RTFPInbox Data Structure Manual

Jordan Boyle

Virginia Tech

Notice To Reader

This is the User Manual for the RTFPInbox Data Structure. At the time of writing this Data Structure is in Version 1.5.0. If the Version in this Manual does not match the Version being utilized by the User please refer to the Author of this Manual or the Maintainer of the Structure if the Author is no longer available.

RTFPInbox Data Structure Overview

This Data Structure was designed for the Use & Abuse of Personal Information Study by members of the Raytheon Technologies Fellowship Program. Note that in this Document User means the person interacting with the Data and the Person or Inbox Owner refers to the Identity associated with the Inbox.

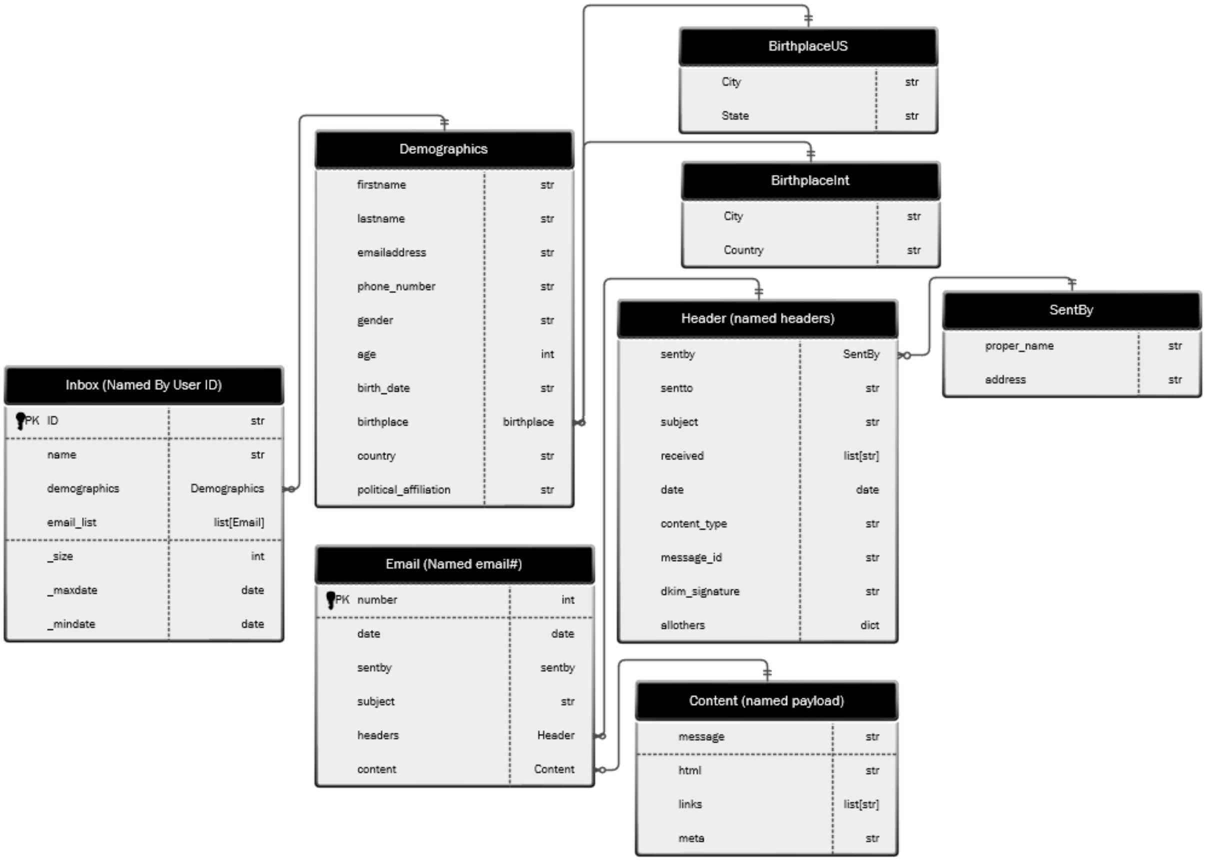
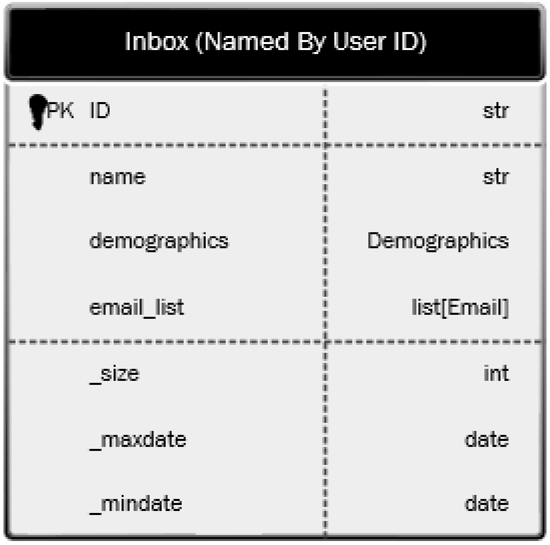


Figure 1 RTFPINBOX Data Structure Block Diagram.

Section 1: Inbox

Overview



Inbox (Named By User ID)		
PK	ID	str
	name	str
	demographics	Demographics
	email_list	list[Email]
	_size	int
	_maxdate	date
	_mindate	date

Figure 2 The Inbox Object

Each object in this Structure represents one person's Inbox. The inbox will be known by a unique ID that will be assigned to it upon creation. This ID is the Primary Key for identifying the Inbox, and the User will note that it is identified as such in Figure 1. The Inbox contains the ID of the person it is associated with and their name, both as strings. It contains the Demographics of that person as a Demographics object, and a list of Email objects that makes up the contents of that person's inbox. It has 3 protected values, size, the number of emails in the inbox as an int, maxdate, the date of the last email received by the inbox, and mindate, the date of the first email received by the inbox. In order for this object to be created only the ID and the Name must be provided. The other values can be added after creation and the protected values are adjusted using the add or remove email methods. With the exception of the protected values, values in the Inbox Object can be accessed via the syntax:

```
nameofinbox.valuerequested
```

Where `nameofinbox` is the name of the inbox and `valuerequested` is the value the user is trying to access.

Methods

The Inbox object contains 14 methods.

AddEmail(message):

The first method in the Inbox Structure is the AddEmail method. This method creates an email object and adds it to the inbox. It will also update the inbox size and the min and max date accordingly. It requires a single message object, and it will throw an error if the input is not a single message object. This method is called using the syntax:

```
nameofinbox.addemail(message)
```

With `nameofinbox` being the name of the inbox and `message` being the message object the user is trying to add to the email list. Normally this method is run in a for loop with a list of messages.

BySender(sender):

This method finds and returns list of messages sent by sender. It takes a string, which can either be the sender's email address or their name, and it searches the email list and returns a list of emails sent by the sender the user inputted. This method is called using the syntax:

```
nameofinbox.bysender(sender)
```

With `nameofinbox` being the name of the inbox and `sender` being the sender the user is searching for.

ByDate(start_date, end_date, inc):

This method finds and returns list of messages between 2 dates. If the Boolean value inc is true this method is inclusive, otherwise it is non-inclusive. It will throw errors if the inputs are not dates or if the dates are out of bounds. This method is called using the syntax:

```
nameofinbox.bydate(start_date, end_date, inc)
```

With `nameofinbox` being the name of the inbox, `start_date` being the start date of the search parameters, `end_date` being the end date of the search parameters and `inc` being the Boolean that triggers inclusive or non-inclusive modes.

ByNumber(start_num, end_num, inc):

This method finds and returns list of messages between 2 indices. If the Boolean value inc is true this method is inclusive, otherwise it is non-inclusive. It will throw errors if the inputs are out of bounds. This method is called using the syntax:

```
nameofinbox.bynumber(start_num, end_num, inc)
```

With `nameofinbox` being the name of the inbox, `start_num` being the starting index of the search parameters, `end_num` being the ending index of the search parameters and `inc` being the Boolean that triggers inclusive or non-inclusive modes.

BySubject(keyword):

This method finds and returns list of messages containing keyword/s. It takes either a string or a list of strings. If the user inputs a string this method will search the subjects of the emails in the email list and return a list of all the emails whose subjects contain the string inputted by the user. If the user inputs a list of strings, it will return a list of messages whose subjects contained all of the strings in the list. This method is called using the syntax:

```
nameofinbox.bysubject(keyword)
```

With `nameofinbox` being the name of the inbox and `keyword` being the string or list the user is searching for.

BySubjectSynonym(keyword):

This method finds and returns list of messages containing synonyms of the inputted keyword. It takes a single word string and will throw an error if the string is multiple words long. This method is called using the syntax:

```
nameofinbox.bysubjectsynonym(keyword)
```

With `nameofinbox` being the name of the inbox and `keyword` being the string the user is searching with.

InboxHeaderSearch(keyword):

This method finds and returns list of messages containing the header inputted by the user. It takes a string. This method is called using the syntax:

```
nameofinbox.inboxheadersearch(keyword)
```

With `nameofinbox` being the name of the inbox and `keyword` being the header the user is searching for.

RemoveEmail(Index):

This method removes an email located at the inputted index in the email list. It will also update the inbox size and the min and max dates accordingly. It will throw an error if the index is out of bounds. This method is called using the syntax:

```
nameofinbox.removeemail(index)
```

With `nameofinbox` being the name of the inbox and `index` being the number of the message object that user is trying to remove from the email list.

GetSize():

This method returns the number of Emails in the email list. It is called using the syntax:

```
nameofinbox.getsize()
```

With `nameofinbox` being the name of the inbox.

IncreaseSize():

This method increases the inbox value `_size`. Since size is a protected value the user is not supposed to interact with it directly and instead use the methods here. This method InceaseSize() is only called by the AddEmail method and **should not be called by the user.**

DecreaseSize():

This method decreases the inbox value `_size`. Since size is a protected value the user is not supposed to interact with it directly and instead use the methods here. This method DeceaseSize() is only called by the RemoveEmail method and **should not be called by the user.**

SetMaxMinDate():

This method checks to ensure that all of the dates of all of the emails in the email list have been properly inputted and will through an error if any haven't been. It will also find the maximum and minimum dates in the email list and assign them to the protected values `_maxdate` and `_mindate`. This method is called by both Add and Remove Email and can, but does not need to be, called by the user.

GetMaxDate():

This method returns latest date of the emails received by the inbox. It is called using the syntax:

```
nameofinbox.getmaxdate()
```

With `nameofinbox` being the name of the inbox.

GetMinDate():

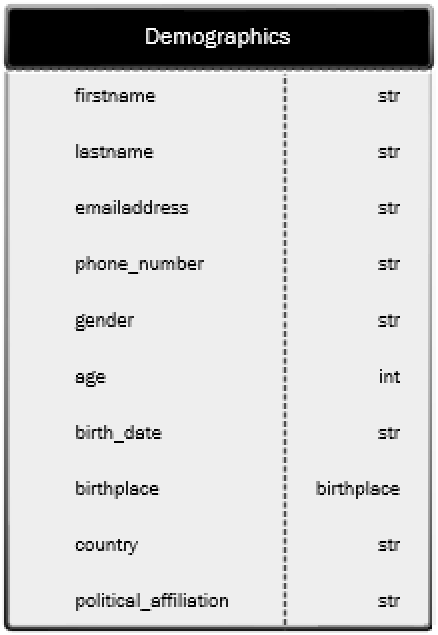
This method returns earliest date of the emails received by the inbox. It is called using the syntax:

```
nameofinbox.getmindate()
```

With `nameofinbox` being the name of the inbox.

Section 2: Demographics

Overview



Demographics	
firstname	str
lastname	str
emailaddress	str
phone_number	str
gender	str
age	int
birth_date	str
birthplace	birthplace
country	str
political_affiliation	str

Figure 3 The Demographics Object

This object contains the Demographic information of the owner of the Inbox. It contains their first and last name, their email address, phone number, gender, birthday, country and political affiliation as strings, the age as an integer, and the birthplace as a Birthplace object (See Section 3: Birthplace). All of these values are needed upon creation of this object and if any are unknown, they should be defined as None. With the exception of birthplace (See Section 3: Birthplace), values in the Demographics Object can be accessed via the syntax:

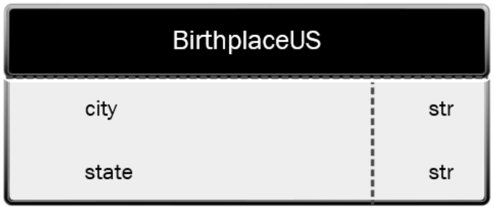
```
nameofinbox.demographics.valuerequested
```

Where `nameofinbox` is the name of the inbox and `valuerequested` is the value the user is trying to access.

There are currently no methods for this object.

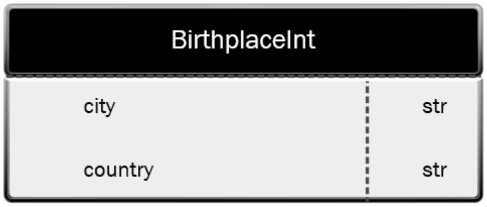
Section 3: Birthplace

Overview



BirthplaceUS	
city	str
state	str

Figure 4 The US Birthplaces object



BirthplaceInt	
city	str
country	str

Figure 5 The International Birthplaces object

This object contains the Birthplace information for the Demographics object. It contains the city and the state, for US birthplaces, or the Country, for international birthplaces, where the owner of the inbox was born as strings. This object was created to make it easier for the user to access the values of the inbox owner's birthplace without having to parse strings. Values in the Birthplace Object can be accessed via the syntax:

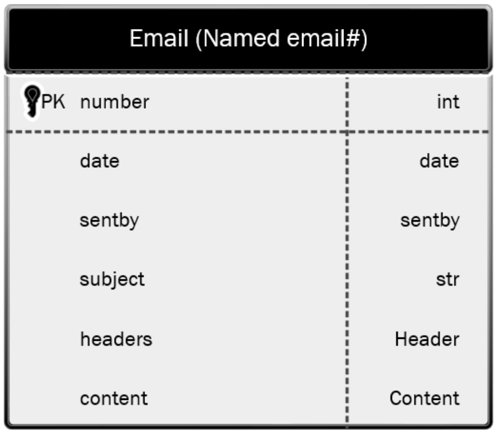
```
nameofinbox.demographics.birthplace.valuerequested
```

Where `nameofinbox` is the name of the inbox and `valuerequested` is the value the user is trying to access.

There are currently no methods for this object.

Section 4: Email

Overview



Email (Named email#)		
PK	number	int
	date	date
	sentby	sentby
	subject	str
	headers	Header
	content	Content

Figure 6 The Email Object

Each object of the Email Structure represents one email. These emails will be stored in a list inside the Inbox object. The emails will be identified by their number in the list. This number is the Primary Key for the Email object. This object also contains the date it was sent as a date object, the sender as a Sentby object (See Section 7: Sentby), the subject of the email as a string, the email's headers, as a Headers object (See Section 6: Headers), and the content of the email as a Content object (See Section 5: Content). Values in the Email object can be accessed using the syntax:

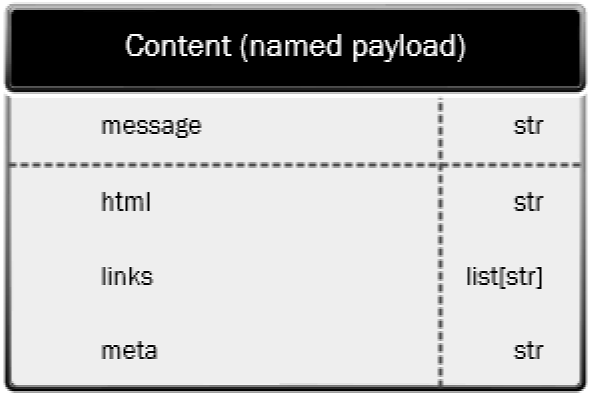
```
Nameofinbox.email_list[numberofemail].valuerequested
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `valuerequested` is the value the user is trying to access.

There are currently no methods for this object.

Section 5: Content

Overview



Content (named payload)	
message	str
html	str
links	list[str]
meta	str

Figure 7 The Content Object

The Content object will contain the contents of the Email Object. It contains the email message, html, and the metadata as a string and the links as a list of strings. Values in the Content object can be accessed using the syntax:

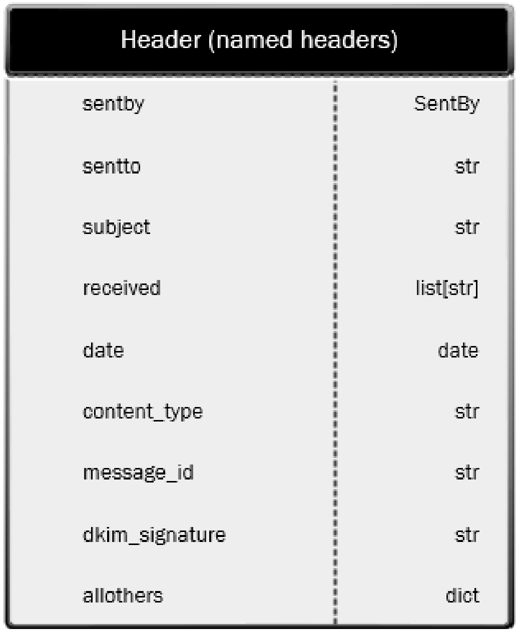
```
Nameofinbox.email_list[numberofemail].content.valuerequested
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `valuerequested` is the value the user is trying to access.

There are currently no methods for this object.

Section 6: Headers

Overview



Header (named headers)	
sentby	SentBy
sentto	str
subject	str
received	list[str]
date	date
content_type	str
message_id	str
dkim_signature	str
allothers	dict

Figure 8 The Headers Object

The Headers object will contain the headers of the Email Object. It contains the date it was sent as a date object, the sender as a Sentby object (See Section 7: Sentby), the subject of the email, the receiver of the email, the content type, message id and the DKIM Signature, as strings the received value as a list, and any other header values in a dictionary. Values in the Headers object can be accessed using the syntax:

```
Nameofinbox.email_list[numberofemail].headers.valuerequested
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `valuerequested` is the value the user is trying to access.

Methods

The Headers object contains 3 methods.

FormatDate(item):

This method takes the string date and turns it into a date object. It then returns the date object. This method is called using the syntax:

```
Nameofinbox.email_list[numberofemail].headers.formatdate(item)
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `item` is the string to be converted.

HeaderSearch(keyword):

This method takes a header key and returns the associated value. If the key in the email it returns None. This method is called using the syntax:

```
Nameofinbox.email_list[numberofemail].headers.headersearch(keyword)
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `keyword` is the header key being queried.

FormatSentby(sentby):

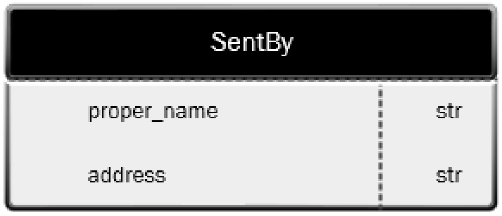
This method takes the string sentby value and turns it into a Sentby object (See Section 7: Sentby). It then returns the Sentby object. This method is called using the syntax:

```
Nameofinbox.email_list[numberofemail].headers.formatsentby(sentby)
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `sentby` is the string to be converted.

Section 7: Sentby

Overview



SentBy	
proper_name	str
address	str

Figure 9 The Sentby Object

This object contains the Sentby information for the Headers and Email objects. It contains the proper name and the email address of the email sender as strings. This object was created to make it easier for the user to access the values of the email sender without having to parse strings. Values in the Sentby object can be accessed using the syntax:

```
Nameofinbox.email_list[numberofemail].headers.sentby.valuerequested
```

Or

```
Nameofinbox.email_list[numberofemail].sentby.valuerequested
```

Where `nameofinbox` is the name of the inbox, `numberofemail` is the index of the email in the email list, and `valuerequested` is the value the user is trying to access.