Implementing Custom Logons

If you need to add fields to the logon message when you initiate a session, do it in the `toAdmin` callback of the `Application`interface. For example, a counterparty may ask to have a user identifier placed in the SenderSubID field and a password put into the RawData field.

In the `toAdmin` callback, if the message is a logon message (being sent to the counterparty) then add your fields. The sent message will include whatever changes you make to the Logon message in the callback.

```
private String userID;
private String password;

public void toAdmin(Message message, SessionID sessionId) {
    if (isMessageOfType(message, MsgType.LOGON)) {
        addLogonField(message);
    }
    super.toAdmin(message, sessionId);
}

private void addLogonField(Message message) {
    message.getHeader().setField(new SenderSubID(userID));
    message.getHeader().setField(new RawDataLength(password.length()));
    message.getHeader().setField(new RawData(password));
}

private boolean isMessageOfType(Message message, String type) {
    try {
        return type.equals(message.getHeader().getField(new
MsgType()).getValue());
    } catch (FieldNotFound e) {
        logErrorToSessionLog(message, e);
        return false;
    }
}

private void logErrorToSessionLog(Message message, FieldNotFound e) {
    LogUtil.logThrowable(MessageUtils.getSessionID(message),
e.getMessage(), e);
}
```