

# IdMappingRetrieval

March 24, 2012

---

Annotation

*The Annotation class*

---

## Description

Package:

**Class Annotation**

[Object](#)

~~|

~~+--Annotation

**Directly known subclasses:**

[AnnotationAffx](#), [AnnotationDavid](#), [AnnotationDavidCsv](#), [AnnotationEnsembl](#), [AnnotationEnsemblCsv](#), [AnnotationEnvision](#), [AnnotationNetAffx](#)

public abstract static class **Annotation**

extends [Object](#)

This is the base annotation class from which the concrete classes like [AnnotationAffx](#), [AnnotationEnvision](#) etc. are derived

## Usage

```
Annotation(cacheFolderName="", primaryColumn="From", secondaryColumn="To", swap=
```

## Arguments

cacheFolderName

The path to a service caching directory for a given Annotation object. The path is relative to the caching subsystem root directory. Default is 'Affymetrix'

primaryColumn

Primary column to be retrieved from a data frame obtained for a given service when getIdMap() on a given annotation object is called. Default is 'From'.

secondaryColumn

Secondary column(s) to be retrieved from a data frame obtained for a given service when getIdMap() on a given annotation object is called. Default is 'To'.

swap	A <code>logical</code> indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the <code>getIdMap()</code> call. Default is <code>FALSE</code> .
species	A <code>character</code> vector or <code>NA</code> indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If <code>NA</code> , no filtering is performed. Default is "Homo sapiens".
verbose	if <code>TRUE</code> enables diagnostic messages. Default is <code>FALSE</code> .
...	Additional parameters

## Fields and Methods

### Methods:

<code>getArrayType</code>	Select the array type.
<code>getArrayTypes</code>	Get all available microarray chip types.
<code>getCredentials</code>	Retrieve credentials for a given service.
<code>getDataFrame</code>	Get the entire data set available from a particular service in a form of a data frame.
<code>getFolderName</code>	Get caching folder name for a given Annotation object.
<code>getIdMap</code>	Get an IdMap object using the data retrieved by a particular service represented by annotation object.
<code>getRoot</code>	Get the path to service caching directory.
<code>getServiceRoot</code>	Get a root directory for a particular annotation object.
<code>init</code>	Init annotation subsystem.
<code>setCredentials</code>	Create and store the credentials information for given service.
<code>setOptions</code>	Set the parameters for an annotation object.

### Methods inherited from Object:

`$`, `$<-`, `[]`, `[[<-`, `as.character`, `attach`, `attachLocally`, `clearCache`, `clone`, `detach`, `equals`, `extend`, `finalize`, `gc`, `getEnvironment`, `getFields`, `getInstantiationTime`, `getStaticInstance`, `hasField`, `hashCode`, `ll`, `load`, `objectSize`, `print`, `registerFinalizer`, `save`

### Author(s)

Alex Lisovich, Roger Day

---

AnnotationAffx      *The AnnotationAffx class*

---

## Description

Package:

**Class AnnotationAffx**

Object

~~|

~~+--Annotation

~~~~~|

~~~~~+--AnnotationAffx

**Directly known subclasses:**[AnnotationNetAffx](#)

```
public static class AnnotationAffx
  extends Annotation
```

The AnnotationAffx class encapsulates the functionality allowing to retrieve data from the Affymetrix annotation data online repository through the `getIdMap()` and `getDataFrame()` calls.

**Usage**

```
AnnotationAffx(cacheFolderName="Affymetrix", primaryColumn="Probe.Set.ID", second...
```

**Arguments**

cacheFolderName

The path to a service caching directory for a given AnnotationAffx object. The path is relative to the caching subsystem root directory. Default is 'Affymetrix'

primaryColumn

Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when `getIdMap()` is called. Default is 'Probe.Set.ID'.

secondaryColumn

Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when `getIdMap()` is called. Default is 'SwissProt'.

swap

Logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the `getIdMap()` call. Default is `TRUE`.

...

Additional parameters, see [Annotation](#).

**Fields and Methods****Methods:**

```
getProbesetList  Get Affymetrix probeset ID list for a given array type.
setCredentials  Create and store the credentials information for Affymetrix service.
```

**Methods inherited from Annotation:**

```
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap,
getRoot, getServiceRoot, init, readDF, setCredentials, setOptions
```

**Methods inherited from Object:**

```
$, $<-, [], [<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, final-
ize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll,
load, objectSize, print, registerFinalizer, save
```

**Author(s)**

Alex Lisovich, Roger Day

---

AnnotationDavid      *The AnnotationDavid class*

---

## Description

Package:

**Class AnnotationDavid**

Object

~~|

~~+--Annotation

~~~~~|

~~~~~+--AnnotationDavid

### Directly known subclasses:

[AnnotationDavidCsv](#)

public static class **AnnotationDavid**

extends [Annotation](#)

The AnnotationDavid class encapsulates the functionality allowing to retrieve data from the DAVID online query system through the getIdMap() and getDataFrame() calls on this AnnotationDavid object.

## Usage

```
AnnotationDavid(cacheFolderName="DAVID", primaryColumn="AFFYMETRIX_3PRIME_IVT_ID
```

## Arguments

cacheFolderName

The symbolic name of a service represented by a given AnnotationDavid object.

primaryColumn

Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'AFFYMETRIX\_3PRIME\_IVT\_ID'.

secondaryColumn

Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'UNIPROT\_ACCESSION'.

swap

A [logical](#) indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call. Default is [TRUE](#).

...

Additional parameters, see [Annotation](#).

## Fields and Methods

### Methods:

*No public methods defined.*

### Methods inherited from Annotation:

getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

**Methods inherited from Object:**

\$, \$<-, [], [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**

Alex Lisovich, Roger Day

**Examples**

```
## Not run:
Annotation$init();
#create DAVID annotation object
annObj<-AnnotationDavid"DAVID",species="Homo sapiens");

## End(Not run)
```

---

AnnotationDavidCsv *The AnnotationDavidCsv class*

---

**Description**

Package:

**Class AnnotationDavidCsv**

[Object](#)

~~|

~~+--[Annotation](#)

~~~~~|

~~~~~+--[AnnotationDavid](#)

~~~~~|

~~~~~+--[AnnotationDavidCsv](#)

**Directly known subclasses:**

public static class **AnnotationDavidCsv**

extends [AnnotationDavid](#)

The AnnotationDavidCsv class encapsulates the functionality allowing to retrieve data from the DAVID backend annotation file system through the Annotation.getIdMap() and Annotation.getDataFrame() calls on this object. The DAVID backend annotation file system requires to submit the request through the e-mail for conversion of one ID type to another. Within a day, the user receives an email reply containing the link to the resulting conversion file. The results are returned in a form of a tab delimited headerless file containing the match pairs, one pair per line. The AnnotationDavidCsv object encapsulates the functionality allowing to interactively choose the resulting files and convert it into a data frame during the Annotation.getIdMap() and Annotation.getDataFrame() calls on the AnnotationDavidCsv object.

**Usage**

```
AnnotationDavidCsv(cacheFolderName="DavidCsv", primaryColumn="AFFY_ID", secondaryColumn="UNIPROT_ACCESSION", swap=TRUE, df_filename="results.txt", ...)
```

**Arguments**

`cacheFolderName` The symbolic name of a service represented by a given `AnnotationDavidCsv` object.

`primaryColumn` Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when `getIdMap()` is called. Default is 'AFFY\_ID'.

`secondaryColumn` Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when `getIdMap()` is called. Default is 'UNIPROT\_ACCESSION'.

`swap` Logical indicating if primary and secondary column(s) need to be swapped at the end of the `IdMap` retrieval during the `getIdMap()` call. Default is `TRUE`.

`df_filename` Character string or `NULL`. In the first case the character string contains the name of conversion results file and in the second case the file name is determined interactively through the Open File dialog during the call to `getIdMap()` or `getDataFrame()` on the `AnnotationDavidCsv` object.

... Additional parameters, see [Annotation](#).

**Fields and Methods****Methods:**

*No public methods defined.*

**Methods inherited from AnnotationDavid:**

`getColumns`, `readDF`

**Methods inherited from Annotation:**

`getArrayType`, `getArrayTypes`, `getColumns`, `getCredentials`, `getDataFrame`, `getFolderName`, `getIdMap`, `getRoot`, `getServiceRoot`, `init`, `readDF`, `setCredentials`, `setOptions`

**Methods inherited from Object:**

`$`, `$<-`, `[]`, `[[<-`, `as.character`, `attach`, `attachLocally`, `clearCache`, `clone`, `detach`, `equals`, `extend`, `finalize`, `gc`, `getEnvironment`, `getFields`, `getInstantiationTime`, `getStaticInstance`, `hasField`, `hashCode`, `ll`, `load`, `objectSize`, `print`, `registerFinalizer`, `save`

**Author(s)**

Alex Lisovich, Roger Day

## Description

Package:

**Class AnnotationEnsembl**

Object

~~|

~~+--Annotation

~~~~~|

~~~~~+--AnnotationEnsembl

### Directly known subclasses:

[AnnotationEnsemblCsv](#)

public static class **AnnotationEnsembl**

extends [Annotation](#)

The AnnotationEnsembl class encapsulates the functionality allowing to retrieve data from the Ensembl BioMart online query system using biomaRt R package through the Annotation.getIdMap() and Annotation.getDataFrame() calls on this object.

## Usage

```
AnnotationEnsembl(cacheFolderName="Ensembl", primaryColumn=c("uniprot_swissprot_
```

## Arguments

cacheFolderName

The symbolic name of a service represented by a given AnnotationEnsembl object.

primaryColumn

Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Tremb accessions in separate columns, it is possible to retrieve either or them or merge them together by explicitly specifying the set of columns to be merged. Default is c('uniprot\_swissprot\_accession','uniprot\_sptrembl').

secondaryColumn

secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If **NA** (default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitly.

swap

A **logical** indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call. Default is **TRUE**.

species

Character vector or **NA** indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If **NA**, no filtering is performed. Default is 'hsapiens\_gene\_ensembl'.

`full.merge` A `logical` indicating which version of primary columns merging algorithm to use. If `TRUE` (default), all unique pairs `<probeset ID, SwissProt>` and `<probeset ID, Trembl>` are generated, and if `FALSE`, only those pairs from `<probeset ID, Trembl>` for which Uniprot ID is not present in `<probeset ID, SwissProt>` pairs are included.

`...` Additional parameters, see [Annotation](#).

## Fields and Methods

### Methods:

*No public methods defined.*

### Methods inherited from Annotation:

`getArrayType`, `getArrayTypes`, `getColumns`, `getCredentials`, `getDataFrame`, `getFolderName`, `getIdMap`, `getRoot`, `getServiceRoot`, `init`, `readDF`, `setCredentials`, `setOptions`

### Methods inherited from Object:

`$`, `$<`, `[[`, `[[<`, `as.character`, `attach`, `attachLocally`, `clearCache`, `clone`, `detach`, `equals`, `extend`, `finalize`, `gc`, `getEnvironment`, `getFields`, `getInstantiationTime`, `getStaticInstance`, `hasField`, `hashCode`, `ll`, `load`, `objectSize`, `print`, `registerFinalizer`, `save`

## Author(s)

Alex Lisovich, Roger Day

## Examples

```
## Not run:
Annotation$init();
#create Ensembl annotation object
annObj<-AnnotationEnsembl(species="hsapiens_gene_ensembl");

## End(Not run)
```

---

AnnotationEnsemblCsv

*The AnnotationEnsemblCsv class*

---

## Description

Package:

**Class AnnotationEnsemblCsv**

`Object`

~~|

~~+--`Annotation`

~~~~~|

~~~~~+--`AnnotationEnsembl`

~~~~~|

~~~~~+--`AnnotationEnsemblCsv`



**Directly known subclasses:**

```
public static class AnnotationEnsemblCsv
extends AnnotationEnsembl
```

The AnnotationEnsemblCsv class encapsulates the functionality allowing to retrieve data from the Ensembl interactive online query system. The ID matching information filtered on species and the microarray chip type is retrieved as comma delimited csv file. The AnnotationEnsemblCsv object encapsulates the functionality allowing to interactively choose the Ensembl query results csv file and convert it into a data frame during the getIdMap() and getDataFrame() calls on the AnnotationEnsemblCsv object.

**Usage**

```
AnnotationEnsemblCsv(cacheFolderName="EnsemblCsv", primaryColumn=c("UniProt.SwissProt", "Trembl"))
```

**Arguments**

|                 |   |
|-----------------|---|
| cacheFolderName | The symbolic name of a service represented by a given AnnotationEnsembl object.   |
| primaryColumn   | Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Trembl accessions in separate columns, it is possible to retrieve either or them or merge them together by explicitly specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession', 'uniprot_sptrembl').   |
| secondaryColumn | secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If <b>NA</b> (default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitly. |
| swap            | A <b>logical</b> indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call. Default is <b>TRUE</b> .  |
| full.merge      | A <b>logical</b> indicating which version of primary columns merging algorithm to use. If <b>TRUE</b> (default), all unique pairs <probeset ID, SwissProt> and <probeset ID, Trembl> are generated, and if <b>FALSE</b> , only those pairs from <probeset ID, Trembl> for which Uniprot ID is not present in <probeset ID, SwissProt> pairs are included.   |
| df_filename     | Character string or NULL. In the first case the character string contains the name of conversion results file and in the second case the file name is determined interactively through the Open File dialog during the call to Annotation.getIdMap() or Annotation.getDataFrame() on the AnnotationEnsemblCsv object.   |
| ...             | Additional parameters, see <a href="#">Annotation</a> .   |

**Fields and Methods****Methods:**

*No public methods defined.*

**Methods inherited from AnnotationEnsembl:**

getColumn, getColumn1, getColumn2, readDF

**Methods inherited from Annotation:**

getArrayType, getArrayTypes, getColumn, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

**Methods inherited from Object:**

\$, \$<-, [], [<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**

Alex Lisovich, Roger Day

**Examples**

```
## Not run:
Annotation$init();
#create Ensembl annotation object
annObj<-AnnotationEnsemblCsv(cacheFolderName="EnsemblCsv");

## End(Not run)
```

---

AnnotationEnvision *The AnnotationEnvision class*

---

**Description**

Package:

**Class AnnotationEnvision**

```
Object
~~|
~~+--Annotation
~~~~~|
~~~~~+--AnnotationEnvision
```

**Directly known subclasses:**

```
public static class AnnotationEnvision
extends Annotation
```

The AnnotationEnvision class encapsulates the functionality allowing to retrieve data from the Envision online query system. The Envision online query system allows to retrieve the ID matching information accompanied by multiple attributes like species and the microarray chip type in the

form of the xml file. The AnnotationEnvision class encapsulates the functionality allowing to filter the Envision query results on species and microarray type attributes and convert it into a data frame during the getIdMap() and getDataFrame() calls on the AnnotationEnvision object.

## Usage

```
AnnotationEnvision(cacheFolderName="EnVision", primaryColumn=c("UniProt.SwissProt"))
```

## Arguments

|                 |  |
|-----------------|--|
| cacheFolderName | The symbolic name of a service represented by a given AnnotationEnsembl object.  |
| primaryColumn   | Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Trembl accessions in separate columns, it is possible to retrieve either or them or merge them together by explicitly specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession','uniprot_sptrembl').   |
| secondaryColumn | secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If NA (default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitly. |
| swap            | A <code>logical</code> indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call. Default is <code>TRUE</code> .   |
| species         | Character vector or NA indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If NA, no filtering is performed. Default is 'Homo sapiens'.   |
| full.merge      | A <code>logical</code> indicating which version of primary columns merging algorithm to use. If <code>TRUE</code> (default), all unique pairs <probeset ID, SwissProt> and <probeset ID, Trembl> are generated, and if <code>FALSE</code> , only those pairs from <probeset ID, Trembl> for which Uniprot ID is not present in <probeset ID, SwissProt> pairs are included.  |
| ...             | Additional parameters, see <a href="#">Annotation</a> .  |

## Fields and Methods

### Methods:

*No public methods defined.*

### Methods inherited from Annotation:

getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

### Methods inherited from Object:

\$. \$<- [, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**

Alex Lisovich, Roger Day

**Examples**

```
## Not run:
Annotation$init();
#create Envision annotation object
annObj<-AnnotationEnvision(species="Homo sapiens");

## End(Not run)
```

---

AnnotationNetAffx *The AnnotationNetAffx class*

---

**Description**

Package:

**Class AnnotationNetAffx**

[Object](#)

```
~~|
~~+--Annotation
~~~~~|
~~~~~+--AnnotationAffx
~~~~~|
~~~~~+--AnnotationNetAffx
```

**Directly known subclasses:**

```
public static class AnnotationNetAffx
extends AnnotationAffx
```

The AnnotationNetAffx class encapsulates the functionality allowing to retrieve data from the NetAffx batch query system through the Annotation.getIdMap() and Annotation.getDataFrame() calls on this object. The NetAffx batch query system requires to submit the probeset IDs by providing text files in a special format, maximum 10000 IDs per file. The results are returned in a form of a tab delimited text file, one file per submission, so the query results for a whole array are presented by a set of such files. The AnnotationNetAffx object encapsulates the functionality allowing to interactively choose the set of result files and merge them into a single data frame during the Annotation.getIdMap() and Annotation.getDataFrame() calls on the AnnotationNetAffx object.

**Usage**

```
AnnotationNetAffx(cacheFolderName="NetAffxCsv", primaryColumn="Probe.Set.ID", se
```

**Arguments**

|                 |  |
|-----------------|--|
| cacheFolderName | The symbolic name of a service represented by a given AnnotationNetAffx object.  |
| primaryColumn   | Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'Probe.Set.ID'.  |
| secondaryColumn | Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'SwissProt'.   |
| swap            | A <i>logical</i> indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call. Default is <i>TRUE</i> .   |
| df_filename     | Character vector, character string or <i>NULL</i> . In the first case the character vector contains the names of a resulting file set, in the second, the character string contains the name of directory in which the files are stored, and in the third the file set is determined interactively through the Open File dialog during the call to getIdMap() or getDataFrame() on the AnnotationAffxCsv object. |
| ...             | Additional parameters, see <i>Annotation</i> .   |

**Fields and Methods****Methods:**

`createSubmission` Create a set of files for NetAffx batch query.

**Methods inherited from AnnotationAffx:**

getColumn, getProbesetList, readDF, setCredentials

**Methods inherited from Annotation:**

getArrayType, getArrayTypes, getColumn, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

**Methods inherited from Object:**

\$, \$<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**

Alex Lisovich, Roger Day

---

IdMappingRetrieval-package

*Id Mapping Retrieval.*

---

**Description**

Id Mapping Retrieval.

**Details**

Package: IdMappingRetrieval  
Type: Package  
Version: 0.99.0  
Date: 2011-06-26  
License: GPL-2  
LazyLoad: yes

### Author(s)

Alex Lisovich, Roger Day

---

ServiceManager      *The ServiceManager class*

---

### Description

Package:  
**Class ServiceManager**

[Object](#)  
~~|  
~~+--ServiceManager

#### Directly known subclasses:

```
public static class ServiceManager  
extends Object
```

ServiceManager class serves as a container for a set of specialized service objects and provides the means for (optionally interactive) handling of such a set simplifying the process of data retrieval from a variety of resources in a batch mode.

### Usage

```
ServiceManager (services=list(), ...)
```

### Arguments

|          |   |
|----------|---|
| services | Annotation services to be contained within the ServiceManager object. Default is <a href="#">NULL</a> . |
| ...      | Not used.   |

### Fields and Methods

#### Methods:

|                                 |   |
|---------------------------------|---|
| <code>addServices</code>        | Add services to the ServiceManager object.  |
| <code>getDataFrameList</code>   | Collect raw data data from various online query systems utilizing Annotation service fun  |
| <code>getDefaultServices</code> | Get the default services available within the annotation system.                          |
| <code>getIdMapList</code>       | Collect ID mapping data from various online query systems utilizing Annotation service    |
| <code>getServices</code>        | Get list of annotation services encapsulated within the particular ServiceManager object. |
| <code>setServices</code>        | Set the list of services for a ServiceManager object.                                     |

**Methods inherited from Object:**

\$, \$<-, [], [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**

Alex Lisovich, Roger Day



# Index

## \*Topic classes

Annotation, 1  
AnnotationAffx, 2  
AnnotationDavid, 4  
AnnotationDavidCsv, 5  
AnnotationEnsembl, 6  
AnnotationEnsemblCsv, 8  
AnnotationEnvision, 10  
AnnotationNetAffx, 12  
ServiceManager, 15

## \*Topic package

IdMappingRetrieval-package,  
13

addServices, 16  
Annotation, 1, 2–13  
AnnotationAffx, 1, 2, 12  
AnnotationDavid, 1, 4, 5  
AnnotationDavidCsv, 1, 4, 5  
AnnotationEnsembl, 1, 6, 8, 9  
AnnotationEnsemblCsv, 1, 7, 8  
AnnotationEnvision, 1, 10  
AnnotationNetAffx, 1, 3, 12

character, 2  
createSubmission, 13

FALSE, 2, 8, 9, 11

getArrayType, 2  
getArrayTypes, 2  
getCredentials, 2  
getDataFrame, 2  
getDataFrameList, 16  
getDefaultServices, 16  
getFolderName, 2  
getIdMap, 2  
getIdMapList, 16  
getProbesetList, 3  
getRoot, 2  
getServiceRoot, 2  
getServices, 16

IdMappingRetrieval  
(*IdMappingRetrieval-package*),  
13

IdMappingRetrieval-package, 13  
init, 2

logical, 2, 4, 7–9, 11, 13

NA, 2, 7, 9, 11  
NULL, 6, 13, 15

Object, 1, 2, 4, 5, 7, 8, 10, 12, 15

ServiceManager, 15  
setCredentials, 2, 3  
setOptions, 2  
setServices, 16

TRUE, 2–4, 6, 7, 9, 11, 13