# Design: CFLEX2.0 Flex GUI Message Center

# Overview

## Requirement:

* Flex User should have facility to send the text message to helpdesk user and also can receive the mail from helpdesk user.

## Scope:

Message center functionality should be available to Flex trader to communicate with helpdesk.

# Implementation

## 2.1 IDL Changes

### 2.1.1 CMI IDL Modification

* None

### Other IDL Modification:

* None

## 3. Global Server

### 3.1 Classes & Interfaces

* None

## 3.2 Scripts

* None

## 3.3 Database Changes

* None

## 3.4 SetContext

* None

## 3.5 XML/DTD

* None

## 4. Client

### WebCAS Changes

### 4.1.1 Classes & Interfaces

### WebCasTextMessageService.java

### We need to add two new methods to process the send message request from trader and accept the message from helpdesk.

### WebCasTextMessageDispatcher.java

### Added three new methods to send the message to helpdesk, update the message status and get the mailbox of user for initial call.

### WebCasAccessManager.java

### Added a method to get reference for administrator service.

### WebCasInProcessSessionManager.java

### Added new methods in the interface for making query of user mailbox and update the message state.

### *public MessageTransportStruct[] getMailboxForUser(String userId) throws SystemException, CommunicationException, AuthorizationException, DataValidationException, TransactionFailedException;*

### *public void updateMessageState(int messageKey,int state) throws SystemException, CommunicationException, AuthorizationException, DataValidationException, TransactionFailedException;*

### WebCasInProcessSessionManagerImpl.java

### Added implementation for the new methods added in WebCasInProcessSessionManager for making query of user mailbox and update the message state.

### WebCasUserSessionAdminConsumer.java

### Added new method to accept the text message coming from helpdesk user.

### *public void acceptTextMessage(MessageStruct message)*

### WebCasUserSessionAdminConsumerImpl.java

### This is the implementation for consumer on webcas for accepting the text message from helpdesk user to trader.

### WebCasClientServiceImpl.java

### Modified method sendMessage to publish data to GUI on the listening service channel.

### Sequence diagram:

### Text Message Subscription:

### 

### Sequence Diagram:

### Send Text messages from user:



## 5. GUI

### 5.1 FlexGUI Changes

### 5.1.1 Changes for Message Center

To show a new pop-up window for pending trades, we need to create below classes:

1. **MessageCenterPanel.java**

Create a panel to show the table containing all the messages and button panel to create, read, print and delete a message.

1. **TablePanelMessage.java**

Create a panel to render the table containing all the messages. Initially it would give a call to getMailboxForUser() method of TextMessagingService to get all the messages.

1. **MessageButtonPanel.java**

Create a panel which has buttons to read, delete, print and create a new message.

1. **MessageCenterTableModel.java**

Create a table model to store all the messages received. It would be listening on IEC channel for text messages from Help desk users. If it receives any message from help desk, it would give a call to Web-CAS to update the message state as “Delivered”.

1. **MessageModelPrintJob.java**

Create a class which would extend JasperPrintJob for printing messages.

1. **MessageReaderDialog.java**

Create a dialog to show the message details.

1. **MessageReaderPanel.java**

Create a panel which has all the components required to show the message details.

1. **MessageStateImpl.java**

Create a class which will update message states.

1. **MessageTableConfig.java**

Create a table configuration for message center.

1. **SendMessageAction.java**

Create an action for sending a new message to help desk. This would use

int sendMessage(MessageStruct)method of Administrator.

1. **PrintMessagesAction.java**

Create an action for printing a message.

1. **ReadMessageAction.java**

Create an action for printing a message. This would call the

void updateMessageState(int, int) method of TextMessaingService to update the message state as “Read”.

1. **NewMessagePanel.java**

Create a new panel which has all the components required to create a new message.

1. **NewMessageAction.java**

Create an action for displaying the new message pop-up window.

1. **DeleteMessageAction.java**

Create an action to delete messages. This would also call the

void updateMessageState(int, int)method of TextMessaingService to update the message state as “Deleted”.

1. **PrintAction.java**

Create an action for printing a message from MessageReaderDialog.

1. **MessagingService.java**

Create an interface which has following methods of TextMessagingService.

void updateMessageState(int, int);

int sendMessage(MessageStruct);

MessageTransportStruct[] getMailboxForUser(String);

1. **MessagingServiceAPI.java**

New web API for messaging service needs to be defined which would actually implement all the methods of Messaging service.

### Sequence Diagram:

### Load Text messages for user:



### Send Text messages to help desk:

### 

### Update message state:

### 

### Message Subscription:

### 