init__()`, `__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

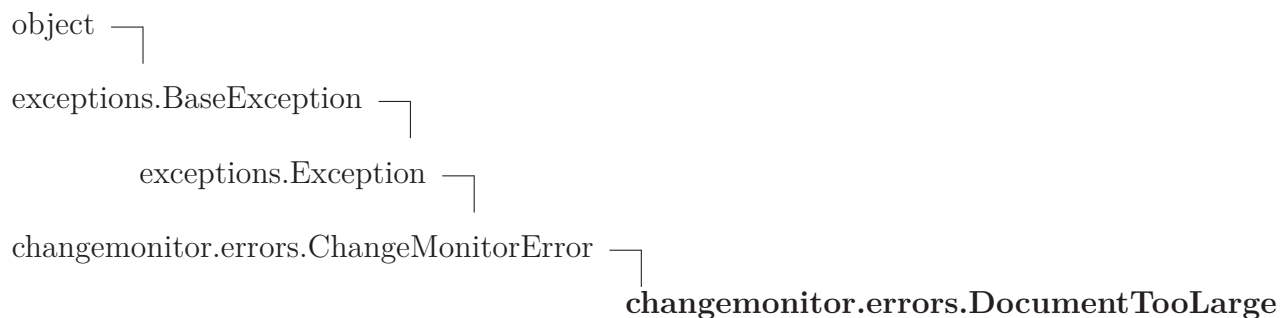
*Inherited from `object`*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

#### 4.2.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i> args, message	
<i>Inherited from object</i> __class__	

### 4.3 Class *DocumentTooLarge*



Raised when monitored document's size exceeds the `LARGE_DOCUMENT_SIZE` constant.

#### 4.3.1 Methods

*Inherited from exceptions.Exception*

\_\_init\_\_(), \_\_new\_\_()

*Inherited from exceptions.BaseException*

\_\_delattr\_\_(), \_\_getattr\_\_(), \_\_getitem\_\_(), \_\_getslice\_\_(), \_\_reduce\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(), \_\_setstate\_\_(), \_\_str\_\_(), \_\_unicode\_\_()

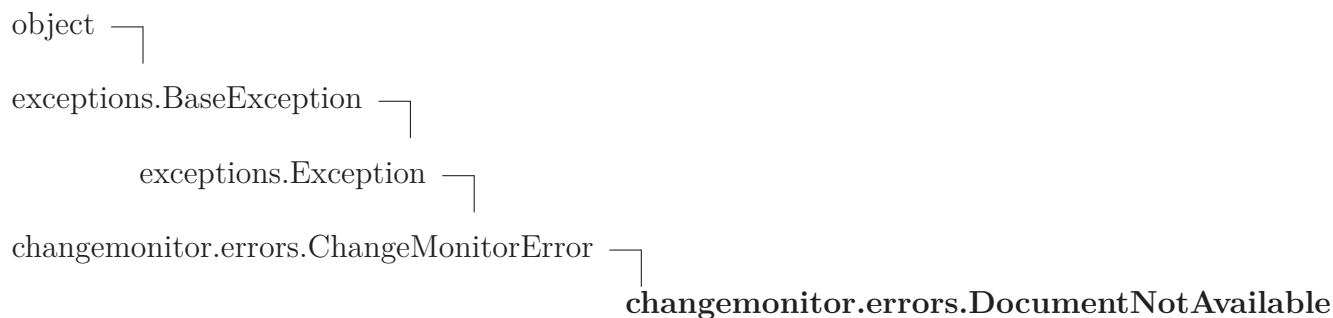
*Inherited from object*

\_\_format\_\_(), \_\_hash\_\_(), \_\_reduce\_ex\_\_(), \_\_sizeof\_\_(), \_\_subclasshook\_\_()

#### 4.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i> args, message	
<i>Inherited from object</i> __class__	

## 4.4 Class `DocumentNotAvailable`



Raised when document is not available on the URL or on the storage.

### 4.4.1 Methods

*Inherited from `exceptions.Exception`*

`__init__()`, `__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattribute__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

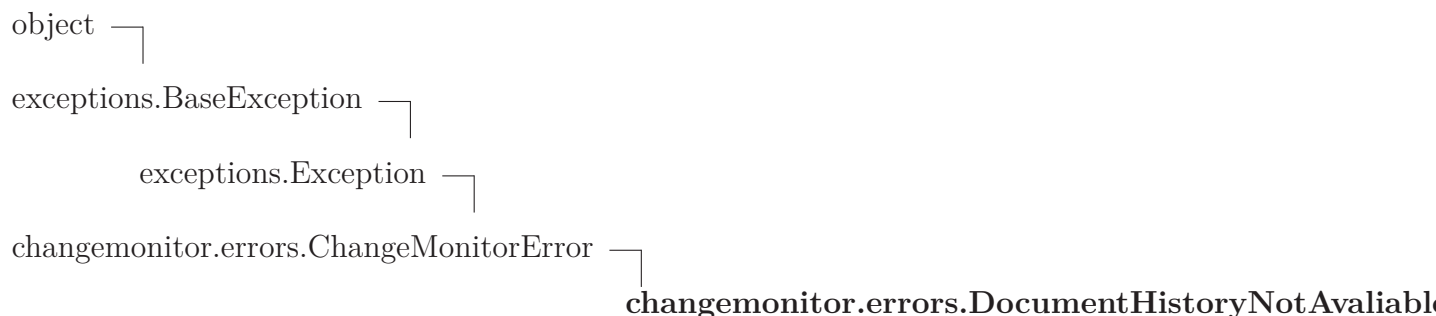
*Inherited from `object`*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

### 4.4.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

## 4.5 Class `DocumentHistoryNotAvaliable`



Raised when trying to get version or diff of document, which version history is not stored in the storage.

### 4.5.1 Methods

*Inherited from `exceptions.Exception`*

`__init__()`, `__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

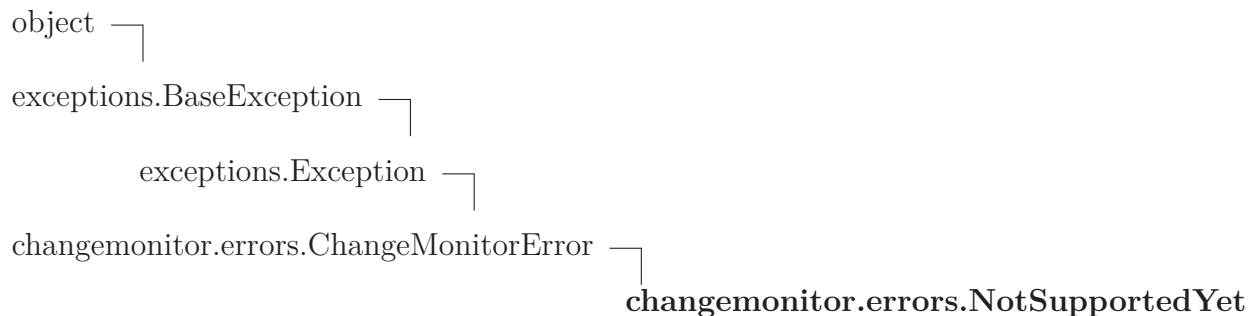
*Inherited from `object`*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

### 4.5.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

## 4.6 Class `NotSupportedYet`



Raised when the method/class/function is not supported in this implementation.

### 4.6.1 Methods

*Inherited from `exceptions.Exception`*

`__init__()`, `__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattribute__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

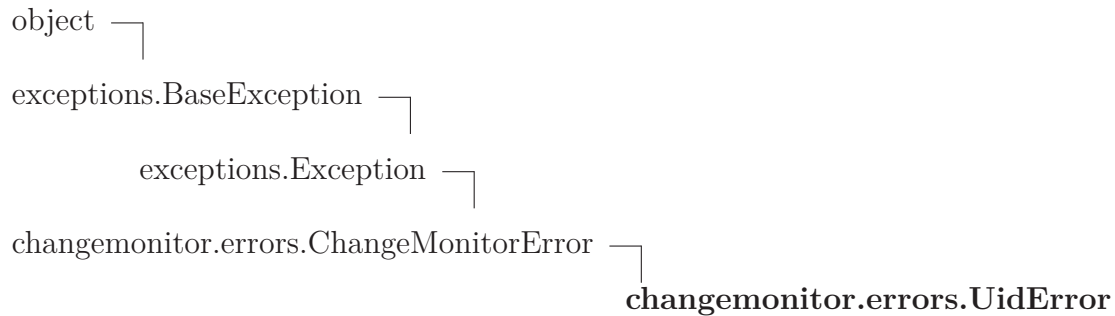
*Inherited from `object`*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

### 4.6.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

## 4.7 Class `UidError`



Raised when some error connected with user id occurred.

### 4.7.1 Methods

*Inherited from `exceptions.Exception`*

`__init__()`, `__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattribute__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

*Inherited from `object`*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

### 4.7.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
args, message	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	



## 5 Module changemonitor.model

Model for changemonitor – interface for accessing the data

This module creates abstraction layer between data storage implementation (SQL/NoSQL database/filesystem).

**Date:** \$23.6.2012 16:33:31\$

**Author:** Stanislav Heller

### 5.1 Variables

Name	Description
__modulename__	<b>Value:</b> "model"
__email__	<b>Value:</b> "xhelle03@stud.fit.vutbr.cz"

### 5.2 Class BaseMongoModel

object └─  
          **changemonitor.model.BaseMongoModel**

Serves as base class, which is inherited by every model class.

#### 5.2.1 Methods

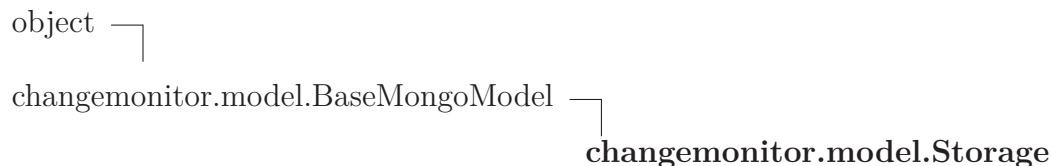
*Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_init\_\_(),  
\_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(),  
\_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 5.2.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

### 5.3 Class Storage



Abstraction of the storage. The purpose of this class is to create abstraction layer, which provides database-independent API for manipulation in the filesystem. The only requirement on the filesystem is that it has to support file versioning (or some workaround which implements versioning within the fs which does not support versioning natively).

The implementation is nowadays built on MongoDB.

Usage:

```

>>> from pymongo import Connection
>>> from model import Storage
>>> store = Storage(Connection(), "myuid", "webarchive")
>>> file = store.get("http://www.myfancypage.com/index.html")
>>> # get last version of the file, which is available in the storage
>>> c = file.get_last_content()
>>> # get the raw data
>>> c.data
"<html>
...
>>> # content type and content length
>>> print c.content_type, c.length
'text/html' 29481
  
```

Design pattern: Factory

### 5.3.1 Methods

<b>__init__</b> ( <i>self</i> , <i>connection</i> , <i>uid</i> , <i>database</i> ="webarchive")
Initializes storage.
<b>Parameters</b>
<i>connection</i> : database connection ( <i>type</i> = <i>pymongo.Connection</i> )
<i>uid</i> : user id (see Monitor.__doc__ for more info) ( <i>type</i> = <i>str</i> )
<i>database</i> : if the storage is based on database, this param represents the name of database to be used within this instance. ( <i>type</i> = <i>str</i> )
Overrides: object.__init__

<b>allow_large_documents</b> ( <i>self</i> )
Allow large objects to be stored in the storage.

<b>get</b> ( <i>self</i> , <i>filename</i> )
Get file object by filename.
<b>Parameters</b>
<i>filename</i> : name of the file. In this case, it will be URL. <i>filename</i> : ( <i>type</i> = <i>str</i> )
<b>Return Value</b>
File object representing file in many versions ( <i>type</i> = <i>File</i> )

<b>check_uid</b> ( <i>self</i> )
----------------------------------

#### *Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(),  
\_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(),  
\_\_str\_\_(), \_\_subclasshook\_\_()

### 5.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

## 5.4 Class File



One file in filesystem. A file can contain more contents in various versions. Main purpose of this class is to get rid of GridFS and GridOut instances and replace it with file-like wrapper.

MongoDB record:

```
content = {
    filename: URL
    md5: str
    sha1: str
    content_type: str
    length: int
    urls = []
}
```

Design pattern: Active Record

### 5.4.1 Methods

<b>__init__</b> ( <i>self, filename, fs, headermeta</i> )
Create new file instance.
<b>Parameters</b>
<b>filename:</b> name of the file ( <i>type=basestring (str or unicode)</i> )
<b>fs:</b> filesystem object ( <i>type=GridFS</i> )
<b>headermeta:</b> http header metadata ( <i>type=HTTPHeaderMeta</i> <i>WARNING: Application developers should generally not need to instantiate this class directly. The only correct way how to get this object is using Storage.get() method.</i> )
Overrides: object.__init__

<b>purge_cache</b> ( <i>self</i> )
Cleans the whole content cache.

<b>refresh_cache</b> ( <i>self</i> )
Refreshes part of cache, which can potentially change (version pointers). This is very useful if we expect the File object to live during more than one check() call. If so, the information about version has to be updated in the cache (version -1 becomes -2 etc.).
This method should be called after every check() call!

**get\_version**(*self*, *timestamp\_or\_version*)

Get content of the file in specific version. Version can be specified by version number (convenience atop the GridFS API by MongoDB) or unix timestamp.

**Parameters**

**timestamp\_or\_version:** version or timestamp of version which we want to retrieve.

(*type=int*)

**Return Value**

content of the file in specified time/version

(*type=Content*)

**get\_last\_version**(*self*)

Loads the last version of the file which is available on the storage. If the monitor is in user-view mode, loads last version, which was checked by specified user.

**Return Value**

most recent content of the file which is on the storage.

(*type=Content*)

**Inherited from object**

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,  
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,  
`__str__()`, `__subclasshook__()`

**5.4.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

**5.5 Class Diffable**

object —  
**changemonitor.model.Diffable**

Interface-like class. A class, which inherits this interface, has to implement the method for diffing two objects and choose of a diff algorithm.

### 5.5.1 Methods

**diff\_to**(*self*, *obj*)

#### *Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_init\_\_(),  
 \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(),  
 \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

### 5.5.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 5.6 Class Content

object —

changemonitor.model.Diffable —  
   **changemonitor.model.Content**

Content of web document in one version.

Implements Diffable interface to get possibility to diff contents to each other. Differ algorithm is choosen automatically.

Implementation detail: wrapper of GridOut instance.

## 5.6.1 Methods

```
__init__(self, gridout)
```

---

Create new instance of content.

WARNING: Do not instantiate this class by yourself, this is done by  
File methods.

@param gridout: gridout instance which was retrieved by GridFS.  
@type gridout: gridfs.grid\_file.GridOut

Overrides: object.\_\_init\_\_

```
__getattr__(self, name)
```

```
diff_to(self, other)
```

---

Creates diff of self and given Content object and returns unicode string  
representing the computed diff: \$ diff-algo self obj

**Parameters**

other: diffed content  
(type=Content)

**Return Value**

computed diff  
(type=unicode)

Overrides: changemonitor.model.Diffable.diff\_to

```
__repr__(self)
```

str(x)

Overrides: object.\_\_repr\_\_ extit(inherited documentation)

```
__str__(self)
```

str(x)

Overrides: object.\_\_str\_\_ extit(inherited documentation)

*Inherited from object*

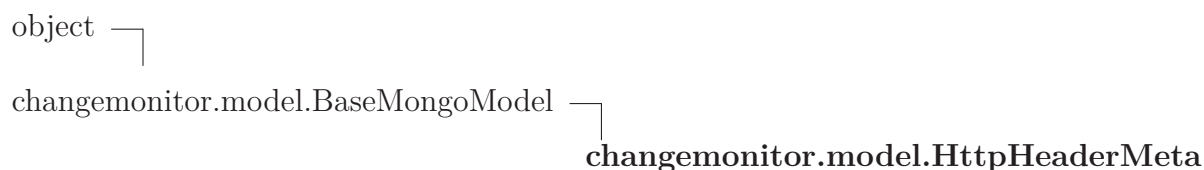
```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),  
__reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()
```



### 5.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 5.7 Class `HTTPHeaderMeta`



Model for HTTP header metadata.

```

header = {
    timestamp: 1341161610.287
    response_code: 200
    last_modified: cosi
    etag: P34lkdfk32jrlkjdfpoqi3
    uid: "rrs_university"
    url+index: "http://www.cosi.cz"
    content: object_id
}
  
```

### 5.7.1 Methods

<code>__init__(self, connection, uid, database)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)
--

---

**get\_by\_time**(*self*, *url*, *timestamp*, *last\_available=False*)**Parameters**

**url:** url of resource to search for  
(*type=string*)

**timestamp:** (*type=int*)

**last\_available:** (*type=Bool*)

**Return Value**

http header metadata of 'url'/None if not found  
(*type=*)

**To Do:** docstring Get record of 'url' with 'timestamp' from HeaderMeta database

---

**get\_by\_version**(*self*, *url*, *version*, *last\_available=False*)**Parameters**

**url:** url of resource to get from db  
(*type=string*)

**version:** version number of record, ...TODO: x,-x,1,0,-1  
(*type=int*)

**last\_available:** (*type=bool*)

**Return Value**

http header metadata of 'url'/None if not found  
(*type=*)

**To Do:** docstring Get 'version' of 'url' from HeaderMeta database

---

**save\_header**(*self*, *url*, *response\_code*, *fields*, *content\_id*)

Save http header into HttpHeaderMeta database

**Parameters**

**url:** url of checked resource

**response\_code:** response code of web server

**fields:** fields of http response

**content\_id:** content-id field of http response

**Return Value**

saved object

```
last_checked(self, url)
```

Get time when 'url' was last checked

WARNING!

if None is returned, then 'url' was never checked  
that should never happen, as 'url' is always checked  
in constructor of MonitoredResource  
but it's possible that the header is not saved  
because of an error, eg. timeout or other

@param url: url of resource checked

@type url: string

@returns: time of last check

@rtype: HTTPDateTime

```
check_uid(self)
```

### *Inherited from object*

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

### 5.7.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 6 Module changemonitor.resolver

**Date:** \$25.6.2012 12:12:44\$

**Author:** Stanislav Heller

### 6.1 Variables

Name	Description
<code>__modulename__</code>	<b>Value:</b> "resolver"
<code>__email__</code>	<b>Value:</b> "xhelle03@stud.fit.vutbr.cz"

### 6.2 Class Resolver

object └─ **changemonitor.resolver.Resolver**

System pro zajisteni funkcionality "zmenilo se neco na dane URL?". Zde se bude osetrovat mnoho vyjimek s http hlavickami, s md5, content-length atp.

Implikace: `changed(HTTP:content-length)` -> content changed `changed(HTTP:md5)` -> content changed `changed(HTTP:response code)` -> it depends.. `changed(HTTP:last-modified)` -> doesn't matter

#### 6.2.1 Methods

```
__init__(self, storage, timeout=10)
```

`x.__init__(...)` initializes x; see `help(type(x))` for signature

Overrides: `object.__init__` extit(inherited documentation)

```
resolve(self, url)
```

#### *Inherited from object*

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

#### 6.2.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

### 6.3 Class Rule

object —  
           **changemonitor.resolver.Rule**

Pravidlo pro resolver. Možno se bude hodit nejake takove rozdeleni tech pravidel... proste rozdel a panuj.

#### 6.3.1 Methods

__call__( <i>self</i> )
Pokud __call__ vraci true, pak rule matchl.

#### *Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_init\_\_(),  
 \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(),  
 \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 6.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## Index

- changemonitor (*package*), 2–9
  - changemonitor.\_http (*module*), 10–13
    - changemonitor.\_http.HTTPDateTime (*class*), 10–13
  - changemonitor.diff (*module*), 14–17
    - changemonitor.diff.BinaryDiff (*class*), 15–16
    - changemonitor.diff.DocumentDiff (*class*), 14
    - changemonitor.diff.HtmlDiff (*class*), 16–17
    - changemonitor.diff.PlainTextDiff (*class*), 14–15
  - changemonitor.errors (*module*), 18–23
    - changemonitor.errors.ChangeMonitorError (*class*), 18–19
    - changemonitor.errors.DocumentHistoryNotAvailable (*class*), 20–21
    - changemonitor.errors.DocumentNotAvailable (*class*), 20
    - changemonitor.errors.DocumentTooLarge (*class*), 19–20
    - changemonitor.errors.NotSupportedYet (*class*), 21–22
    - changemonitor.errors.UidError (*class*), 22–23
  - changemonitor.model (*module*), 24–34
    - changemonitor.model.BaseMongoModel (*class*), 24
    - changemonitor.model.Content (*class*), 30–32
    - changemonitor.model.Diffable (*class*), 29–30
    - changemonitor.model.File (*class*), 27–29
    - changemonitor.model.HttpHeaderMeta (*class*), 32–34
    - changemonitor.model.Storage (*class*), 24–27
  - changemonitor.Monitor (*class*), 5–9
    - changemonitor.Monitor.allow\_large\_documents (*method*), 8
    - changemonitor.Monitor.check\_multi (*method*), 8
    - changemonitor.Monitor.check\_uid (*method*), 8
    - changemonitor.Monitor.get (*method*), 7
  - changemonitor.MonitoredResource (*class*), 2–5
    - changemonitor.MonitoredResource.available (*method*), 5
    - changemonitor.MonitoredResource.check (*method*), 4
    - changemonitor.MonitoredResource.get\_diff (*method*), 4
    - changemonitor.MonitoredResource.get\_last\_version (*method*), 4
    - changemonitor.MonitoredResource.get\_version (*method*), 4
    - changemonitor.MonitoredResource.last\_checked (*method*), 5
  - changemonitor.resolver (*module*), 35–36
    - changemonitor.resolver.Resolver (*class*), 35–36
    - changemonitor.resolver.Rule (*class*), 36