

Final Project Report

Mukesh Jha
mjha@masdar.ac.ae

Bikash Joshi
bjoshi@masdar.ac.ae

Puskar Budhathoki
pbudhathoki@masdar.ac.ae

Abraham Xiao
yxiao@masdar.ac.ae

December 11, 2013

1 Introduction

It was a successful team work which culminated in implementation of five features. All the team-members contributed with their full capacity. We worked as a closely related team and implemented the project with concept of agile software development paradigm.

2 Feature: Backup-Restore

This feature enables user to create the backup of application data and also helps to restore the it. It allows interactive GUI for users to facilitate the functionality of this feature.

1. Forum Thread
2. Pull Request

2.1 User Story

Title:

Backup/Restore Data

Description:

A user should be able to create a backup his Inbox, Sent and other application data and conveniently restore some key and message from remote location through intuitive GUI. This feature also provides selective restoration of only Inbox.

Acceptance criteria:

The Backup/Restore mechanism should have the ability to backup data from user specified location. The feature should be able to Backup data at ”./Backup” folder by default. After restoring the data, earlier message should be flushed.

2.2 Changelog

December 11, 2013

- bitmessageqt.py : Added a class NewBackupRestoreDialog and functionality of backup, restore and delete-backup.
- backuprestore.py : Introduced this to create the GUI.
- backuprestore.ui : Introduced this file to give other developers easy access to modify the GUI

To implement this feature, we created a GUI to enable users for preference. With active inputs from the bitmessage community, We created this feature as per their requirements.

The formal documentation of the class:

Class Reference [NewBackupRestore module]

The NewBackupRestore class specifies a functionality associated with GUI form created by backuprestore.py. It consists of following methods:

Methods

- Boolean verify_form(self): Verifies the inputs given by the user before implementing it.
- checkPermission(self): checks the permission if the task assigned by the use is feasible or not.
- BackupRestoreaccepted(self): implements the backup, restore and delete feature.
- selectDirectory(self): module to enable browser to brows through the directory structure.

2.3 Screen-Shots

The GUI of the implemented feature:

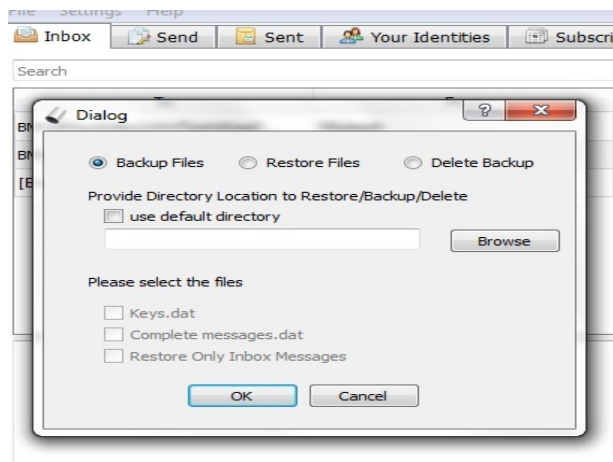


Figure 1: GUI

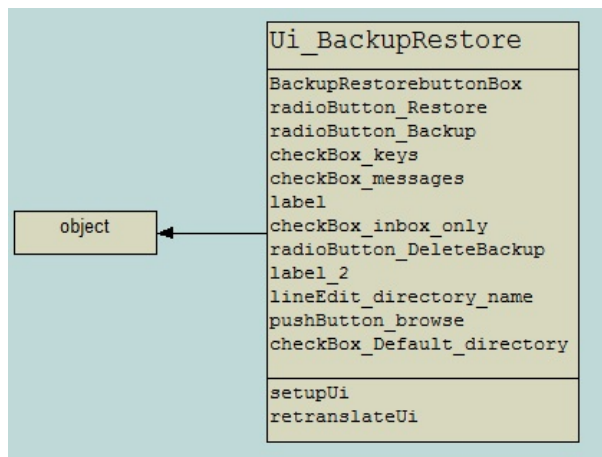


Figure 2: Class Diagram

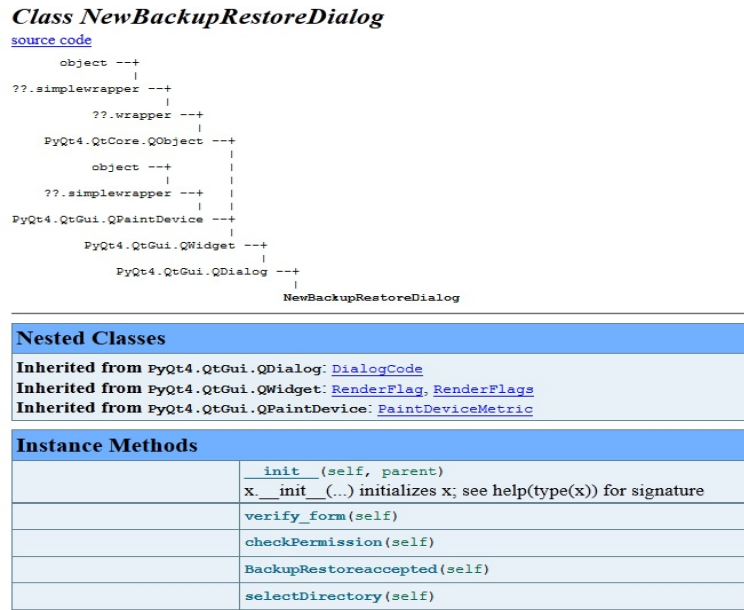


Figure 3: Method Diagram

3 Feature: add option ‘add Subscription to Address book in the context menu of subscription list

3.1 Introduction

This feature allows the users to add the selected subscription (from the subscription list) to the address book. We have added Add Subscription to Address Book option in the context menu for Subscriptions. When a user selects a subscription from the subscription list, he can see this option in the context menu which can be viewed by right clicking the selected subscription. So, when the user clicks this option, it will first check if the selected address is already present in the Address book or not. If it is already present in the address book then it displays an error message Error: You cannot add the same address to your address book twice. Try renaming the existing one if you want. But if the address is not present in the address book, it adds the selected address and corresponding label to the Address Book. And it displays a success message Entry added to the Address Book. Edit the label to your liking. Detailed description and screenshots for this feature are also shown in:

1. Forum Thread
2. Pull Request

3.2 User-Story

- Title: Add new subscriptions to address book
- Description: As a bitmessage user, I want the functionality to add new subscriptions to address book list so that I can save the subscription list in the Address book for future use.
- Acceptance Criteria: The Subscription list contains the address of subscriptions. Adding the addresses from subscription list to address book, should copy the selected address from subscription list and make a new record in the Address book.

3.3 Change Log

- Class: To implement this feature we have made changes to the init.py (bitmessageqt.py) class in the bitmessageqt package. We added pop up menu option for 'add Subscription to Address book' in the context menu of Subscription.
- Method: We have added a method on_action_SubscriptionAddSenderToAddressBook to bitmessageqt.py class. This contains the main functionality for adding selected subscription to the address book.

3.4 Screenshots:

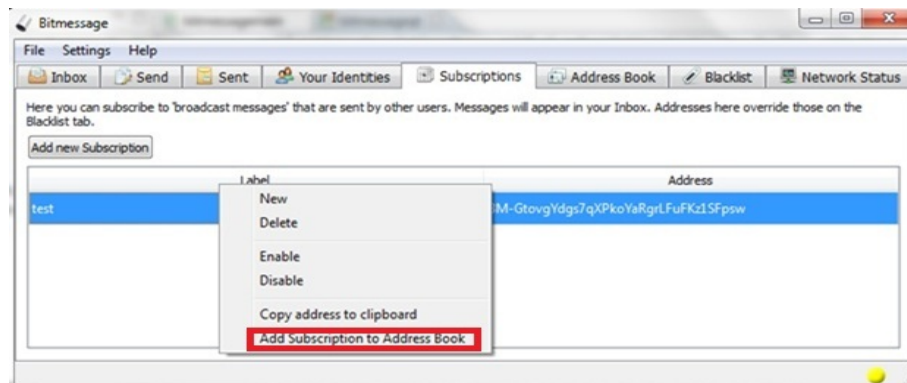


Figure 4: Screen showing Add Subscription to Address book option in the context menu for subscription

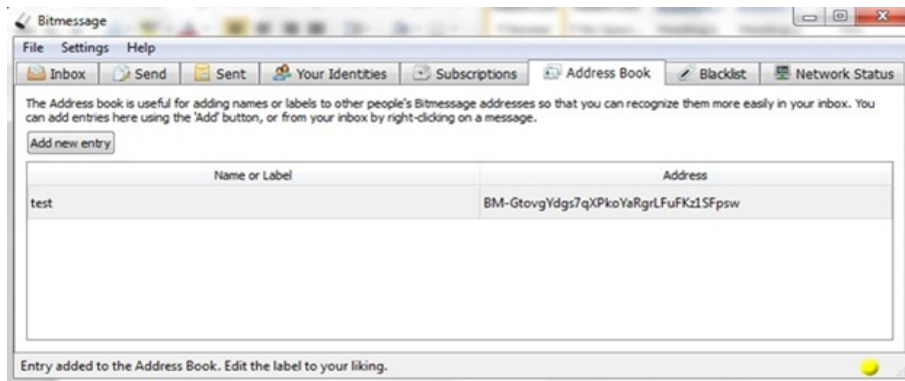


Figure 5: Screen showing the selected subscription added to the address book when the selected subscription is not already present in the address book

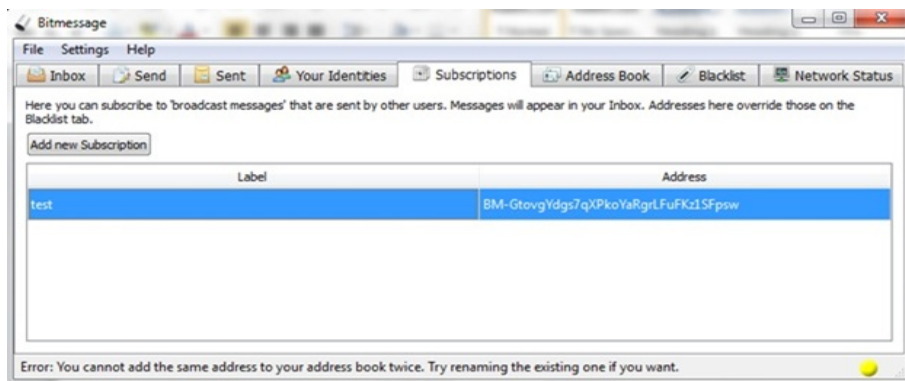


Figure 6: Screen showing the error message when the selected subscription is already present in the address book

4 Feature: When replying to a message, populate the 'from' field

4.1 Introduction

A user can have more than one identity. Each identity has a unique address, label and icon. Currently, when the user replies to a message the 'from' field is left blank. So, when user replies to a message sent to one of his addresses, the 'from' field should show the corresponding user label and icon. This feature helps to populate the 'from' field with the corresponding label and icon for the address to which the message was sent.

When the user selects a message in the 'inbox' and selects the 'reply' option by right clicking on it, the from field populates the corresponding label for the address of the user in which the message was sent to, whereas the address is displayed in

the label adjacent to the 'from' field. If that address does not have an address then it only displays the icon for that address in the 'from' field. Detailed description and screenshots for this feature are also shown in:

1. Forum Thread
2. Pull Request

4.2 User-story:

- Title: Populate 'from' field while replying to a message.
- Description: As a bitmessage software user, I need the ability to populate the 'from' field while replying to a message so that I can easily visualize the sender and receiver addresses in the reply message.
- Acceptance criteria: The message inbox has the ability to 'Reply' an existing message. The 'Reply' message will have a text field to populate the 'from' field

4.3 Change Log:

- Class: To implement this feature we have made changes to the `init.py` (`bitmessageqt.py`) class in the `bitmessageqt` package.
- Method: We have modified the `on_action_InboxReply` function in the `bitmessageqt.py` class. In this method we have added the code to set the index of from field label and icons to corresponding address to which the message was sent.

4.4 Screenshots:

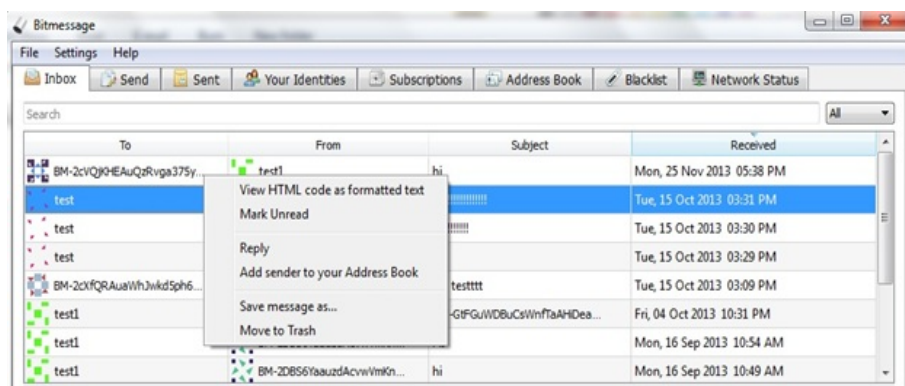


Figure 7: Screen showing when user replies to a message sent to his address having label and icon

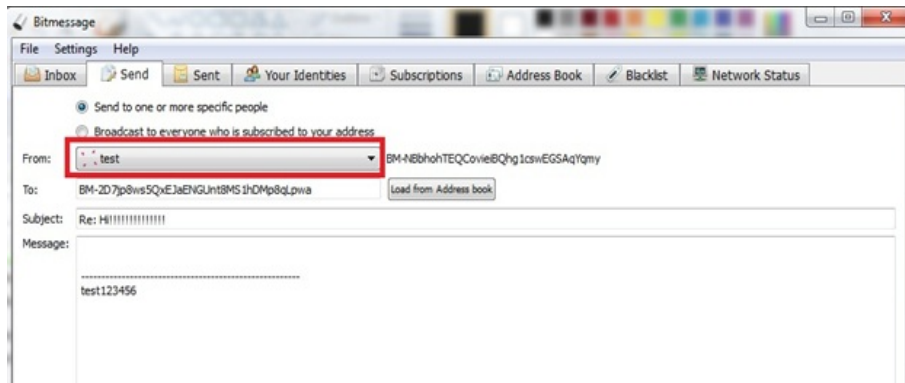


Figure 8: Screen showing the 'from' field populated with corresponding label and icon

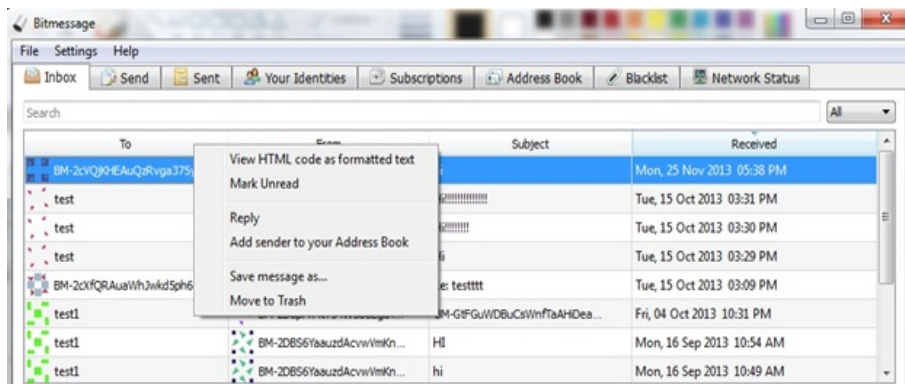


Figure 9: Screen showing when user replies to message sent to his address not having a label

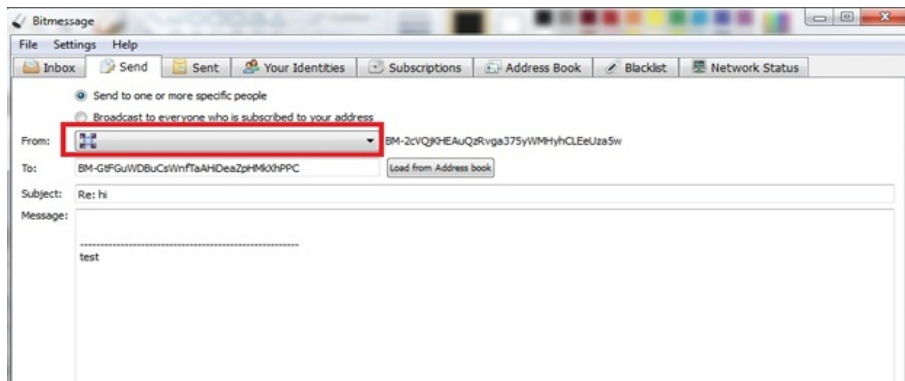


Figure 10: Screen showing the 'from' field populated with corresponding icon

5 Feature: Windows Installer

5.1 Introduction:

1. Window Installer: Window installer is an installation and configuration service based on data driven model giving installation data and instruction in a single package which focuses on what to install. It is used in Microsoft windows system for software installation, software maintenance and software removal. The advantage of window installer is that they synchronize with other installers and consistency in installed product internal database. This feature allows user to install BitMessage in Windows system, provide optional shortcuts in common location, allow installer to ask if it should be started at login or not and auto-update new Bitmessage if new version of Bitmessage exists in authenticated server. BitMessage has two dependencies, PyQt and OpenSSL. Window installer include both dependencies so that user can install it in windows which doesnot have PyQt and OpenSSL and it allows user to install BitMessage in common locations, flexibility of starting application at login and auto-check for new version of Bitmessage and update it if new version exists. Simple and powerful language for scripting which supports various GUI packages makes python suitable to develop application. But during compilation of python project from idle, it doesnt create executable as in C# , C++ etc. So, to create standalone executable from python project, separate plugin is required. Since PyQt and OpenSSL are two dependencies in BitMessage, proper handling of it during creation of executable should be done. PyInstaller is used to create executable from bitmessage project since it also support loading external dynamic link library so that we can resolve external dependencies. To resolve OpenSSL DLL in exe, libeay32.dll is included in standalone executable. For the window installer, a script is written in pascal programming language which also includes standalone executable and compiled with open source setup compiler INNO setup. Moreover, auto-update feature of Bitmessage is written in C#.
2. PyInstaller: PyInstaller is mutliplatform program used to convert python program into executable which works with any version of python since 2.4. It also supports loading dynamic link library ensuring full compatibility. Main goal of pyinstaller is to integrate third party package along with executable ensuring all dependencies are satisfied.

3. INNO Setup: It is a free script-driven installation system for window programs. In terms of feature and stability, it is competitive and even surpasses some commercial software. The main advantage is that it support multilingual, Unicode and digitally signed installs and uninstalls. Other advantages are it supports every window after 2000 and extensive support for 64 bit windows as well. It also allows customizable setup like full, minimal or custom and helps to create uninstall capabilities.
4. Auto-update: During installation, the version of the Bitmessage in text file is also downloaded. The auto-update feature checks the version of Bitmessage of local repository with the version of Bitmessage in authenticated server and update Bitmessage from authenticated remote server if new version exists over there. Authenticated web urls are provided by Bitmessage contributor for auto-update feature. The window installer with auto-update feature is tested in the 32-bit and 64-bit window machine which doesnt have python, PyQt and OpenSSL.
 - (a) Forum Thread
 - (b) Pull Request

5.2 User Story

Title: Window Installer

Description: As a window installer, it needs to be installed and configured in 32bit/64bit windows with installation data and instruction in single package. Similarly it should prompt the message to allow for optional shortcuts in common locations.

Acceptance criteria: The installer package should include all dependencies like PyQt and OpenSSL and it should be compatible with both 32 bit and 64 bit windows. Similarly the user should complete all the procedure prompted in installation dialog box to complete the setup procedure.

Title: Auto-update of window installer

Description: The application should be able to auto-update in each startup to a newer version if new version exists.

Acceptance criteria: The installer should be updated to newer version if the Bitmessage in local machine is of different version than the authenticated

server Bitmessage.

5.3 Changelog

The setup scripting is written in pascal programming language and can be compiled with open source setup compiler INNO setup .Similarly auto-update feature is written in C#. Change the location of "OutputDir" to user desired location to generate installer file and give location of BitUpdate.exe in "[Files] Source": in scripting file setup.iss. Compile it and get the Installer

5.4 Step-wise guide to installation

After we get the window installer from above process, the following steps should be done to complete installation

- (a) Window dialog box giving detail about installer information along with version is appeared.

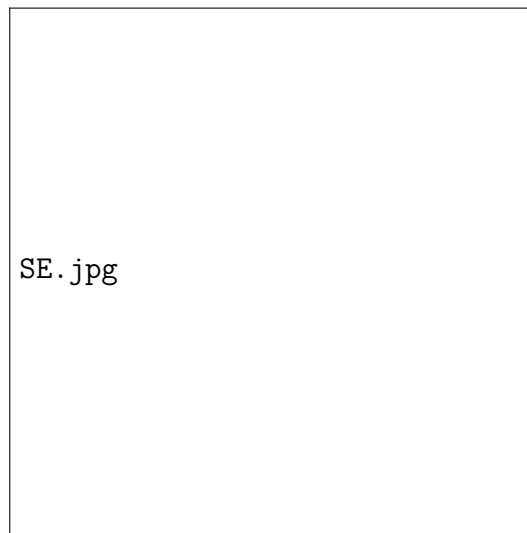


Figure 11: Bitmessge window installer with version

- (b) Window installer provides flexibility to install the application in any drive.

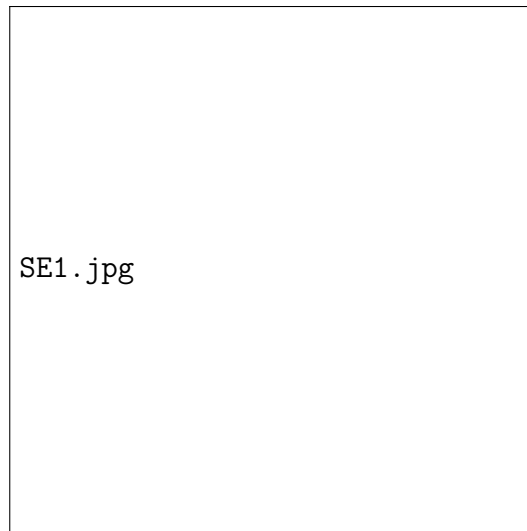


Figure 12: Installation Path

- (c) If the window installer is already installed in the path specified, it gives the information to user.

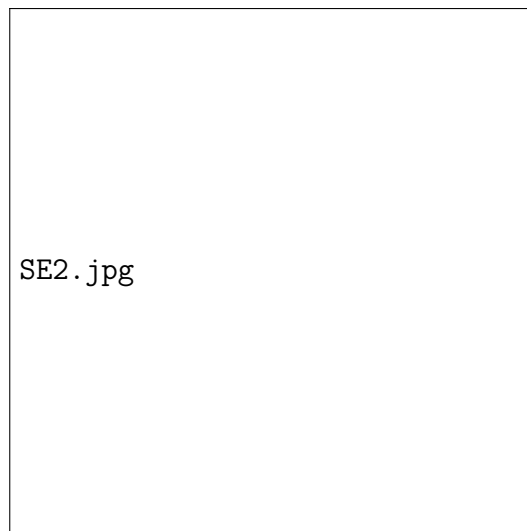


Figure 13: Warning prompt

- (d) It also provides flexibility to change the start menu name and the path.

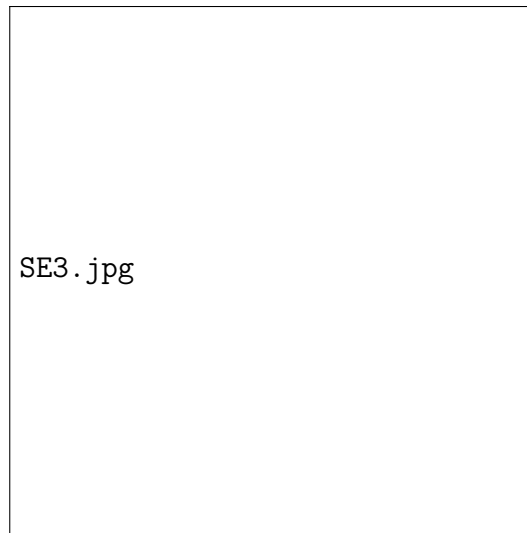


Figure 14: Start menu with installation path

- (e) Installer allow for optional shortcuts in common locations like creating desktop icon.

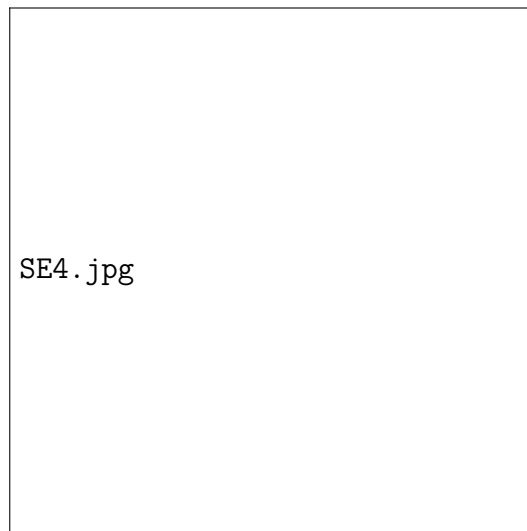


Figure 15: optional shortcut

- (f) Then after it gives the summary of the whole installation process.

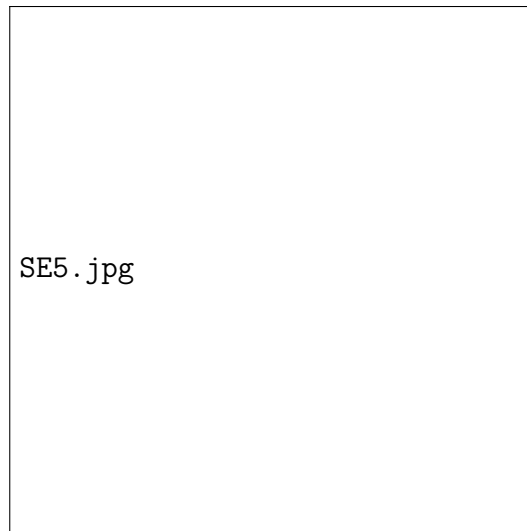


Figure 16: Installation summary

- (g) Finally it check version of Bitmessage from authenticated server and install window installer from it.



Figure 17: Installation progress