Licas Services Guide

Version 2.29

[A guide on how to run and write a service using the licas framework]

Kieran Greer,
Email: kgreer@distributedcomputingsystems.co.uk.
http://licas.sourceforge.net

Table of Contents

1 Introduction

This document describes some of the custom applications that are available as part of the licas package. A number of services are included with the All-in-One GUI and fall into two broad categories. The first category is services that can be used as is, typically for personal or business use. The second category is to accommodate using the GUI for more scientific programs or testing. In that respect, it may be necessary to extend the base services with your own code and functionality, or to create your own configuration by including specific types and setups. But the GUI provides forms to allow you to do that and can also provide a view of any test results. This document provides descriptions of the services that are available with the installation. It also includes more technical sections, describing some of the service class code, as a light introduction to programming them further. The intention has been to provide a framework in which you can run you own services, where the easiest way to do that is to extend the licas base classes and follow that format.

The default services therefore include: remote messaging, file-sharing, email, web service invocation and search. There is also an Information service that can reference different types of resource, consider the IoT for example, but defaults as an image viewer or Web Page viewer, for example.

The rest of the document is organised as follows: Section 3 describes the default application services that can be used without any programming. Section 4 describes other services provided as additional modules, where you can follow that model to add your own. Section 5 then describes the test or scientific service classes that you can extend to write your own test programs.

2 Loading a Service onto the Server

Before you can use a service, you need to load it onto the server. This is easily done as follows:

1. After starting the All-in-One GUI, select the `Open Main GUI` button from the first menu.
2. From the Admin menu, select the `Load GUI Config` option. This also loads in your default services.
3. Change the server port and passwords if desired and then click `Start Server` to start the local server.
4. From the `Modules` column, click the `Service` button.
5. Select the top RHS Server graphic and click the licas server icon. This should add the name `HttpServer` to the top selected service box.
6. From the `All service types` combo box, select the service type, Message, for example.
7. Add a `Service ID` as the user name for the service.
8. Change the service passwords if desired.
9. Click `Load Service` to load it onto the licas server.
10. Clicking on the top button in the `Toolbar` column will refresh the graphic.
11. Right-click on the service icon to open its GUI.

Some of the service buttons are represented by icons, where the following coding has been applied.

⏩	Execute the operation on the service.
▶	Start a service loop.
⏹	Stop a service loop.
🏅	Switch the security certificate checks off.
💾	Save the current configuration.
⇥	Exit the application GUI.

3 Personal and Business Services

The personal and business services are complete solutions to some functionality and typically include their own GUI for interacting with. When you install the All-in-One GUI, a `services` folder is added to the `user/licasData` folder. A `licas_services` package and other jars are added to the services folder and they include the service suite provided as default. New jar files and service packages can then be loaded through the main GUI, but the following sections describe the service suite provided with the download.

3.1 Instant Messenger Service

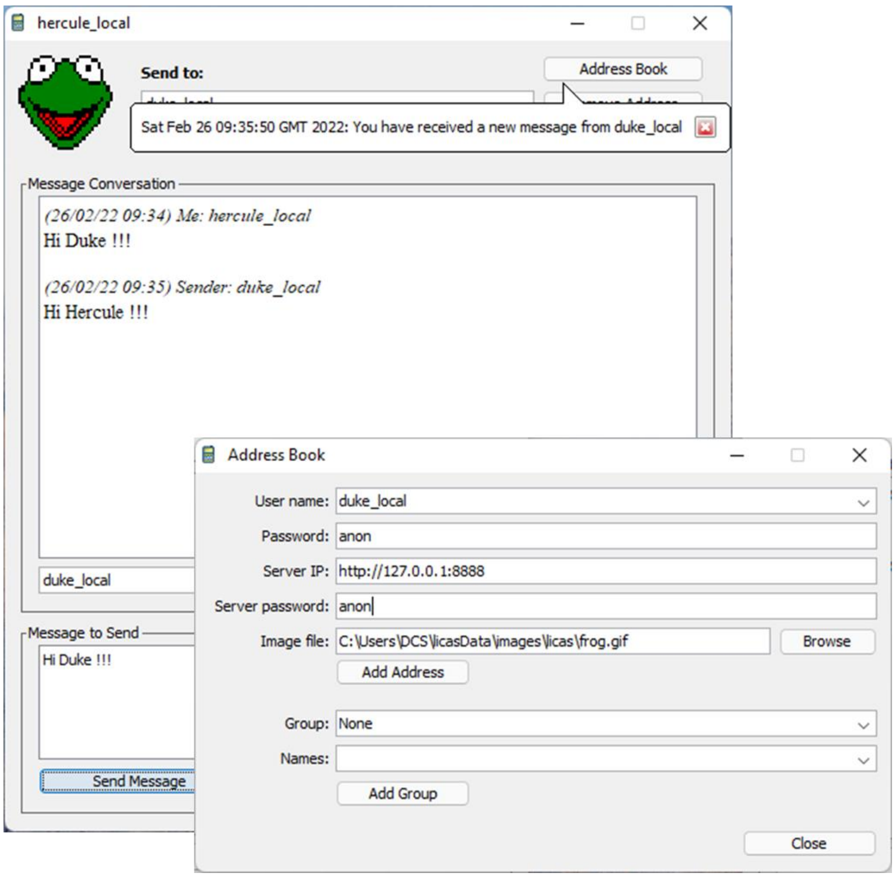
An instant messenger service can be used to send and receive text messages from another licas message service. To use this application, you firstly load a message service onto your server. You then use the popup menu from the service icon to open the related message GUI interface. Through the interface, you can ask the service to send a message to any other message service and receive the replies. The service can carry out multiple conversations and/or retrieve each message thread separately.

3.1.1GUI Service Interface

The interface provides a user-friendly view and allows the service to be configured, including storing addresses in an address book. An example of the GUI interface is shown in Figure 1. The top LHS of the GUI can display a picture of your client and also lists the address you are sending the message to. The top RHS of the GUI contains 3 buttons for the following features:

- **Address Book:** This opens the address book. An address can be added and saved to the address book. It can then also be selected as the client address by the service.
- **Remove Address:** This allows you to permanently remove the address from this GUI and does not require you to open the book again.
- **Block/Unblock:** While you can select what message thread to display, you also have the option of blocking any client for the duration of the current session, through a list of addresses using this button to open that choice. You have to `Update` the selection to make it permanent for the session only.

A full user description requires the HTTP address of the remote server and the name of the service (user) running on the remote server. Passwords for both the remote service and the server are required, but these can default to anon. The User name, displayed on the GUI interface, is also the ‘service’ name that you would call on the remote server. Each address can have an image associated with it, or a default one is displayed if none is available. These are also entered as part of the address details as an image file, where a size of 70 x 70 pixels would work best.



Sat Feb 26 09:35:50 GMT 2022: You have received a new message from duke_local

(26/02/22 09:34) Me: hercule_local
Hi Duke !!!

(26/02/22 09:35) Sender: duke_local
Hi Hercule !!!

duke_local

Hi Duke !!!

User name: duke_local
Password: anon
Server IP: http://127.0.0.1:8888
Server password: anon
Image file: C:\Users\DCS\licasData\images\licas\frog.gif
Group: None
Names:

Figure 1. Message Service GUI.

You then send a message by selecting an address, writing the message into the 'to send' text area and clicking the `Send Message` button. All replies are displayed in the top 'conversation' text area, which can be saved using the `Save Conversation` button. When you receive a message or reply, the first time will also output a balloon alert, indicating that something has been sent. There is then a time gap, from the same sender, before a balloon alert is output again. The bottom LHS of the GUI lists any clients who have sent messages during the session. You can therefore select a client address from there, instead of using your address book list.

3.1.2 Address Book

The address book stores details for each client address, where the addresses can also be added to a group. This means that you can send a message to all clients in the group, in one call event. You then enter new client addresses through the address book. The top user name combo box is editable, so you can enter a new client name there. You can also change the other fields before clicking the `Add Address` button, where:

- `User name`: this is the service name running on the remote server.
- `Password`: this is the password for that service (defaults to anon).
- `Server IP`: this is the full ip address of the remote server, http://127.0.0.1:8080, for example.
- `Server Password`: this is the password for the remote server (defaults to anon).
- `Image file`: the location of an image to display for the client address. You might want to save your images to the default `user/licasData/images` folder. The All-in-One GUI's Service Factory can maybe help with adding images.
- `Group`: select or add a new group for the client. If you enter a different name into the 'Group' combo box, you can click the `Add Group` button to save it. You can then send a message to a single client or a whole group. The `None` value means do not add to a group.

You will then be asked to confirm that you want to either add a new address, or update an existing one. To add addresses to the group, you need to click the 'Add Address', button after all fields have been entered. You also do this to update a change. If you save an address with a group name showing, then you are asked if you want to save it to the group as well. To remove addresses or groups, you need to use the `Remove Address` button in the main messenger GUI form.

3.1.3 Group Messages

Group names are listed in the main messenger form at the end of the individual addresses. There is a separator between them and the single addresses. If one is selected, it should list `Multiple entries` as the HTTP address. If you send a message to a group, it gets sent to each address in the group. If one of those addresses replies, then the conversation for that client only is displayed. However, you can simply click the `Retrieve Messages` button to retrieve the whole conversation thread for every client in the group combined. The thread that is retrieved depends only on the currently displayed address or group address, but for a client reply, it automatically retrieves that client only.

3.1.4 Program Code

This section describes a method call made behind the scenes. The method that is invoked to send a message is as follows:

```
sendMessage(String serverIP, String serverPassword, String commID, String
serviceUUID, String theMessage)
```

Where:

- `serverIP` is the ip address of the server the remote service runs on.
- `serverPassword` is the password for the remote server.
- `commID` is the communication id for the message thread.
- `serviceUUID` is the uuid of the remote service to invoke.
- `theMessage` is the message to send.

So, for example, you load a message service onto your server and give it a uuid of 'Hercule'. The person you want to talk to has a service on his server with the uuid of 'Duke'. The fields for calling this method could look like:

- The remote server ip address is http://192.168.1.1:8888/. This should be the same address that is in your address book.
- The server password is added automatically.
- The communication ID is generated automatically.
- The serviceUUID is Duke. This must have the correct ip address in the address book.
- The message is the message that you want to send.

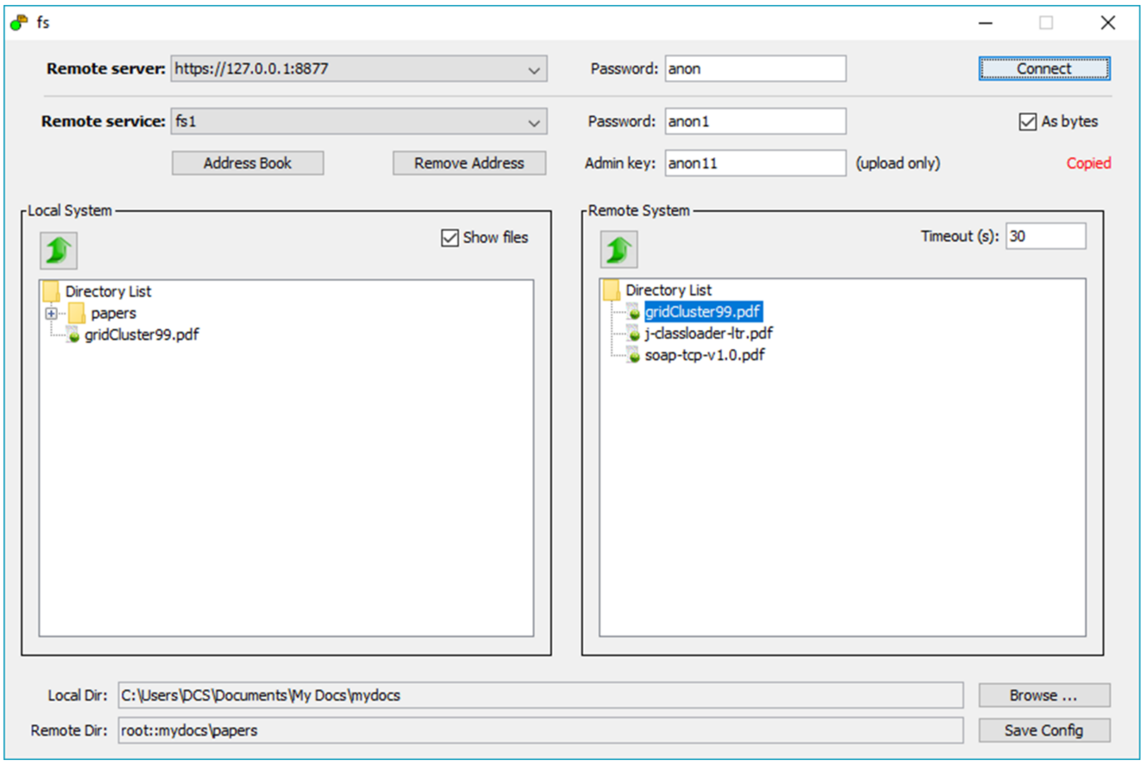
After you invoke this message, this message service stores the communication ID as a new conversation thread and also stores the message under that ID. Any replies with the same communication ID are stored in the same place. On being invoked, the Hercule service sends the message to the Duke service. The client of Duke can then retrieve the message and send a reply when appropriate.

3.2 File Sharing Service

There is also a p2p file sharing service. Using this service, you can send the contents of a file from one server to another. The service sends the file contents as either a `String` or an array of `Bytes`. 'As bytes' is the default option that should work with all file types, including the binary ones. Note that you can both download from a remote service and also upload to it.

3.2.1GUI Service Interface

The file service also has an interface, shown in Figure 2. You load a file service onto your server and then open the GUI interface, in the same way as for the message service. The service then communicates with a similar remote file service, in a p2p manner. The GUI allows you to store a local configuration, which can restrict folder access and also the addresses of remote services that you want to connect with. You would firstly make sure of the connection by clicking on the `Connect` button. When you do this, the button text changes to `Change` and some of the combo boxes are disabled. This is to remind you not to change your remote settings by accident. If you want to change to a different remote service, you need to click the Change button first and then the combo boxes are enabled again.



Remote server: https://127.0.0.1:8877
Password: anon
Remote service: fs1
Password: anon1
☑ As bytes
Admin key: anon11 (upload only)
☑ Show files
Timeout (s): 30

Directory List
papers
gridCluster99.pdf

Directory List
gridCluster99.pdf
j-classloader-ltr.pdf
soap-tcp-v1.0.pdf

Local Dir: C:\Users\DCS\Documents\My Docs\mydocs
Remote Dir: root::mydocs\papers

Figure 2. File Service GUI.

There is a GUI-specific configuration that is saved using the `Save Config` button on the bottom right of the form. This is for the local file service that is running and not the remote one and includes basic information such as transport format (as bytes) and timeout. For your local service, you can `Browse` to set a local directory ‘root folder’. Only folders and files underneath that one will then be accessible to any other service and therefore provides some protection to other parts of the file system. There is also an address book to store addresses of the remote file services you connect with. An administrator might setup more than one file service on the same server and so the address book allows for more than one service uuid for each server ip address. The same service uuid can also be run on different servers and so the address book needs to store full addresses, as for the message service.

You can also store the default root folder for a particular service name. If you then start a service running with one of the specified names, the config file is checked and if it is saved, the local root folder is retrieved and setup as part of the service initialisation. So this can be remembered and automatically setup again through the service uuid name only. In Figure 2, for example, the local service has the name `fs` and the local root directory that has been browsed to is 'C:\Users\DCS\Documents\My Docs\mydocs'. If you click the `Save Config` button, this folder will be associated with this name. If you ever create a file service called 'fs' again, the config file is read and the root folder is automatically set to this value. Note that using the `Browse` button and changing the local system directory both update the local directory.

3.2.2 Copying a File

To copy a file, you enter the details of the remote file service and server that you want to invoke. You need the passwords for both the remote server and the file service, as both might be password protected. You then click the `Connect` button to retrieve the remote directory root folder details. Clicking on a folder will move to that folder and the new path should be displayed at the bottom of the GUI. To copy a file: drag it from the remote directory tree to anywhere in the local directory tree area. It should then be copied to the specified local folder. There are also the following features:

- The `Address Book` button opens the address book.
- The `Remove Address` button permanently removes an address.
- The large green arrow button at the top moves the path back to the parent of the current path, but not beyond the specified root folders.
- The `Show files` check box should be checked to show local directory files.
- The process of transferring larger files can take some time and so the `Timeout` box, in the remote system area, can be used to specify a longer maximum allowed amount of time for the message transaction. The default is 10 seconds, but you can enter a longer time period (in seconds) here. The call should automatically time out after the allowed time period. Using smaller packet sizes therefore also helps here, as the time is then for each packet transaction and not the whole file transaction.
- The `As bytes` check box can be used to specify that all file contents should be transferred as a list of bytes. This is currently very inefficient, but potentially more safe. The default for file transferring is currently set to false.

The communication is asynchronous and so the file is loaded sometime later and not immediately. There is therefore a status message that indicates when the file is being copied and when it has been fully downloaded. The check is that the file now exists on the local server.

3.2.3 Uploading to a File Service

Any file service can act both as the server to retrieve from or the client to download other files to. If you want to upload to the remote file service, you also need to know the service's admin key. This is because uploading is an intrusion on the remote system and so you need administrator privileges to do it. For a completely open system, the passwords can always be set to `anon`, for example. To upload then, you simply select the file from the local system area and drag it to the remote system area.

3.2.4 Program Code

When programming, the local file service variables are initialised with the details of the remote service to invoke, for the subsequent communication. This is done by invoking the method:

```
setServiceDetails(String theServiceUuid, String servicePassword, String serviceKey,
Element theServerUrl, String serverPassword)
```

After that, the local service knows the remote service address, passwords, etc. To retrieve and save a file, the main method to invoke is then the following:

```
getAndSaveFile(String filePath, String savePath, boolean asBytes, int
packetSize, int callTimeout)
```

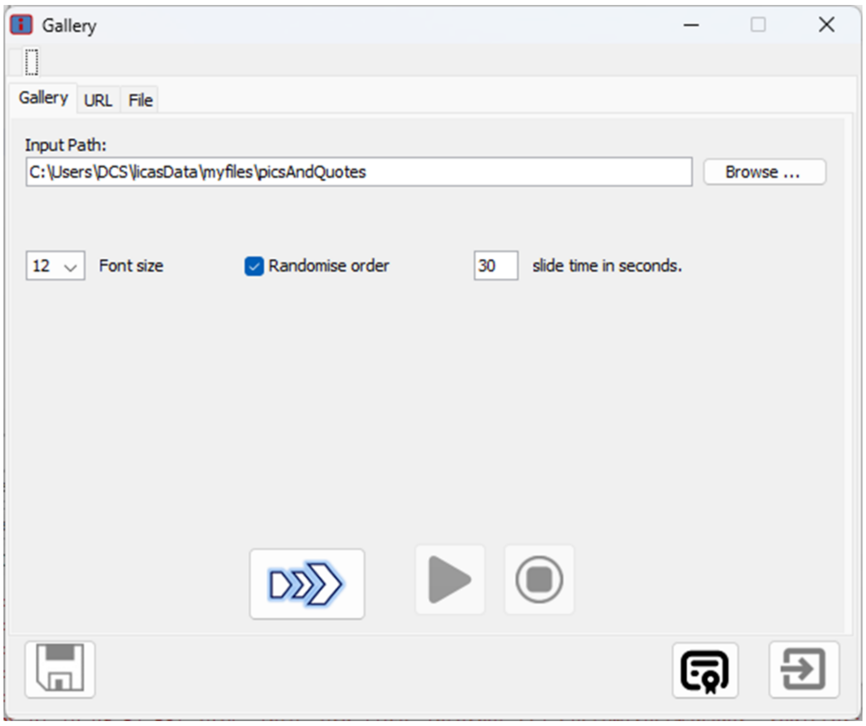
The user can enter these values through the GUI interface and they are passed to the service for it to use as part of the method call. The method functionality asks a remote service for the contents of a file and then saves that file locally, where:

- `filePath` is the path to the file on the remote server that you want to retrieve. The remote service will add its root directory to the path before trying to retrieve the file.
- `savePath` is the path to the file on the local server that you want to save the contents to. This is the full path specification.
- `asBytes` indicates if the file contents should be transferred as a list of bytes, no matter what the file type is. The contents can also be stored in string or XML-based formats, but it might be simpler to transfer all files as a list of bytes. The FileObject has a setting to change this. A file type-specific format is the default, so for bytes in all cases, the boolean value must be set.
- `packetSize` indicates the maximum size of a reply message when transferring the file contents. A large file can be split up into a number of message calls that are saved to an XML file on the client side and then read back into the FileObject when all of the message parts have been received.

- `callTimeout` indicates the maximum allowed amount to time for a message transaction. The default is 10 seconds, but this might be too short for transferring larger files. The file service GUI allows you to specify this is seconds, but the method interface requires the value in milliseconds (thousands of a second). So 10 seconds would be converted into 10000 milliseconds. The main All-in-One GUI also requires milliseconds.

3.3 Information Service

An information service can be used to connect to an information source, read its contents and display the result. The GUI provided with the Information service turns it into an app for flexible loading and configuration of the file addresses. It can be used to read online web pages, or as a text or image Gallery for local files. There is a separate `Feed Service` for reading RSS or Atom feeds, as described in section 3.4. The information service can reference different types of file and can display a static piece of information, or view the contents of a URL reference. The GUI consists of a `View` panel and a `Config` panel. The config panel is separated into tabs for the different types of information. An information service stores its information in resource objects (see the 'licasUserGuide'), where it can store text, XML, binary (image) files, or URL addresses to remote locations. The details that you enter can be saved at any stage by clicking on the `Save` button in the Config panel. The save option also stores some of the GUI component settings, such as time delays or cycles.



Input Path: C:\Users\DCS\licasData\myfiles\picsAndQuotes

12 Font size

✓ Randomise order

30 slide time in seconds.

Figure 3. Config panel for the Gallery Viewer.

3.3.1 GUI Slide Show

One option for the Information Service is as a slide show, where the configuration for this is shown in Figure 3. You can browse to a folder from the top `Input Path` box, by selecting the `Browse` button. You then click the service button to transfer the folder contents to the service. They are transferred as URL addresses, not as the file content itself and this then enables the start and stop buttons for viewing. You can set a time in seconds and a text size and `Start` the slide show running. The `Randomise` check box will return the slides in a random order, where both text and images can be included. The `Stop` button will then stop the slide show and Figure 4 shows what the viewer itself looks like.

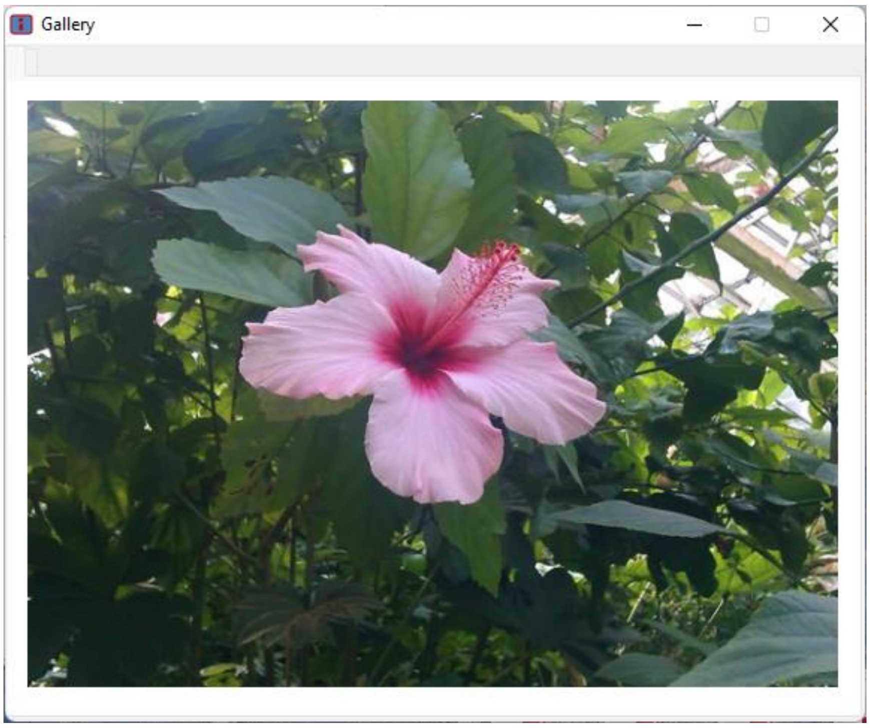
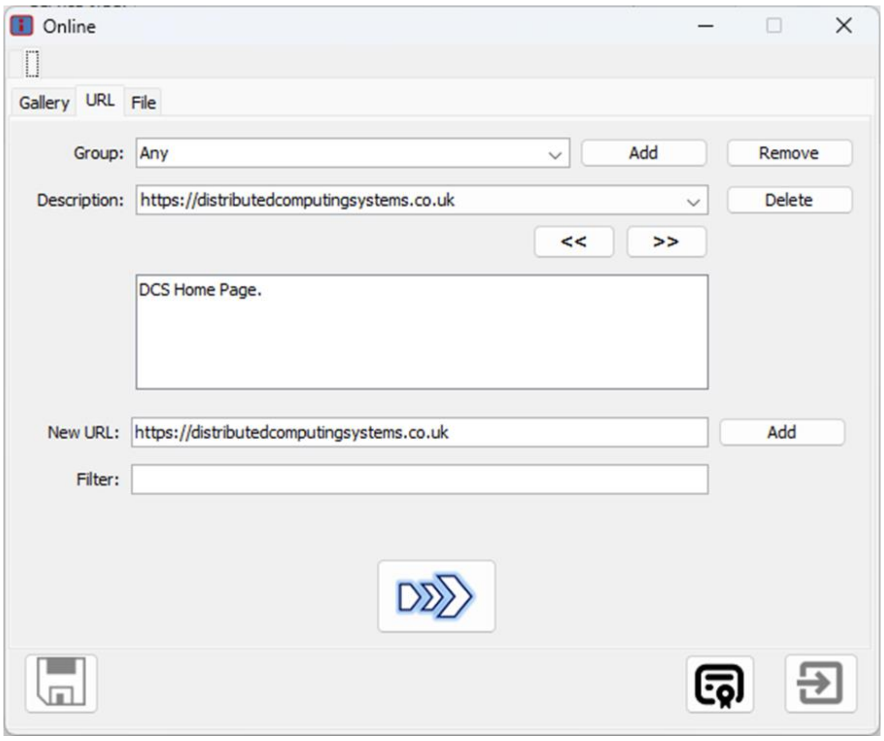


Figure 4. Information Service GUI as a Slide Show

3.3.2 URL Addresses

A second config panel allows you to load `URL addresses` into the service. This would be useful, not only for html pages, but also an online search with query results. The configuration panel for this is shown in Figure 5. The URL addresses are assigned to a `Category Group`. There is a default `Any` group and then you can add new ones, through the top combo box and the `Add` and `Remove` buttons beside it. You then enter the URI address into the `New URL` text field and click the `Add` button. You can then select the address from the list and enter a description for it. You then have to `Save` the new configuration to make it permanent. When you select a group, only the addresses listed for the group are displayed. Then, you can select any address and use the service button to display its contents in the viewer. There is also a `Keyword filter` search box at the bottom of the panel. If you type some text in there, then only URIs with an address or description that contains the filter term are displayed. The keywords can be comma-separated when a match with each separate term is made. You can therefore match with one word sequence, add a comma, and then filter that result further with another word sequence, and so on.



Group: Any

Description: https://distributedcomputingsystems.co.uk

DCS Home Page.

New URL: https://distributedcomputingsystems.co.uk

Filter:

Figure 5. Load URL config panel.

3.3.3Security Certificates

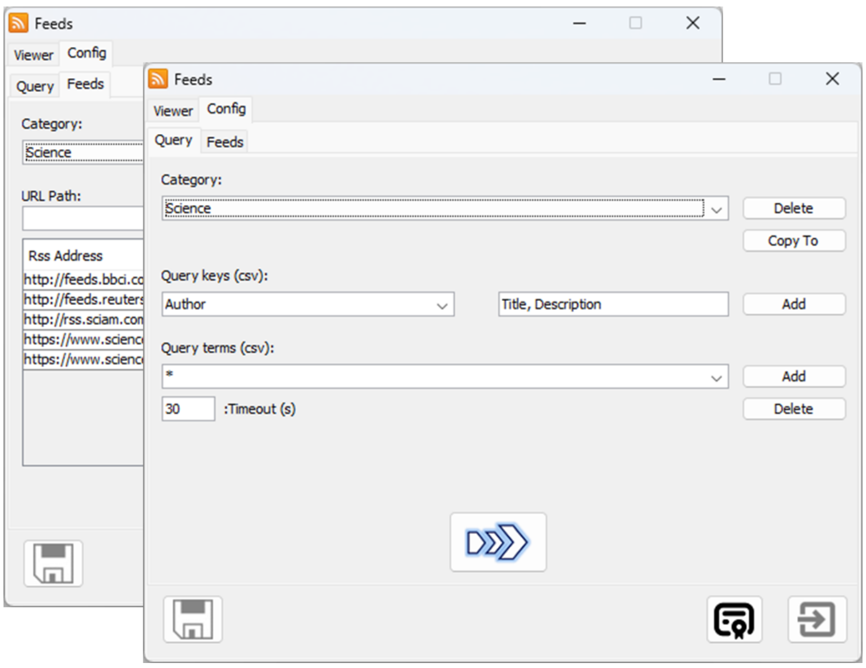
Licas has its own security certificate, but it is not a fully-robust server and so with some other online sites, the certificate matching might not work. Because the interaction with those sites would be as a client only, it is possible to disable the security certificate checks by clicking the certificate button on the bottom toolbar. If the certificate check is disabled however, it cannot be switched back on for the current session. It simply means that the certificates are not checked. This would leave your online activity more vulnerable, so it should only be done for sites that you trust. If you receive feedback about errors trying to connect to online sites however, then disabling the check can help.

3.3.4 File Paths

If you wish, you can also load local file contents into the service. There is again a description field for saving some information about the file link. The setup is the same as for the URL addresses, except that there is no category grouping of the addresses.

3.4 Online Feeds Service

An information service can be used to connect to an information source, read its contents and display the result. One type of online information are RSS or Atom feeds. A Feeds service can be used to connect to those sources specifically, query them and view the result. The process is the same as using an information service and the example is shown in Figure 6 and Figure 7. The setup is slightly more complicated, as you need to enter the feed addresses and a query clause to filter the results. There is a default '*' query term, that will return all of the feed. The feed addresses may need to be checked and should return the feed in RSS or Atom format.



Category: Science

URL Path:

Rss Address
http://feeds.bbci.co
http://feeds.reuters
http://rss.sciam.con
https://www.scienc
https://www.scienc

Category: Science

Query keys (csv): Author | Title, Description

Query terms (csv): *

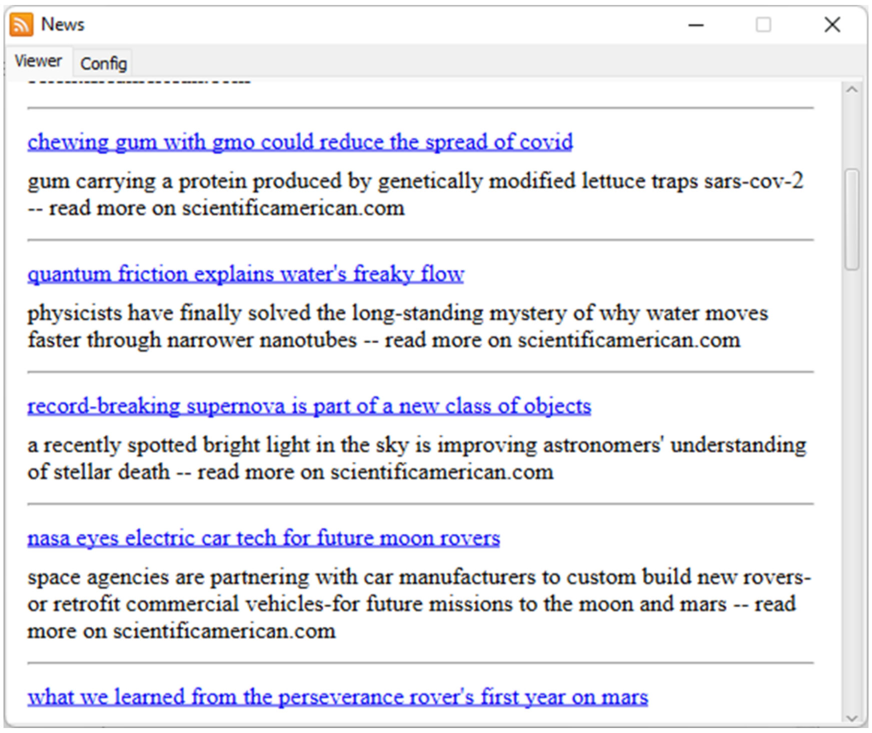
30 :Timeout (s)

Figure 6. Config panel for the Feeds Viewer.

3.4.1 Adding New Feeds

You add a list of feed addresses to a particular category. You can then select a set of query keywords to look for in the different fields of each returned feed. From the `Config –`

Feeds panel, you firstly add a new category. Enter the category into the top right text box and click the top right Add button, to add it to the top left list. To add an address, paste it into the URL Path text box and click the Add button beside that. You can then select or de-select addresses using the check box in the address list, or you can delete or copy them. If you then go to the Query panel, you can query one or more of the returned feeds.



chewing gum with gmo could reduce the spread of covid

gum carrying a protein produced by genetically modified lettuce traps sars-cov-2 -- read more on scientificamerican.com

quantum friction explains water's freaky flow

physicists have finally solved the long-standing mystery of why water moves faster through narrower nanotubes -- read more on scientificamerican.com

record-breaking supernova is part of a new class of objects

a recently spotted bright light in the sky is improving astronomers' understanding of stellar death -- read more on scientificamerican.com

nasa eyes electric car tech for future moon rovers

space agencies are partnering with car manufacturers to custom build new rovers-or retrofit commercial vehicles-for future missions to the moon and mars -- read more on scientificamerican.com

what we learned from the perseverance rover's first year on mars

Figure 7. Service GUI as a RSS/Atom feed.

3.4.2 Query a Feed Category

From the Query panel, select a feed category at the top. There is a Copy To button that allows you to clone the whole category, which may be useful. When you run a query, you try to match with words in one of the feed fields. You enter a list of keywords through the Query terms text box and compare that with the Query keys fields list. The options there are title, description, or author. Both are stored in csv format and you should use the related Add button to add to add each option. You can manually enter any number of query terms that must also be separated by commas. If you then Save the config, these settings should get saved and re-loaded when the GUI is opened again. The config will also save the current query category for the service UUID. This means that when you load the same service again, it will automatically select the query category you are currently using for it.

When you transfer this config to the service, it will look-up the listed feeds and retrieve any current messages. It will then filter those replies on the query conditions and list only the ones that match. If you have already run a query for a particular category, its feed reply is saved. The query is then a further filter on that, but the filter can be done on the result already returned, which should save time. So, if you then execute a different query on the same category, you are asked if you want to retrieve the feeds again from the sites, or execute it on the stored feed replies.

3.4.3Context Menu

Right-clicking on the RSS viewer activates a popup menu that allows you to navigate backwards or forwards through the browse history. It also allows you to copy the URL, to paste it into a different application.

3.5 Email Service

The email service can be used either as a first-class service or as a utility service. As a first-class service, you can monitor a number of email accounts, to be notified when new emails arrive. As a utility service, you can use it programmatically to send email messages when some event has occurred. You add a utility email service through the main GUI service popup menu. You can also manually send a message through the GUI interface, but a standard email client would probably be more useful. There is typically a 'password' and sometimes additional 'username' for email accounts, as well as some port and address settings and need to be saved to the application first. Figure 8 shows the GUI interface for monitoring email accounts.

The program does not remove emails from your account, but asks the server for details about any new emails that have arrived. You start the process by clicking on the `Start` button. The program will then access each email account every specified number of minutes and look to see if any new emails have been posted. Basic header information about these emails and the first few lines of content is then displayed. Each time you change the settings, you need to `Stop` the current loop and re-start it again.

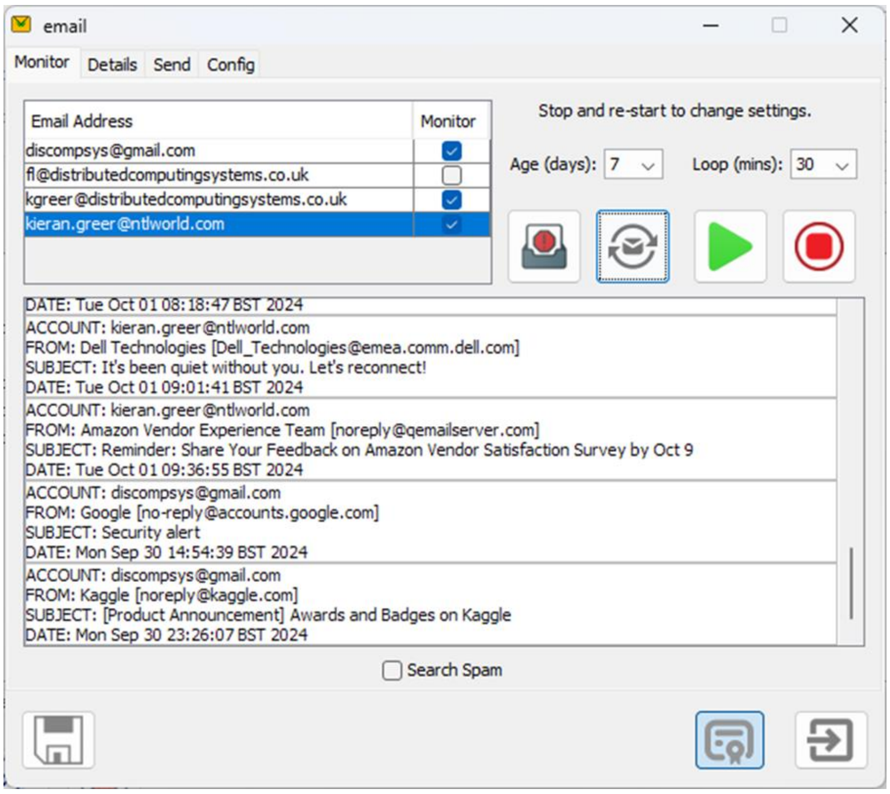
3.5.1Monitor Email Configuration

You can setup an email address to use, using the config panel, shown in Figure 9. All of your email accounts are then displayed in the monitor panel and you can select the ones to monitor through the check box. If you click on an email, some details at the first few lines are shown in the `Display` panel. The monitoring can be configured as follows:

- **Days old time:** This is the maximum age of the email to return and is defined in days. It can have a value from 1 to 7. In the figure, emails 2 days old or less are returned.
- **Loop time interval:** This is the time to sleep between monitor invocations. It is defined in minutes, from 1 minute to 60 minutes.

Then there are a set of 4 buttons as follows:

- **Spam settings:** This opens the spam filter form, described in section 3.5.7. To search your spam folder, you also need to click the `Search Spam` check box on the main form, otherwise the keyword search is not performed.
- **Refresh View:** This will refresh the current email list view.
- **Start monitoring:** this starts the monitoring.
- **Stop monitoring:** this stops the monitoring.



Email Address	Monitor
discompsys@gmail.com	✓
fl@distributedcomputingsystems.co.uk	☐
kgreer@distributedcomputingsystems.co.uk	✓
kieran.greer@ntlworld.com	✓

Stop and re-start to change settings.

Age (days): 7 Loop (mins): 30

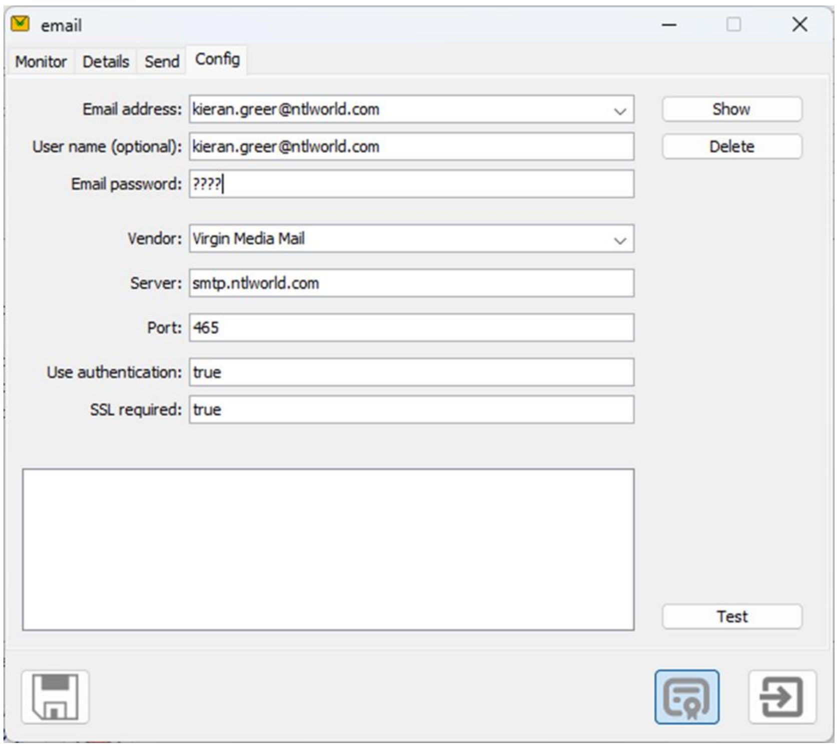
```
DATE: Tue Oct 01 08:18:47 BST 2024
ACCOUNT: kieran.greer@ntlworld.com
FROM: Dell Technologies [Dell_Technologies@emea.comm.dell.com]
SUBJECT: It's been quiet without you. Let's reconnect!
DATE: Tue Oct 01 09:01:41 BST 2024
ACCOUNT: kieran.greer@ntlworld.com
FROM: Amazon Vendor Experience Team [noreply@qemailserver.com]
SUBJECT: Reminder: Share Your Feedback on Amazon Vendor Satisfaction Survey by Oct 9
DATE: Tue Oct 01 09:36:55 BST 2024
ACCOUNT: discompsys@gmail.com
FROM: Google [no-reply@accounts.google.com]
SUBJECT: Security alert
DATE: Mon Sep 30 14:54:39 BST 2024
ACCOUNT: discompsys@gmail.com
FROM: Kaggle [noreply@kaggle.com]
SUBJECT: [Product Announcement] Awards and Badges on Kaggle
DATE: Mon Sep 30 23:26:07 BST 2024
```

☐ Search Spam

Figure 8. Email Service GUI Monitor Panel.

3.5.2 Email Account Settings

Most email accounts are password protected, and run using specific ports and addresses, and so you need to enter this information for your own particular accounts. Some of the well-known email providers have default information that you can find, or you may need to ask your administrator what the correct settings are. You can then enter and save the information through the config panel of Figure 9. All of the fields are editable, so if something does not match, you can change it manually. The `Show` button can reaffirm that the currently selected email settings are being shown. After you enter new details, click the `Save` button to save them and the `Delete` button will delete them. The save option also stores some of the GUI component settings, such as time delay or email age. So, if you set something in the `Monitor` panel, you can also save it here. At the bottom of the panel there is a Test option that will try to send a test email to the specified account and also try to retrieve some information. The test sends an email from the account to itself and a trace of the result is output. The send test may fail, when the message will not be output, but you can still retrieve information about your emails, even if this fails.



Email address: kieran.greer@ntlworld.com
User name (optional): kieran.greer@ntlworld.com
Email password: ????
Vendor: Virgin Media Mail
Server: smtp.ntlworld.com
Port: 465
Use authentication: true
SSL required: true

Figure 9. Email Service GUI Config Panel.

3.5.3 Gmail and Authentication

There are new security checks now and it is more difficult for a third-party app to be able to access your email accounts. With Gmail, for example, you need to give special permissions in your account. The first thing you need to do is enable 2-Factor Authentication. This can be done in the security tab. After that, search for ‘App Passwords’ in you account search box and you will be directed to a page where you can create an app password. Enter a name, e.g. licas, and a 16 character password is displayed. Copy the password and replace your email account password in Figure 9 with the app password – spaces removed. Save this and then run the service, and the app password should allow access to your account.

3.5.4 Popup Menu

After you receive a list of emails, you can select an item on the list and right-click to open a popup menu with a `Delete` option. After confirmation, you can delete the email from your email server account. This also adds the email domain address to a temporary list. If you perform the action again on the same domain, you are asked if you want to move the address to a permanent delete list. You can then open a Spam filter form and ask for all emails from the specified domains to be deleted from all of your accounts. Alternatively, you can add the specific address to a safe list. In that case, you can delete any number of emails and you are not asked about spam for the address again.

The popup menu includes two other options, where only 1 is shown depending on the address state. The first option is Add to safe list. If selected, then the current sender is added to a safe addresses list. You can then delete any number of emails from that sender and you will not be asked if you want to spam the address or domain. The second option allows you to remove the address from the safe sender’s list, so that you will be asked if you want to spam the email again.

3.5.5 Description Panel

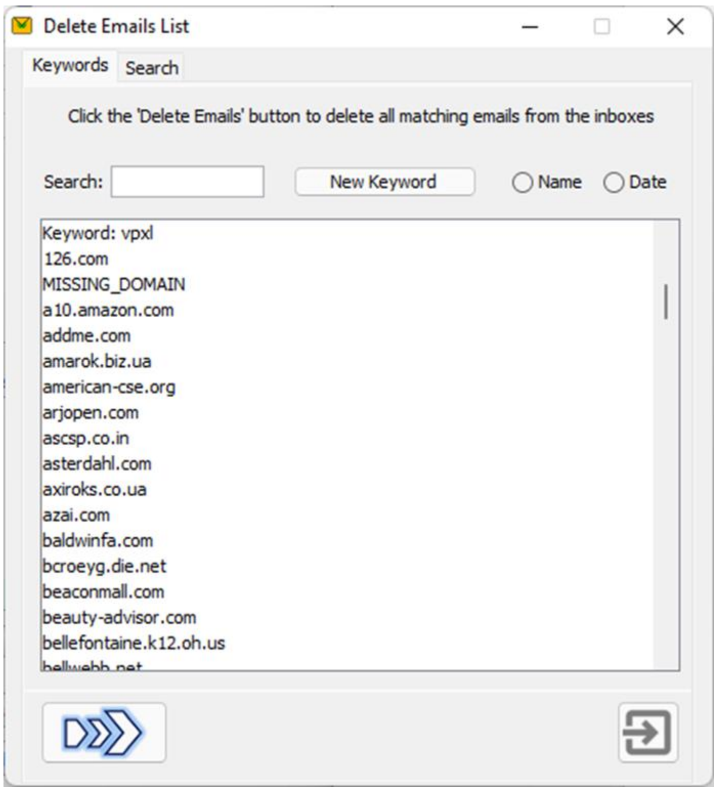
If you click on an email in the monitor list, this panel gives a slightly more detailed description of the sender details. It does not retrieve and display the whole email, but it gives more details about the sender and receiver addresses, the content type and the first few lines of the message only.

3.5.6 Send Message Panel

It is possible to send an email using a selected account through the `Send` panel, although, using a standard email client would probably be recommended. To send an email, you need to firstly need to enter the details for the client you are sending the email to. You then load the account settings onto the service using the `Service` button and then you can `Send` the email.

3.5.7Spam Filter

When you open the Spam Filter form, shown in Figure 10, it gives two options for managing the spam emails. The first `Keywords` panel, lists the email domains that you have selected as spam domains and it also lists keywords that you have selected for removing emails with. You can add keywords by clicking the `New Keyword` button and entering the information. Domains are added when you delete emails from the inbox list. The spam list also includes a popup menu allowing you to remove an entry again. There is also a filter box to filter through the list. The second panel allows you to enter keywords again, but this time to be used as search terms. You click on the `Query Term` button and similarly add keywords to a list. Then when you monitor your emails, the spam folder will be checked and any emails in that folder that contain any of the search terms will be listed in the monitor list. They are identified by the fact the they start with SPAM ACCOUNT and not just ACCOUNT.



Click the 'Delete Emails' button to delete all matching emails from the inboxes

Search:

Keyword: vpxl
126.com
MISSING_DOMAIN
a10.amazon.com
addme.com
amarok.biz.ua
american-cse.org
arjopen.com
ascsp.co.in
asterdahl.com
axiroks.co.ua
azai.com
baldwinfa.com
bcroeyg.die.net
beaconmall.com
beauty-advisor.com
bellefontaine.k12.oh.us

Figure 10. Spam Email Filter.

3.5.8 Additional Cases

- Some vendors have higher levels of protection or security and this is constantly being updated, when the email application may not work properly. If using Google to send your email, for example, you would need to log into your account and allow 'access for less secure apps' in your settings. If you try to send an email and it is blocked, then you should get a message in your account giving you this option.
- If the `Server` address of the `Config` panel contains `replace_with_address`, as with 'VirginMedia', then that part is replaced by your personal email address part, after the `@` symbol. It might be the case that one provider uses more than one address, so this is just a general description that then gets updated. So, `smtp.replace_with_address`, might be changed to `smtp.ntlworld.com`, if the user address is 'joe.bloggs@ntlworld.com'.

Some of the more popular email providers are listed. If one is missing and you send details, then I will add it to the default list.

3.5.9 Program Code

To send an email through the service code, you create an `EmailInfo` object to describe the message and send this on the `EmailService`. Using my own two email accounts, this was as easy as:

```
EmailInfo emailInfo = new EmailInfo();
emailInfo.recipient.addElement("my email account 1");
emailInfo.username = "my username";
emailInfo.password = "my password";
emailInfo.sender = "my email account 2";
emailInfo.senderHost = "smtp.sender_host_address";
emailInfo.message = "test message";
emailInfo.subject = "test email";
emailInfo.attachmentFile = "Can add an attachment file path";

EmailService emailService = new EmailService();
emailService.setEmailInfo(emailInfo);
emailService.sendEmail(sendTo, subject, message, attachmentFile);
```

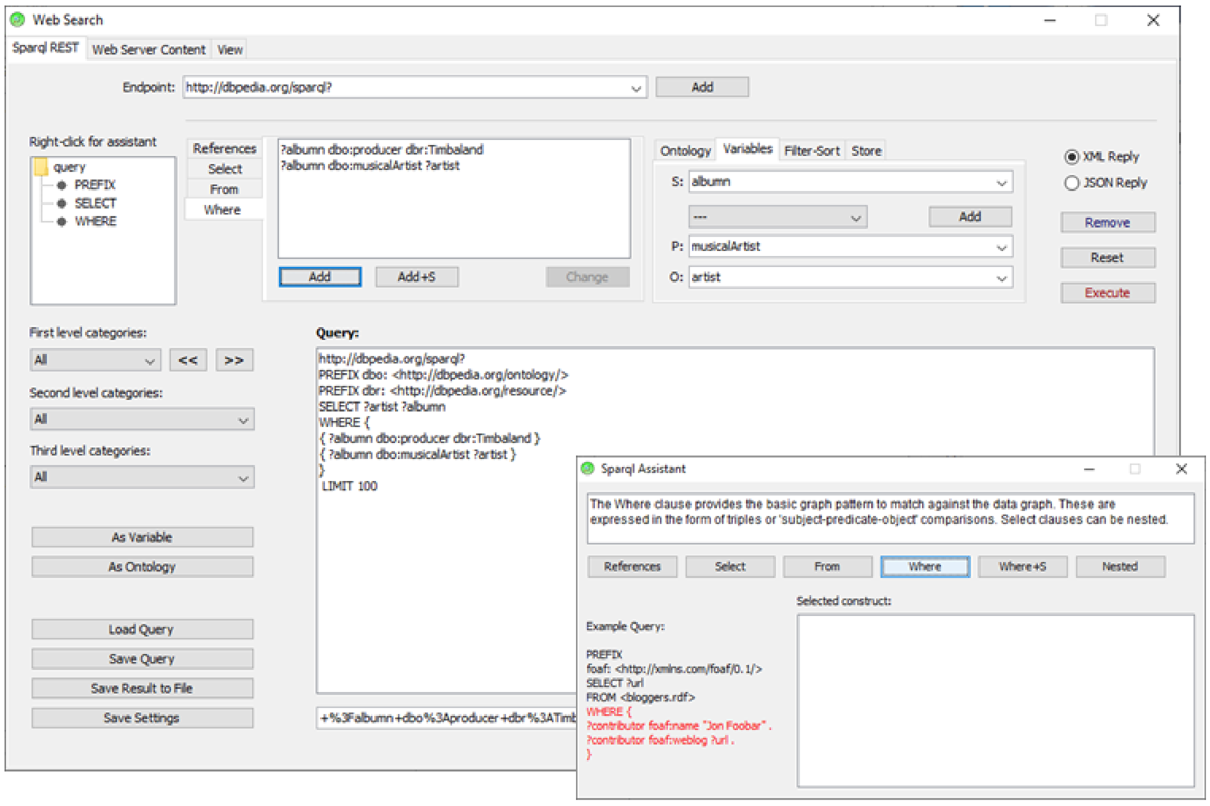
There is also a 'username' and 'password' for some accounts, where the GUI will help to show what values are required. Note that you can add the subject or message through the setup `setEmailInfo` call, or through the direct `sendEmail` call.

3.6 ‘Web Search’ Service

This service can be used to search over different types of online content. The service consists of a number of panels. One is for the SPARQL query language, to query RDF sources over the Internet. The other allows you to query licas servers and ask about the contents of the web folders. With SPARQL, you can construct a complex database query and there is some help. With licas, you can query for basic matches to file names or file contents. A third option is to match with online text. If you have performed an RDF query first, then that might have returned some source links. You then have the option of matching with the text from one of those online sources. If the RDF query is considered to be a metadata description, then this second basic matching query can be performed on the sources, to drill down deeper into the source content.

3.6.1SPARQL Queries

The query services GUI is shown in Figure 11. This shows the SPARQL query panel that has constructed a basic ‘select with conditions’ query. The REST interface is then used execute the query. As SPARQL is a complicated language, there is some help in the form of an assistant panel that can be opened by right-clicking on the query tree on the far left. The help form can display further descriptions and open the appropriate tabs for each type of construct.



Endpoint: http://dbpedia.org/sparql?

```
?albumn dbo:producer dbr:Timbaland
?albumn dbo:musicalArtist ?artist
```

S: albumn

P: musicalArtist
O: artist

[✓ XML Reply] JSON Reply

First level categories: All
Second level categories: All
Third level categories: All

Query:

```
http://dbpedia.org/sparql?
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?artist ?albumn
WHERE {
{ ?albumn dbo:producer dbr:Timbaland }
{ ?albumn dbo:musicalArtist ?artist }
}
 LIMIT 100
```

```
+%3Falbumn+dbo%3Aproducer+dbr%3ATimb
```

The Where clause provides the basic graph pattern to match against the data graph. These are expressed in the form of triples or 'subject-predicate-object' comparisons. Select clauses can be nested.

Example Query:

```
PREFIX
foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url
FROM <bloggers.rdf>
WHERE {
?contributor foaf:name "Jon Foobar" .
?contributor foaf:weblog ?url .
}
```

Figure 11. The Web Search service, with the SPARQL query panel.

After constructing a query, it can be saved and re-loaded again. However, it can then only be executed again and not updated or changed. SPARQL is a very verbose query language and only a subset of the features is currently available. It would require a larger document to describe everything and so it might be best to try to create some queries to see how the service generates them. Note that you can enter alternative RDF databases, ontology addresses with names, key tags or names and other config and syntax that you might repeatedly use. The reply can be returned in either XML or JSON and you can then scroll through the results.

3.6.1.1 VARIABLES AND VALUES

The RHS tabbed pane allows you to enter values and also to select the values to add to your query. Remember to Save Settings after adding new values, if you want to save them permanently.

Ontologies, Prefixes and Resources

The Ontologies panel allows you to add new ontology addresses, with a related prefix, to your query. This is added at the top and relates to the ontology list at the start of a query. Resources are then value-specific addresses that you use in your query. There is therefore a separate set of GUI boxes below the ontology list, so that you add them separately. To add a new ontology with its prefix, you need to use the `As Ontology` button and type the URI address into the text area. The `As Variable` button does the same thing for adding new variables, although, these can also be added in the RHS selection panels. The update an ontology address for an existing prefix, you can change it in the RHS panels instead.

Variables

The variables panel allows you to select values for the RDF tuple. In some cases, you only need the first or `S` box (Select clause) and in some cases, you need to enter values for the whole tuple (Where clause). You can also enter a text value into a field, but make sure that you then click somewhere else to set it before it will be added. With the Where clause, a second form then opens when you go to add it, so that you can add other information, as required.

Filter-Sort

Other query-wide filter or sort statements can be added from this panel.

Store

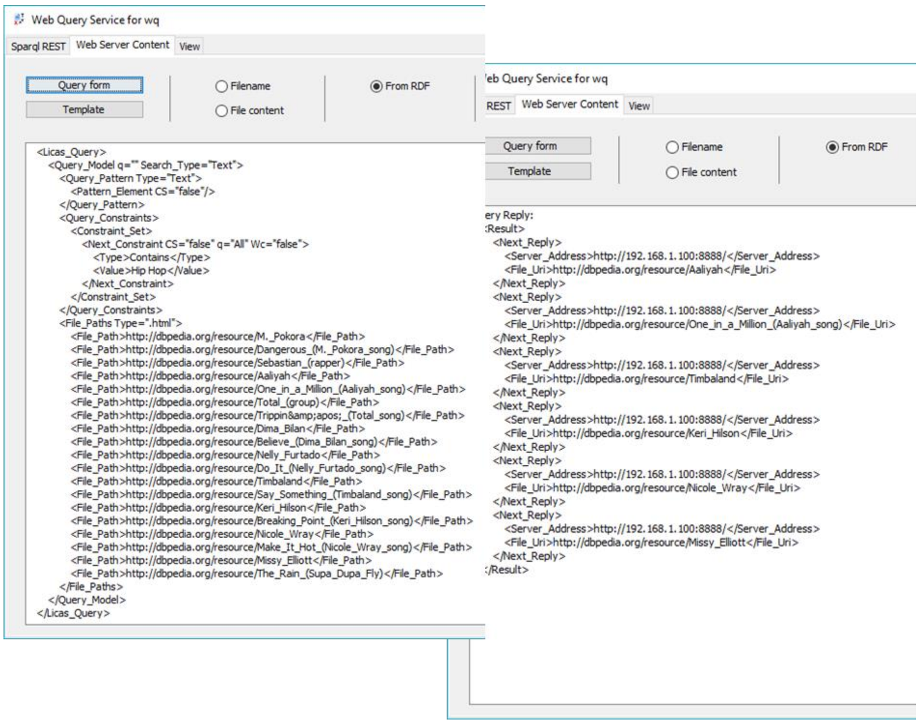
If you construct a Select query, you can `Store` it using this panel. The main query is then reset and you can construct a new main query. As part of the `Where` clause, you can then `Retrieve` any of the saved Select clauses.

3.6.1.2 CONSTRUCTING A QUERY

The middle-tabbed pane then contains the different types of clause. There are Select-From or Where, plus the query-wide ontology list. Services can be added as part of a Where clause and they use the `Resources` Address as their value. To add a new value to the query, you select the appropriate middle panel tab and any related values from the right-hand set of tabs and then click the Add button. The `Change` option may allow you to change the contents of a currently selected clause, instead of re-writing the whole query.

3.6.2 Web and Server Queries

A second query panel allows you to use the licas query form to make specific types of comparison on the text contents or files names. This can be done on the licas web folder or it can be done on the source list returned by the RDF query, for example. It is still a work in progress, but Figure 12 gives one example. In this example, the `From RDF` option is selected first. The `Query Form` button is then used to open a query form where a basic `Contains` query is entered. As this is a text query, the XML-based elements are empty, the pattern element, for example. Case sensitive and Wildcards are also set to default. Note the addition of a list of file sources to the query description and so when the query is executed, it is carried out over those sources only. The query asks for web pages that contain the words ‘hip hop’ and the following sources are returned, shown in the RHS figure. A third panel is a convenience `Viewer` for some of the stored resources.



( ) Filename ( ) File content [✓ From RDF]

```
<Licas_Query>
  <Query_Model q="" Search_Type="Text">
    <Query_Pattern Type="Text">
      <Pattern_Element CS="false"/>
    </Query_Pattern>
    <Query_Constraints>
      <Constraint_Set>
        <Next_Constraint CS="false" q="All" Wc="false">
          <Type>Contains</Type>
          <Value>Hip Hop</Value>
        </Next_Constraint>
      </Constraint_Set>
    </Query_Constraints>
    <File_Paths Type=".html">
      <File_Path>http://dbpedia.org/resource/M._Pokora</File_Path>
      <File_Path>http://dbpedia.org/resource/Dangerous_(M._Pokora_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Sebastian_(rapper)</File_Path>
      <File_Path>http://dbpedia.org/resource/Aaliyah</File_Path>
      <File_Path>http://dbpedia.org/resource/One_in_a_Million_(Aaliyah_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Total_(group)</File_Path>
      <File_Path>http://dbpedia.org/resource/Trippin&amp;apos;_(Total_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Dima_Bilan</File_Path>
      <File_Path>http://dbpedia.org/resource/Believe_(Dima_Bilan_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Nelly_Furtado</File_Path>
      <File_Path>http://dbpedia.org/resource/Do_It_(Nelly_Furtado_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Timbaland</File_Path>
      <File_Path>http://dbpedia.org/resource/Say_Something_(Timbaland_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Keri_Hilson</File_Path>
      <File_Path>http://dbpedia.org/resource/Breaking_Point_(Keri_Hilson_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Nicole_Wray</File_Path>
      <File_Path>http://dbpedia.org/resource/Make_It_Hot_(Nicole_Wray_song)</File_Path>
      <File_Path>http://dbpedia.org/resource/Missy_Elliott</File_Path>
      <File_Path>http://dbpedia.org/resource/The_Rain_(Supa_Dupa_Fly)</File_Path>
    </File_Paths>
  </Query_Model>
</Licas_Query>
```

```
ery Reply:
:Result>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/Aaliyah</File_Uri>
  </Next_Reply>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/One_in_a_Million_(Aaliyah_song)</File_Uri>
  </Next_Reply>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/Timbaland</File_Uri>
  </Next_Reply>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/Keri_Hilson</File_Uri>
  </Next_Reply>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/Nicole_Wray</File_Uri>
  </Next_Reply>
  <Next_Reply>
    <Server_Address>http://192.168.1.100:8888/</Server_Address>
    <File_Uri>http://dbpedia.org/resource/Missy_Elliott</File_Uri>
  </Next_Reply>
:/Result>
```

Figure 12. Query template and result to query specific file sources.

3.7 'Web Service' Service

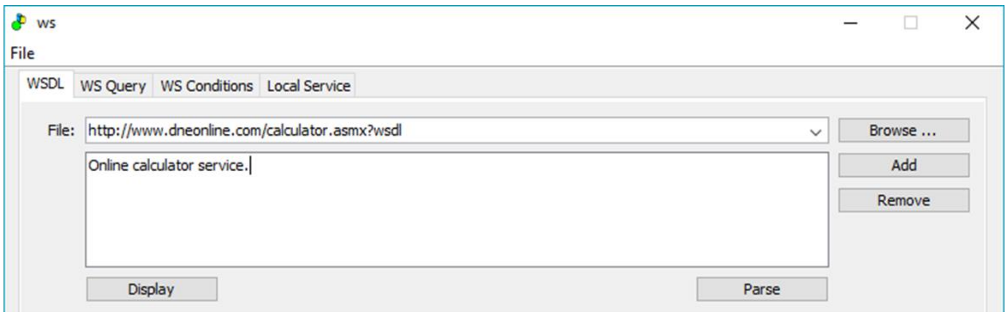
This service can be used to dynamically configure and invoke Web Service calls, using WSDL and the SOAP protocol. It also has limited functionality to run a web service locally, when it can be invoked through another instance of the service, using the default licas RPC protocol. After receiving a reply from a web service, the system can optionally save it, or filter it using a query condition. It can also run the query process on a timer, to periodically make a call, evaluate and save the result. For advanced programming, something like an email service (see section 3.5) can be added as a utility service and asked to send an email message through a script, when a suitable reply is received. The service is able to dynamically parse a WSDL document and construct a Web Service call from that. You need to enter the call details by selecting the method to invoke and entering a value for each field in the method, but the GUI interface interactively helps with that. You can then also set reply conditions. For example, you might want the reply to match with a basic query condition. You can also choose to save the successful replies to a file.

3.7.1GUI Client Interface

The Web Service GUI client interface is shown in Figure 13. to Figure 15. There is a small menu for loading or saving an existing configuration. The service runs a lot of threads and so some care should be taken when closing it down. You should probably use the `Exit` menu option. Also, if you are running some calls, new ones will probably not work until the current batch has finished. The GUI then contains 3 client panels and 1 server panel. The first client panel is for loading in a WSDL description of a web service, the second is for selecting the method to call and the third panel is for configuring the query and call conditions.

WSDL Description Panel

The first panel in Figure 13 contains a `Browse` button. This allows you to browse to a WSDL file stored locally, or more likely, you can enter a remote `HTTP address`, for accessing an online file. If you then click the `Add` button, it is permanently saved to an address book and you should also add a brief description first. If there is a fault with the address, you can use the `Remove` button to remove it. When the GUI is re-opened, the saved addresses are automatically retrieved and listed. The `Display` button will load and display the whole WSDL file in a text area for this panel, while the `Parse` button parses it into methods for the other panels.



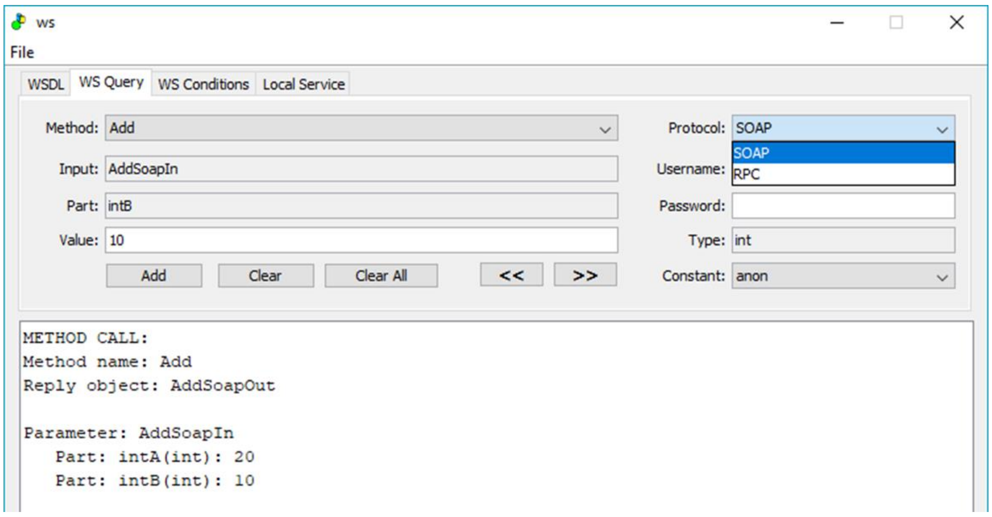
File: http://www.dneonline.com/calculator.asmx?wsdl

Online calculator service.

Figure 13. Panel to load in a Web Service description.

Client Query Panel

After parsing the wsdl document, the second `WS Query` panel, shown in Figure 14, can be used to select the method to invoke. In this case, the selected method is called `Add` and the reply is shown in Figure 15. Note that for SOAP calls, the protocol type must be SOAP and this is really the only option for WSDL construction. There is another RPC protocol that you have to use if trying to access a web service running on licas itself.



Method: Add
Input: AddSoapIn
Part: intB
Value: 10
Protocol: SOAP
SOAP
RPC
Username:
Password:
Type: int
Constant: anon

```
METHOD CALL:
Method name: Add
Reply object: AddSoapOut

Parameter: AddSoapIn
   Part: intA(int): 20
   Part: intB(int): 10
```

Figure 14. Panel with method to execute and values.

The method details are entered as follows:

- `Method`: this is a list of methods, parsed from the wsdl document. You select a method name first, which resets the other fields.
- `Input`: this is the related input section for the selected method, where the root name is displayed. You then need to enter a value for each part of this section.
- `Part`: this is the current input sub-section part, or parameter, that you enter a value for. You use the Add button to enter the value, when the part field should change to the next variable. If a mistake is made, you can click the `Clear` button and start again. The two arrow buttons are useful for traversing back to previous values, to change them and then traversing forwards to the parameters with un-entered values.
- `Value`: this is where you actually enter the part value, before you click `Add` to add it. If you leave the `Value` text box empty, you can add a `Constant` value instead, which gets automatically selected. Also, if there is a set of enumeration values for a particular parameter, they should be added to the constant list for that parameter, when it is selected. Each time you 'Add' a new value, the whole method description is displayed in the text area. In the figure, there is the method name, reply type and a parameter part with a value.

Passwords

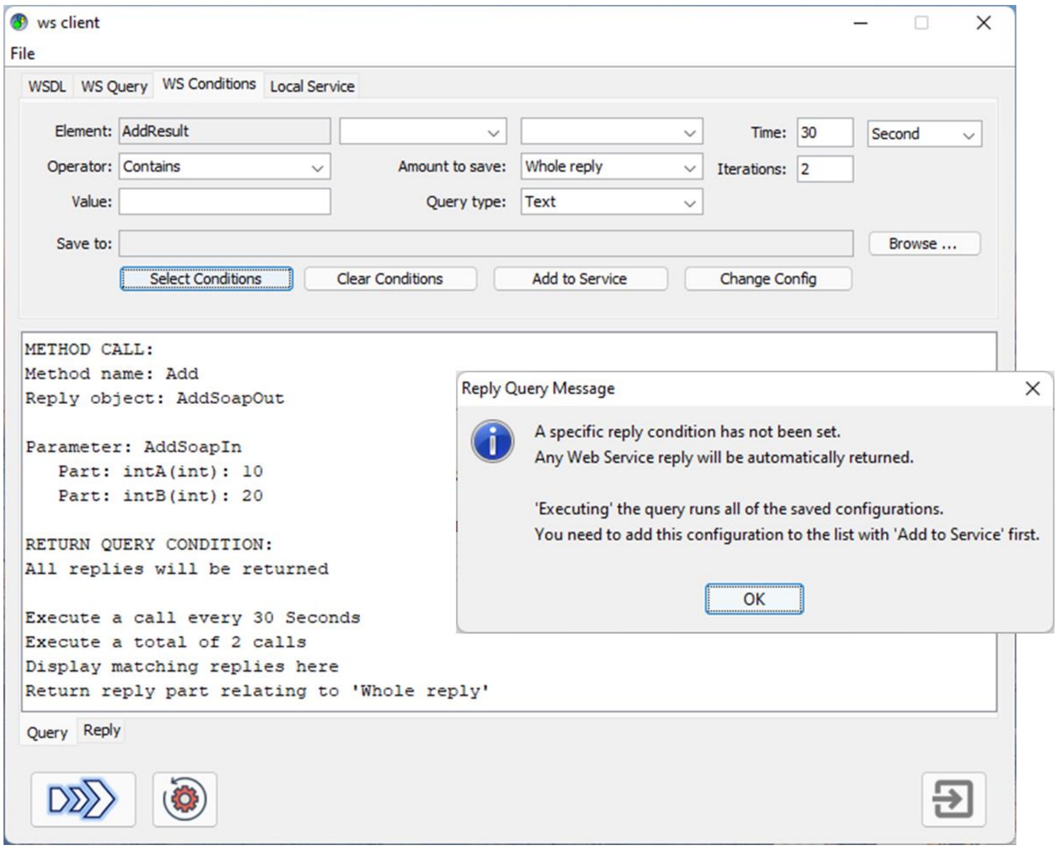
Some web services are private and require a `Username` and a `Password`. There are two fields on the right to allow you to enter these values. If text is entered into these fields, then when the web service call is constructed, they get added to it. As a technical point, they are added to the SOAP call header, not to its main body, in case that causes a problem. It is understood that this is typically where the access information is requested from, but that is not always the case.

Call and Query Conditions Panel

The third `WS Conditions` panel, shown in Figure 15, allows you to enter call and query conditions. You do not have to set a reply condition, when any reply will be accepted, which would be the same as simply executing the web service call. If you do include a query condition, then the reply is parsed to try to match a section of it with the condition.

The top section of this third panel therefore lists the reply object type and then beside that, sub-parameters to two further levels and this is the depth to which a query section can be selected. If the whole reply is included, then a query could simply be a check that the reply 'contains' some statement and the whole reply will still be saved. If a sub-section is specified, then information starting at the 'sub-element' can be asked for. So, if the reply returns a list of several elements, this filter can require a specific element at a specified depth before the reply is accepted.

The query can also be executed in a time loop and for a maximum number of executions. The time can be in seconds, minutes, hours, etc and one execution takes place each time interval. If a number of iterations are specified, then the query process will stop after that number. The filter can be applied to each timed reply and the result saved or notifications sent when there is a matching result. The example of this section uses a timer, but no sub-sections or query evaluation. The reply query and conditions are then displayed below the method call. If these are what you want to do, you can click the `Select Conditions` button to save the configuration. A message is then output describing the current setup. Note that the call has only been configured at the moment and you need to then send it to the service for executing. Therefore, click the `Add to Service` button to send an instance of this query to the service itself. To then start the web service invocation process, click the `Service` icon button at the bottom left pf the GUI. Any results are then shown in the `Reply` panel. Note that the `Reset` button will clear the current query list, but it will not stop any processes that are currently running.



Element: AddResult
Operator: Contains
Value:
Amount to save: Whole reply
Query type: Text
Time: 30 Second
Iterations: 2
Save to:

```
METHOD CALL:
Method name: Add
Reply object: AddSoapOut

Parameter: AddSoapIn
   Part: intA(int): 10
   Part: intB(int): 20

RETURN QUERY CONDITION:
All replies will be returned

Execute a call every 30 Seconds
Execute a total of 2 calls
Display matching replies here
Return reply part relating to 'Whole reply'
```

Reply Query Message

A specific reply condition has not been set.
Any Web Service reply will be automatically returned.

'Executing' the query runs all of the saved configurations.
You need to add this configuration to the list with 'Add to Service' first.

Figure 15. Panel with reply query conditions.

To summarise again, there is an idea that you want to retrieve some information based on particular values, but you may wish to save the reply only if a reply element matches a query value. This can be configured as follows:

- You can always save the reply, or when saving a successfully evaluated reply: the list of specified (sub)parameter names with the query condition and value will define what part of the web service reply is evaluated.
- There is then a `Store` combo box that defines what section to save if the evaluation is true. If you want to evaluate a part of the reply but save the whole web service reply, for example, keep the `Store` combo box value as 'Whole reply'. If you want to save only from the evaluated reply (sub)part onwards, set the combo box value to 'From sub-element', or 'From sub-sub-element' depending on the required level of nesting. Note that the outer-most element is sometimes a container for a list of replies and so you might want only a selection of those.
- Every time you enter a value, you can click the `Set` button to save it. The description is displayed in the text area. It may take a couple of goes to get it right, but it is not difficult to set up.

- You can also specify a time period for executing the call. The default value of 'None' will mean that the web service is invoked only once. In that case, you can manually use the same web service several times. If you start and stop a loop, this then disables the web service from starting again, but manual execution should still be possible.
- You can also browse to enter a save file path, to where any replies that match the condition will be saved.

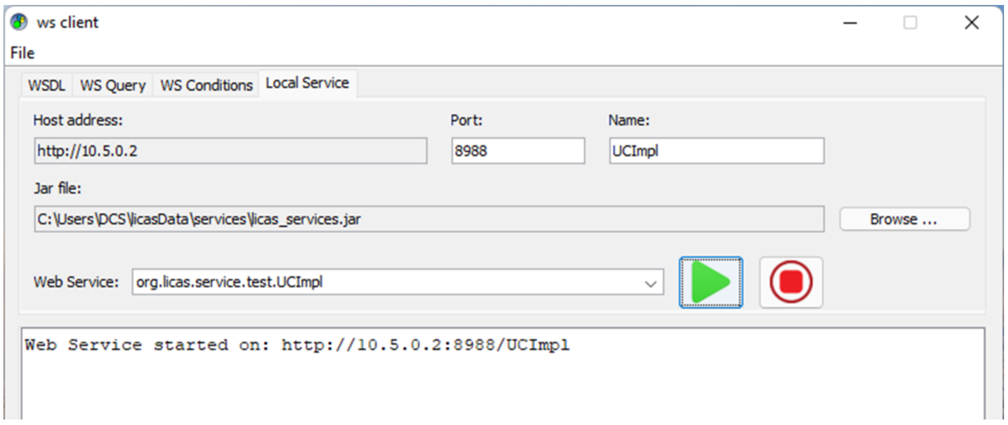
Adding Call Configurations

You can add more than 1 call configuration to the service and then ask the service to execute all of them at the same time. This can include time loops and iterations that may be different for each call. You therefore need to create the config first and then add it to the service. Several buttons can help with this as follows:

- The `Select Conditions` button adds the current settings to a call list, but does not transfer it to the client licas service yet.
- The `Clear Conditions` button clears the current field values.
- To transfer the call to the service, you need to click the `Add to Service` button.
- After adding calls, you can remove them again by clicking the `Change Config` button. This lists each call in-turn and asks if you want to keep it.
- To execute the calls, you then click the `Execute` button, when you are first shown a summary of calls you are going to make.

3.7.2 Interface for Local Web Server

There is also limited functionality to run a web service from the GUI, in a minimal server. This is shown in Figure 16. When you open this tab, the local server ip address is shown and a possible port number is specified. You can change the port number if you wish. To load a web service, you need to load in its Java classes. These should be written as an interface and implementation, where the implementation name is 'interface+Impl'. You can `Browse` to a Java jar file when all of the classes will be listed. As an example, the default services jar file has a test web service with an interface called `UC` and an implementation called `UCImpl`. When you select a class, its name is added to the end of the address that it will run on. The 'licasUserGuide' gives more details on how to program the web service. You select the implementation class and then click `Run` to start the web service. The `Stop` button should stop it again.



Host address: http://10.5.0.2
Port: 8988
Name: UCImpl
Jar file: C:\Users\DCS\licasData\services\licas_services.jar
Web Service: org.licas.service.test.UCImpl

```
Web Service started on: http://10.5.0.2:8988/UCImpl
```

Figure 16. Web Service running in separate minimal server.

The service runs in a separate minimal server and so it cannot be accessed through the licas communication protocol, but needs to be accessed directly. It is slightly different to the SOAP specification, requiring a stub class and so you need to select the RPC option when sending a query. The query invocation, or client-side process, is still done through the licas communication protocol, which then invokes the web service address directly. Apart from that, you load in its WSDL document (address?wsdl) and construct the query from that. The reply may not be in XML and so further parsing may not be possible.

3.7.3Web Service Config Script

As well as parsing the WSDL document, it is possible to save the whole configuration that you use to make a Web Service call, so that you can load it in again without requiring to enter each value automatically. After 'Setting' all of the parameter values, you can use the `Save Config` menu option to open at the scripts folder, to save the configuration in XML format. The next time you start a Web Service service and open the GUI, you can load in the script again using the `Load Config` menu option, where all of the values should be set in the form panels. After re-loading the script, you still need to add it to the service.

3.7.4Threading Issues

The Web Service service uses a more complicated threading setup. The service running on the licas server and the service interface both run their own loops. The interface uses a default value of 5 seconds for sleeping, while calls to a web service use the user-specified value. The GUI periodically invokes the web client service through an interface, retrieves any successful replies and displays the result. So this is a different loop that the interface class uses. The licas client service uses its own autonomous loop to make calls on remote web services, depending on the submitted configurations. This is even for single instance calls, so that several configurations can be loaded and run. When the process is stopped however, the thread cannot be started again and so this effectively kills the service. You can use the `Stop` button to stop the control loop early but not kill the threads.

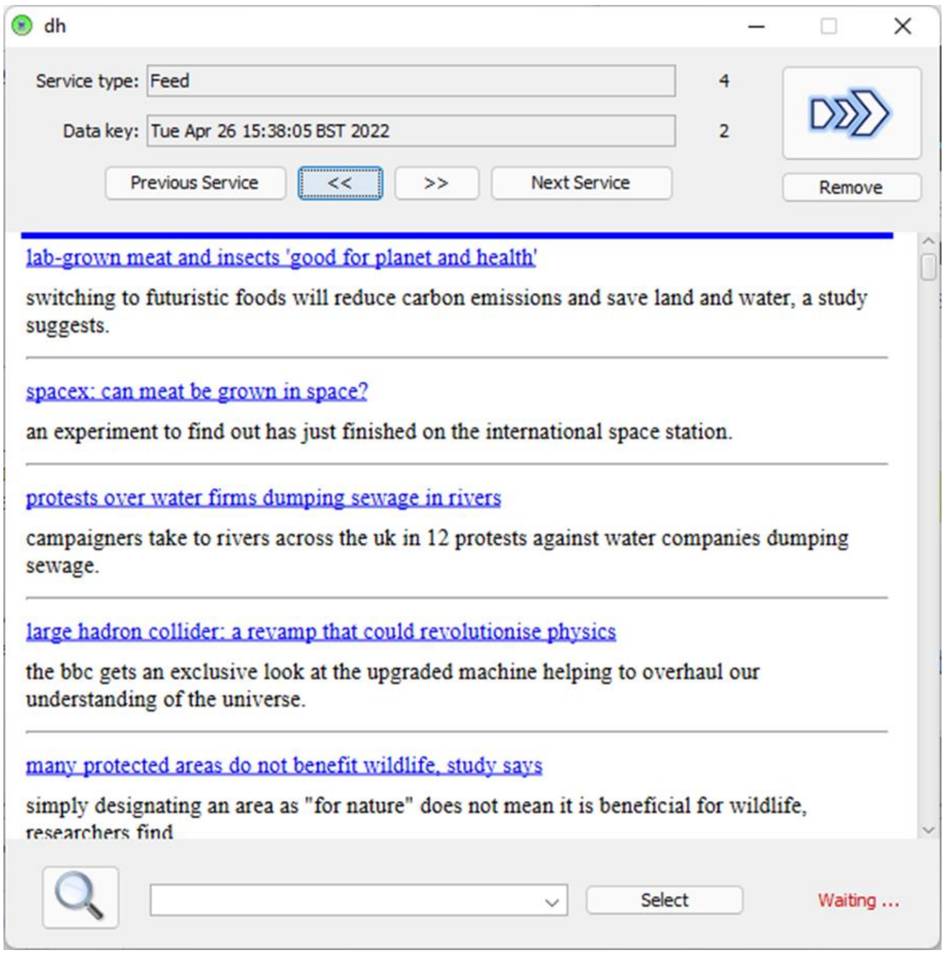
To shut the service down again, the `Stop` button will stop the execution process, but keep the service loaded on the server. When the GUI closes, the service is again shut-down, but it can probably be re-started because the threads are not automatically killed. As with the other apps, you can open the GUI on them and run it again.

3.7.5 Mobile Devices

Using the Web Service from a mobile or remote device requires a slightly different setup, where some configuration on the service itself is required first. All of the interaction is with the client service only, not through the GUI, and so what you need to do is create and save some web service configurations. You then start the service and load the configurations in (through the GUI, section 3.7.3), but do not then change or add them to the service. Instead, click the `Start` button to start the service control loops. It is then possible for the mobile device to communicate directly with the client service and ask it to load and run a particular configuration. The result is saved and can be retrieved for display. This is what the example mobile app does.

3.8 Data Hub Service

Like the scientific Behaviour Mediator, this is an aggregating service that is automatically sent text-based results from the other services. If you load this service onto the server, any of the text-based services will look for it and send it their query results. You can then run a matching query from the service over the stored data cache and any matching text from any of the services will be listed. It can therefore, also be used as a cache for all of the text-based information retrieved during a session. The interface is shown in Figure 17.



Service type: Feed 4

Data key: Tue Apr 26 15:38:05 BST 2022 2

lab-grown meat and insects 'good for planet and health'

switching to futuristic foods will reduce carbon emissions and save land and water, a study suggests.

spacex: can meat be grown in space?

an experiment to find out has just finished on the international space station.

protests over water firms dumping sewage in rivers

campaigners take to rivers across the uk in 12 protests against water companies dumping sewage.

large hadron collider: a revamp that could revolutionise physics

the bbc gets an exclusive look at the upgraded machine helping to overhaul our understanding of the universe.

many protected areas do not benefit wildlife, study says

simply designating an area as "for nature" does not mean it is beneficial for wildlife, researchers find

Figure 17. Data Hub Service.

3.8.1 Data View

When the information services return some data, they look for a Data Hub service and if one is loaded, they send the result there, where it is stored. On the Data Hub GUI, the `Execute` button will then update the view by loading into the GUI any of the currently cached data entries. They are saved under service type and then a timestamp key for each different transaction. The number of service types and entries for the currently selected type is also displayed and you scroll through these, or you can scroll directly to the next/previous service type. The `Remove` button will remove the current data entry.

3.8.2 Data Query

At the bottom of the form, you can open a query form using the `Query` icon button. This is the licas query form that allows you to configure an XML-based query with a pattern and set of constraints. It is easiest to simply indicate what terms you want to include (contains) in the text. You then submit the query to the data cache using the `Add to Main form` button on the query form, and this also executes the query in this case. Any matching data is listed in the bottom combo box, again as a Type-Timestamp key. After selecting an option from this combo box, click the `Select` button to display it. If you hover your mouse over the Select button, the current query should be displayed.

3.9 Message Board Service

This service is still a work in progress and has not been updated recently. It is different to the instant messenger in the way that it shares and distributes messages among all members using the app. There is no central server, but rather, the app retrieves its information for a list of servers. If any service or user shuts down, so long as any other user is still running, the messages can still be retrieved. Also, all of the addresses for a particular group (hashtag) can be shared and so you can connect with people or users that you are not initially familiar with. The idea is to try to save some time by making all of the messages available from any of the other message board services that are registered and running, but it can also provide a randomised set of information blogs. It can also send different types of content, or at least indicate different content types in the message. All of the content is duplicated to each service, which is resource expensive, but then there are the multiple copies. Therefore, an artificial limit is put on the amount of information that gets stored, which is currently set to 200 addresses or 200 messages.

3.9.1GUI Service Interface

To use the service, you need to connect to or join a group. To do that, you need to know the address and passwords for at least one service running in the group and also the group name. If there is no group, then you can create one and start your service running. This configuration can be carried out through the GUI interface, as shown in Figure 18, where the following entries are required:

- `Group name`: the name of the group you want to connect to. This is a type of hashtag and is for identification purposes only.
- `Service`: the shortened service address notation. This is actually the REST-style address of the full server IP address plus the service name. The short notation is the URL first (http://ip address:port) and then '/' plus the service name – mb1 in this case.
- `Server Pswd`: the server password.
- `Service Pswd`: the service password.

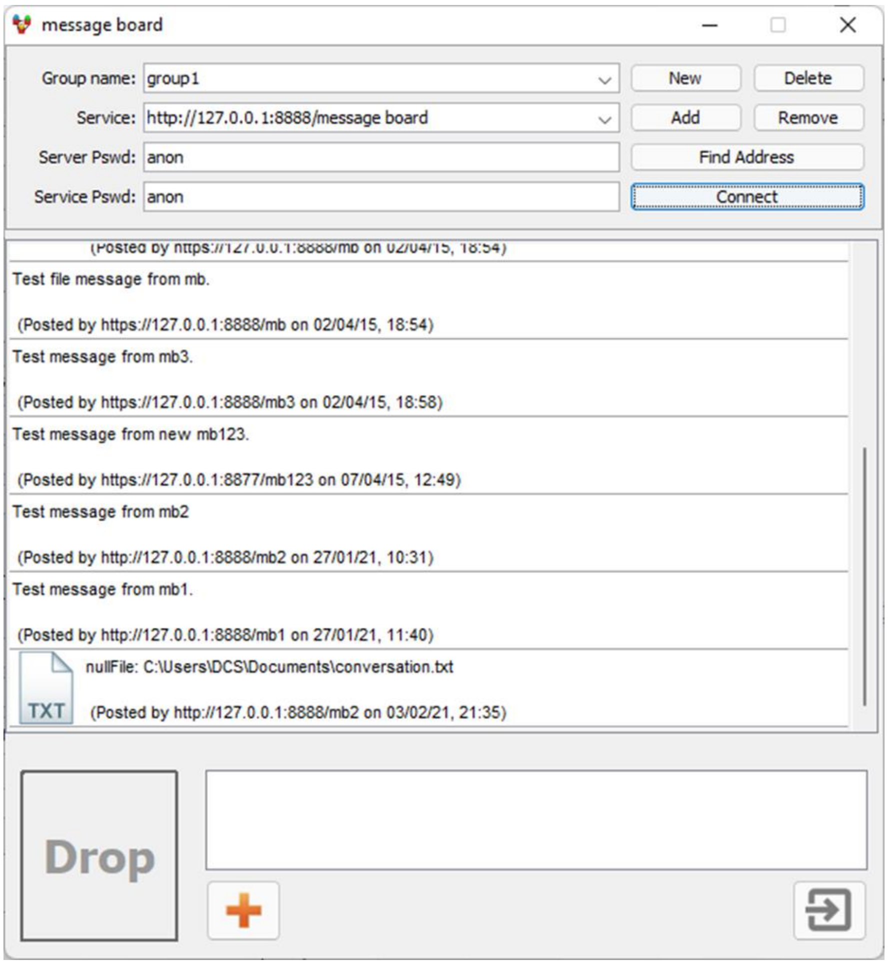
To create a network of users, you just need to connect to any other user in the group. Addresses and messages will then randomly get shared between all of the services, resulting in them being available to all of the other services running in the network. You therefore do not need to manually enter these details as the service performs the operation automatically. A text message can be entered in the bottom text box and posted as is. You can also drag file or image references to the `Drop` zone to add them. They get added as file path descriptions. Each message is also tagged with the service that added it and a date. The idea would then be to also run a file service from which a remote file can be downloaded, for example.

3.9.2 Address Book

The address book is not used in exactly the same way as for the instant messenger. At the top you can enter a new group name, maybe something similar to a hashtag name. You then click `New` to add it. This should also add your own service address and passwords. If you are the initiator of the group, you then click `Find Address` and automatically afterwards, `Connect`, to start your service running. If you wish to join an existing group, you need to enter the details of at least one service that is currently running in the group. Required is the full HTTP address plus the service name and also the server and service passwords. The address is the short notation for the full licas Handle description, as shown in Figure 18, in the Service combo box. You then click `Add` to add the service address.

3.9.3 Message Boxes

There are three different types of message that can be added from the bottom of the GUI. A basic text message can be added by entering the text and clicking the `Plus` button. This should be displayed on your own message board first, but then should be automatically transferred to all of the message boards (services) that you link with. There is also a `Drop` zone for dragging document or image files. When these are added, a small image icon is used to describe it, but the main message is the file path to the document. This is again displayed on your own board first, which is then available to copy to any other board. All of the messages include the sender service address and date. You cannot retrieve the remote file directly, but you might use the file service to do that, for example.



Group name: group1

Service: http://127.0.0.1:8888/message board

Server Pswd: anon

Service Pswd: anon

(Posted by https://127.0.0.1:8888/mb on 02/04/15, 18:54)

Test file message from mb.

(Posted by https://127.0.0.1:8888/mb on 02/04/15, 18:54)

Test message from mb3.

(Posted by https://127.0.0.1:8888/mb3 on 02/04/15, 18:58)

Test message from new mb123.

(Posted by https://127.0.0.1:8877/mb123 on 07/04/15, 12:49)

Test message from mb2

(Posted by http://127.0.0.1:8888/mb2 on 27/01/21, 10:31)

Test message from mb1.

(Posted by http://127.0.0.1:8888/mb1 on 27/01/21, 11:40)

nullFile: C:\Users\DCS\Documents\conversation.txt

(Posted by http://127.0.0.1:8888/mb2 on 03/02/21, 21:35)

Figure 18. Message Board Service GUI.

4 Additional Service Modules

The system has been written to create services, by using a ClassLoader to load-in class instances. This allows in theory, external classes and jar files to be used, to create instances of new service types. The All-in-One GUI also accommodates this by providing the config setup to declare new modules, in the form of external jar files. To demonstrate how this would work in practice, there is a ‘modules’ page on the main DCS web site, from where you can download and run some. The default ‘licas_services’ package is written to be slightly integrated with the core classes. These new modules however can be loaded completely separately of it, even if they are built from the same base platform. You need to declare the module to the service factory, but then it should be loaded in automatically and saved to the default ‘modules’ folder. See the web site for details.

4.1 Text Search Module

This is the integrated Web Query service. It has been added as a separate module, but it is included now as a default service with the main download. So the external modules do not now include this option, but you can still follow the instructions to write and add your own.

5 Scientific Services, or for Programming

This section describes some service classes that you might extend when writing more scientific code. These classes contain the default functionality that is used by the system. You can load and run a service without these classes, but extending them would make your own service most compatible with the rest of the system.

5.1 Programming an Application Service

To write your own service, you need to perform a small amount of programming, to link your classes with the default licas system. This section explains how to do that, using the instant messenger service as the example. This section describes what methods need to be called and what they are for. The default architecture is to have a Windows GUI form interface, with an integrated service interface class, where the interface class then talks to the actual local service running on the local server. This is displayed in Figure 19. The GUI form needs to implement the `ServiceGuiDef` interface of the base licas package. This is so that the default All-In-One GUI can recognise it and call the default initialisation methods.

You do not need to have a service interface class as well, but the default package uses them and there are two base classes that you can extend. The 'control' package classes, for example, extend either `ServiceInterface` or `AutoInterface` of the main 'org.licas.service.model' package. The Auto version extends the base `Auto` class and therefore provides the autonomous features to the control class. Both provide some additional default functionality, for configuring the interface with the GUI. These are the interface class shaded area in Figure 19 and have an implementation of the set details method, for example:

```
setServiceDetails(String theServiceUuid, String servicePassword, String serviceKey,
Element theServerUrl, String serverPassword)
```

Where:

- `serverUuid` is the uuid of the related service running on the local server.
- `servicePassword` is the password for the related service.
- `serviceKey` is the admin key for the related service.
- `theServerUrl` is the full URI of the local server.
- `serverPassword` is the password for the local server.

After that, the service interface class can make calls on the actual service, on behalf of the GUI interface. You can then fully interact with a service running on the server, through its own GUI interface. One idea is that the GUI view can change, but keep the same interface connector code with the service. So a type of personalised view can be written and quickly added.

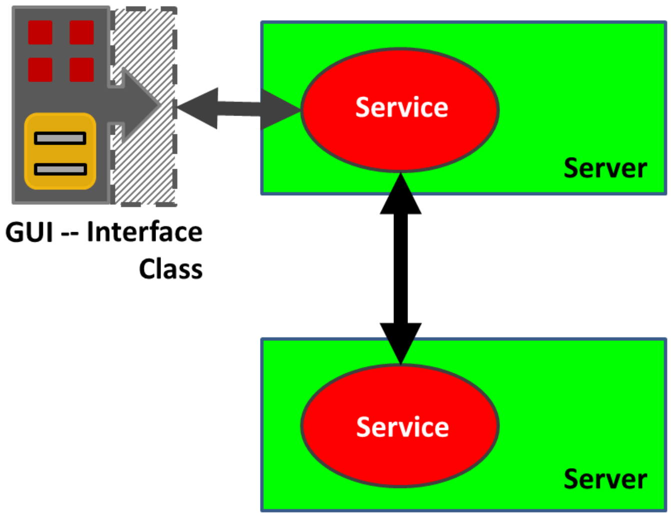


Figure 19. Service GUI with interface, calling its service, to call another service.

Also, there is a base `DynamicJFrame` class that extends JFrame. This should be used with the system because the main GUI can look for some methods in it that help with configuration of the related service. Instead of extending JFrame when writing the service GUI, extend DynamicJFrame.

5.2 Auto Service

The `Auto` class is abstract, but does implement a service Thread's `run` method. This is the method that starts the thread, which will invoke a behaviour to periodically execute something. To write the behaviour application-specific functionality, the Auto class is extended and the `behaviourAction` method, at least, must be implemented. One example of this is the `LinkService` service. The `Auto` class by itself has the framework in place to manage messages with the autonomic manager, with only a few methods requiring updating/implementing for complete functionality. The 'licasUserGuide' describes this in detail. This means that you can execute a new behaviour activity by extending the `Auto` class and changing just one or two methods, while keeping most of the class the same. The 'licasAuto' document gives full details on how to run the autonomous features.

5.3 Scientific Services

Scientific services can be configured and run through the scientific panels of the All-in-One GUI. There is a default example to show you how it would work, with the classes that are loaded at startup. The default behaviour class is a `LinkService` and it extends Auto. If you do not want to use Auto, then a `DataService` might be tried. An evaluation service however might typically be added as a utility service to another one and it needs to implement the `EvaluateServiceDef` interface. If you load in a script and follow the instructions to start behaviours running from this panel, the appropriate classes will be automatically loaded and run. The default link service behaviour retrieves the data from all other services and passes this to the evaluator service for evaluation. The evaluator processes the data and replies with what links to make. Changing the evaluation service can then change how the network nodes link with each other.

If you also need to provide or use a data generator class, it should implement the `DataGenDef` interface. This is used to generate random data that can be processed by the algorithm. The data would then be stored in the admin 'Data' section. Alternatively, this could happen during initialisation, where the 'licasAdminGuide' describes the format for passing a data value into the service in this way. The value can then be retrieve from the `AdminInfo` object using the `getData()` method, and your extended service will then process it in whatever way is appropriate. The evaluator can then be given this result and can evaluate the processed data in the most suitable way. Therefore, a different data generator or evaluator can produce different results. The default LinkService shows how to retrieve the data and how to pass it to an evaluator and how to update links. It also shows how to automatically try to retrieve passwords.

5.4 Server Services

The Http server can now store and execute methods on more centralised services. The server service is recognised by its 'type' and not its 'ID'. The calling service therefore does not need to know the service password, only the base server password. Some of these are additional classes that are part of the server, to execute some functionality and some are separate services that are loaded and then invoked.

5.5 Behaviour Mediator

The behaviour mediator can be used to draw some basic graphs. Services extending the `DataService` class, when executing their behaviour, will automatically look for the mediator service and send it information relating to their current state or result. This is performed through their `sendInfoXML` method that retrieves their data result and tries to find a behaviour mediator service to send it to. The data result relates exactly to the data element of the service admin metadata object that can be used to store any type of information. The behaviour mediator then stores this information and when asked, can pass it to a GUI interface for display purposes. The current version will also display clustering information, based on the feedback that it receives.

5.5.1Graphic Interface

The behaviour mediator has an interface, shown in Figure 20. The text panel is always available and you can configure and add a number of graphic panels to display the service replies visually. Currently, there is a basic list display of clusters, or an aggregated view of them. A detailed description can be found in the 'licasProblemSolver' document.

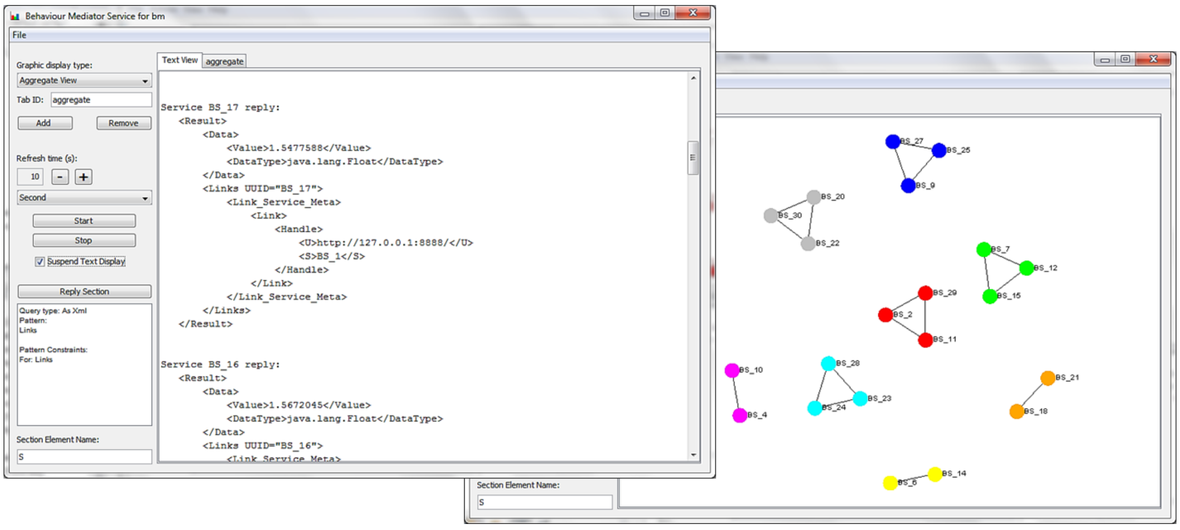
5.5.1.1 REGISTER FOR CLUSTERING

As well as the `Text` panel, there is also a default `Main View` panel. This can be used to perform an obvious clustering of the services in the main GUI graphic and return a representation of the clusters instead. In the main view panel, there are two buttons – one to register the mediator as the main GUI view and one to de-register it again. If you register the mediator, then before the main GUI draws its server graphic, it sends the metadata to the mediator. The mediator would then create clusters based on the sets of permanent links. The mediator then returns the clusters as the metadata to draw, where only the cluster groups are then drawn, each as a cluster service. These are just graphic representations of the services. No new services are actually created. The point for this is that if there are a lot of services running, then it may be easier to see them classified as clusters instead of showing all services with their permanent links.

You can then enable metadata in the main GUI and select the cluster service to then see all of its child services. The behaviour mediator also prints out what services it has clustered in the text area. If the addresses are known, then the check box can be selected to remove the address part from the print-out.

5.5.2Drawing Graphs

You can load a behaviour mediator service onto a server and then open the GUI interface. This should be done before running or loading any tests. To start the mediator, you enter a loop time and click the `Start` button to start it running. The mediator can be used to possibly display more complex behaviour results. The figure shows text information that has been sent, for example, from the default problem solving process, which will now also send information to a mediator if one is available. Each service sends its data and linking information to the mediator for processing. The text view receives information from each service that is running, as an XML-document and then simply prints this out. The timer returns a batch set of information every *x* seconds and converts it for display or viewing purposes. If you have setup a graph configuration, you can save it to a file and re-load it in again using the menu options.



Graphic display type: Aggregate View
Tab ID: aggregate
Refresh time (s): 10 Second
[✓] Suspend Text Display

```
Query type: As Xml
Pattern:
Links

Pattern Constraints:
For: Links
```

```
Service BS_17 reply:
   <Result>
       <Data>
           <Value>1.5477588</Value>
           <DataType>java.lang.Float</DataType>
       </Data>
       <Links UUID="BS_17">
           <Link_Service_Meta>
               <Link>
                   <Handle>
                       <U>http://127.0.0.1:8888/</U>
                       <S>BS_1</S>
                   </Handle>
               </Link>
           </Link_Service_Meta>
       </Links>
   </Result>


Service BS_16 reply:
   <Result>
       <Data>
           <Value>1.5672045</Value>
           <DataType>java.lang.Float</DataType>
       </Data>
       <Links UUID="BS_16">
           <Link_Service_Meta>
```

Section Element Name: S

Figure 20. Behaviour Mediator Text Display and GUI.

6 Utility Services

This section describes utility services that should not be added to a server directly, but to another service, to perform some operation for it. The main All-in-One GUI has menu options for adding these to base services.

6.1 Linker Service

The dynamic linking mechanism is described in other documents, but it can be added as a utility service to any base service. Then, certain methods will look it up and add linking information related to the feedback that they receive.

6.2 Email Service

The email service can also be used as a utility service, where a base service might ask it to send a message.

6.3 Timer Service

A `TimerService` can be used to time method calls between services. This class is not actually a service itself and it does not extend any of the licas service classes. It can be used however to record the start and the end time of a method call for a parent service, as part of timing tests, for example. The default timer service works as follows:

There are two methods to call.

1. At the start of the timing, call the `startTime` method with the `method name` and an optional unique `ID`. This could be the uuid of the service being called, for example. If no ID is entered, one is created and the method returns the ID that it uses for the timing. This needs to be stored in the parent service, along with the method name.
2. When the method invocation reply is received, call the timer service again. Call the `endTime` method, with the `method name` and the unique `ID`. This method then returns the total time for the call, in milliseconds.

So this is a very simple class that will store the start time of a method call and return the total call time when asked. Most of the time processing then needs to take place in the parent service.

6.3.1Timed Callbacks

The timer service can also be used however to make periodic callbacks on the parent service. The `MethodInfo` class of the licas system can be used to describe and method and if the serviceUri is set to be the parent service itself, the method will be invoked on that service directly. You can setup a callback method very easily, where the following code is used in the Information Service GUI:

```
timerService = new TimerService();
methodInfo = new MethodInfo();
methodInfo.setServiceURI(this);
methodInfo.setPassword(passwordHandler.getPassword());
methodInfo.setName("getDisplayResourceContent");
methodInfo.setRtnType(TypeConst.BOOLEAN);
timerService.setCallbackMethod(methodInfo);

methodInfo = new MethodInfo();
methodInfo.setServiceURI(this);
methodInfo.setPassword(passwordHandler.getPassword());
methodInfo.setName("getSleepTime");
methodInfo.setRtnType(TypeConst.INTEGER);
timerService.setSleepMethod(methodInfo);

timerService.start();
```

So you just need to set the callback method and the thread sleep method is optional. When the thread is started these methods are invoked if they are present. They should just invoke the parent service's method and not expect to pass any parameter lists. The callback method (`getDisplayResourceContent` in this case) should return a boolean to indicate if the thread should continue. The sleep time method (getSleepTime in this case) should return the time in milliseconds (1000 ms equals 1 second), where a value <= 0 will also terminate the thread.