# [**Topcoder - Build processor to update InformixDB**](http://apps.topcoder.com/forums/?module=Category&categoryID=48778)

**Verification Guide**

# 

# Setup

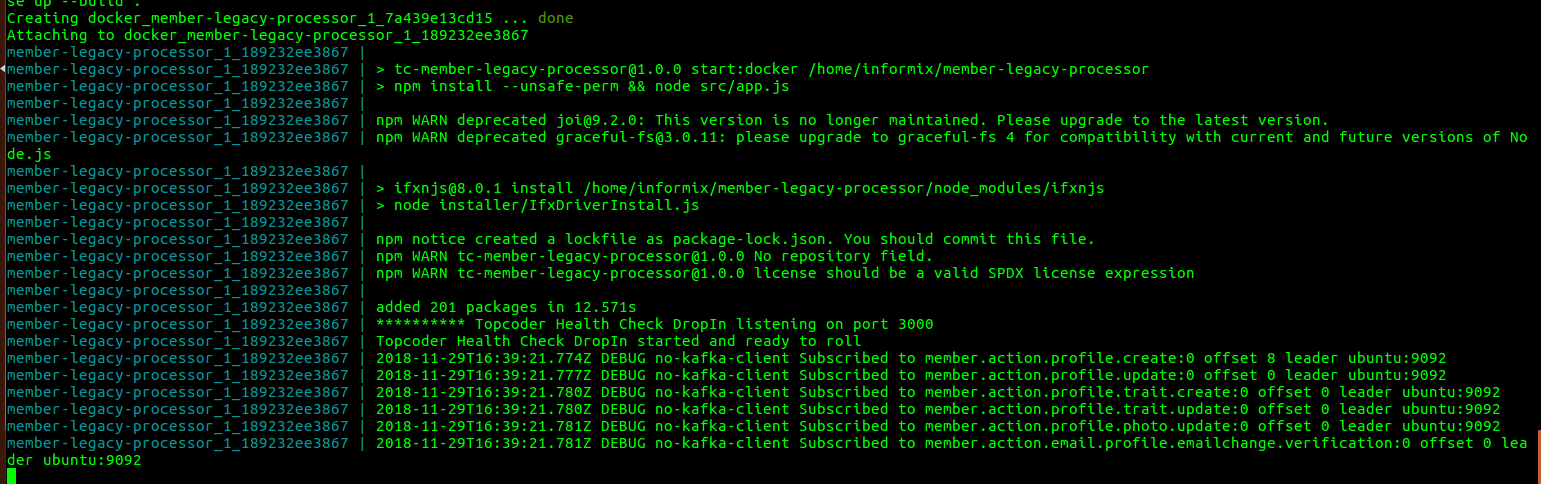
Follow instructions in member-legacy-processor/README.md to deploy the processor, basically the following steps are required for the processor deployment :

– Start and configure Kafka with the needed topics.

– Start Informix and make sure that the ‘sequence\_address\_seq’ sequence is created ( see Informix setup section in README.md)

– Start the processor.

Once the processor is started, the output looks like the following :



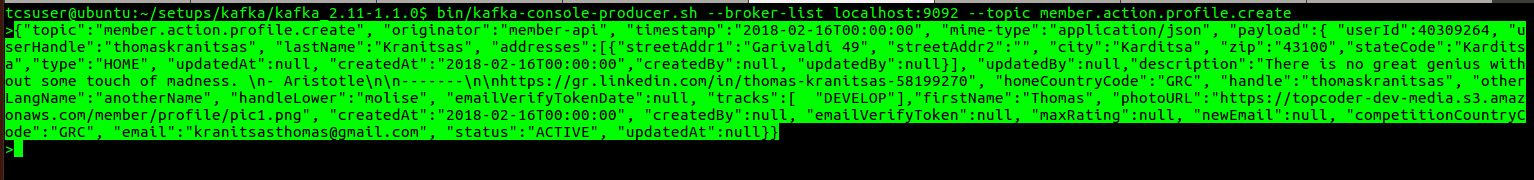
# Verification

Once the processor is started, create a Kafka producer for the ‘member.action.profile.create’ topic by running the following command (See section “Local Kafka Setup” in README.md):

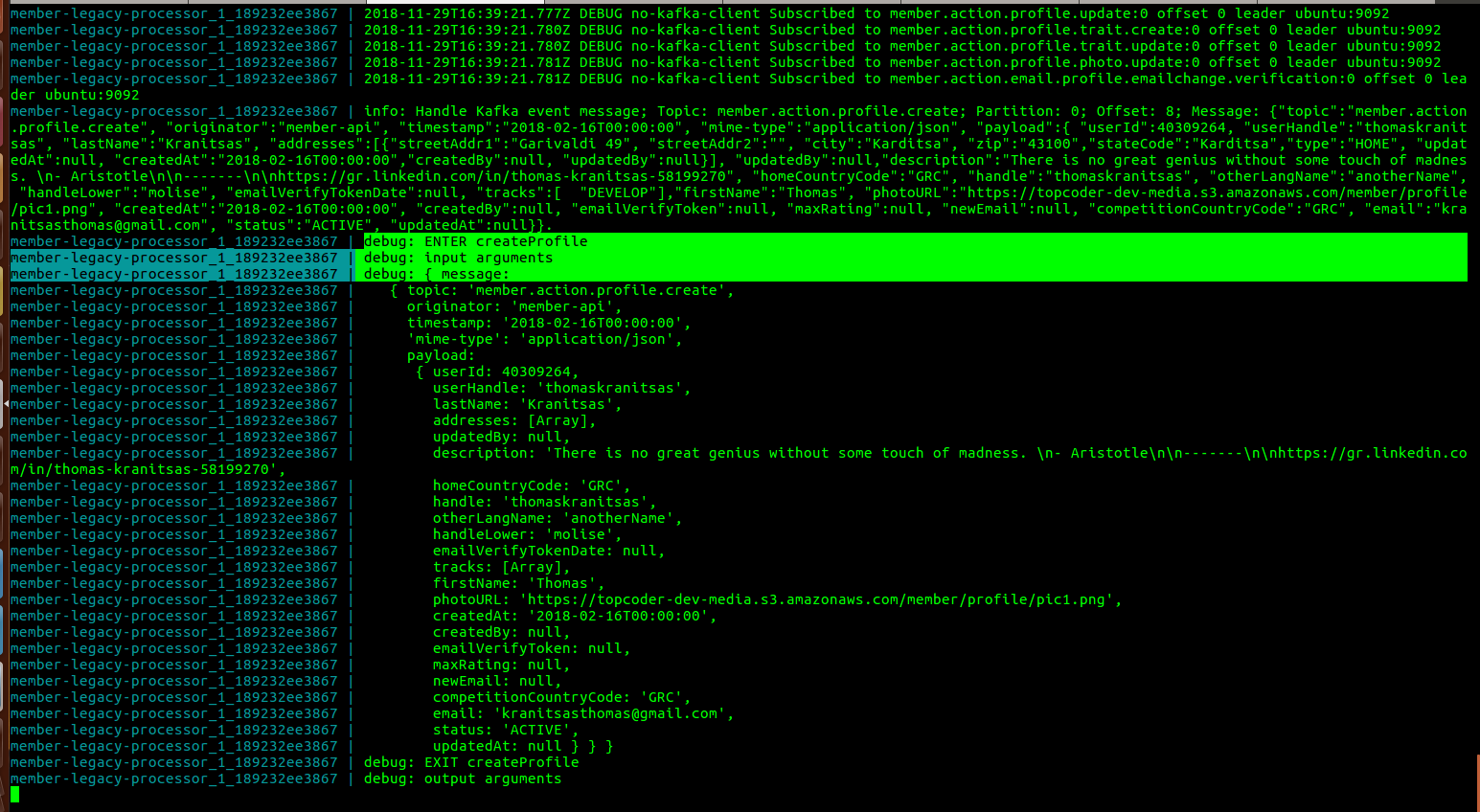
*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic member.action.profile.create*

A sample message to be used for testing the profile creation is provided within the submission under member-legacy-processor/docs/sample-messages/create-profile.json

Copy the content of the create-profile.json and paste it in the producer console and press Enter, as shown in this screenshot :



After sending the message in the producer, the processor will handle it and the following logs will be shown on the processor logs terminal :

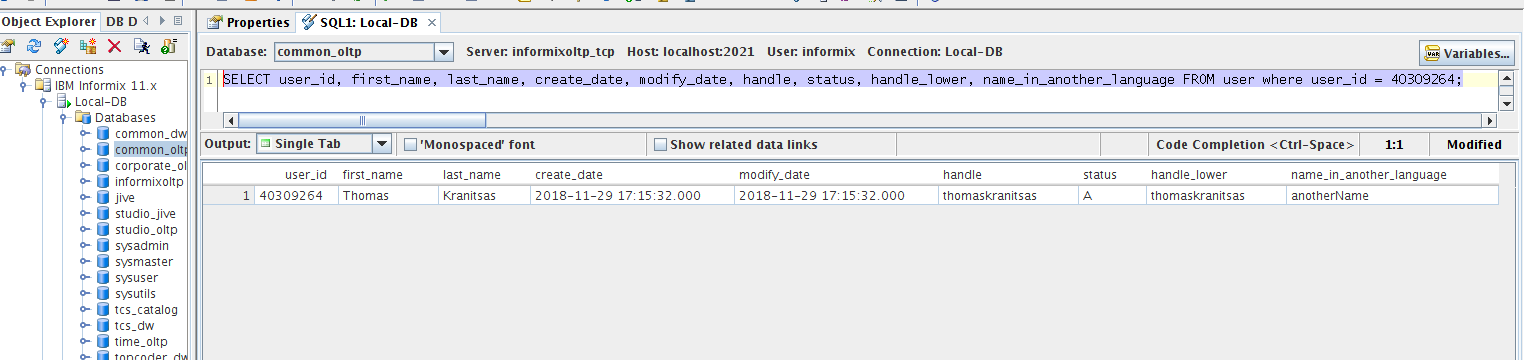


Connect to Informix common\_oltp database with credentials ‘informix/1nf0rm1x’ and verify the inserted data using the following SQL statements :

– select *user\_id, first\_name, last\_name, create\_date, modify\_date, handle, status, handle\_lower, name\_in\_another\_language FROM user where user\_id = 40309264;*

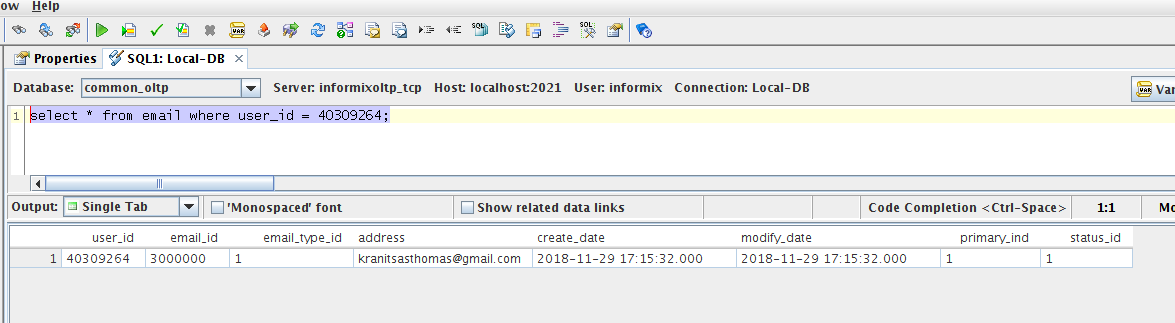
The user\_id value in the above statement comes from the message sent in the kafka producer, if you used a different user\_id, then make sure to update the statement accordingly.

The result of the above statement is shown on this screenshot :



– *select \* from email where user\_id = 40309264;*

The result of this statement is shown on the screenshot below :



– *select addr.address\_id, addr\_type.address\_type\_desc,*

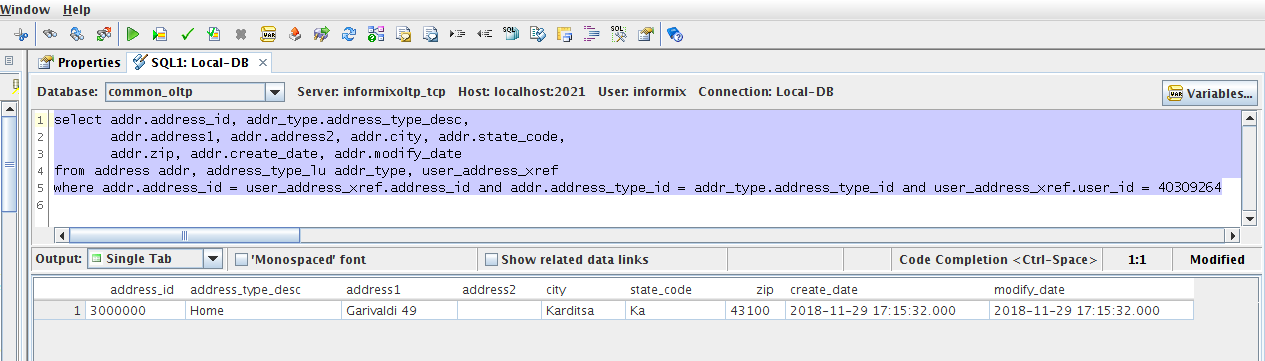
*addr.address1, addr.address2, addr.city, addr.state\_code,*

*addr.zip, addr.create\_date, addr.modify\_date*

*from address addr, address\_type\_lu addr\_type, user\_address\_xref*

*where addr.address\_id = user\_address\_xref.address\_id and addr.address\_type\_id = addr\_type.address\_type\_id and user\_address\_xref.user\_id = 40309264;*

The result of this statement is shown on the screenshot below :

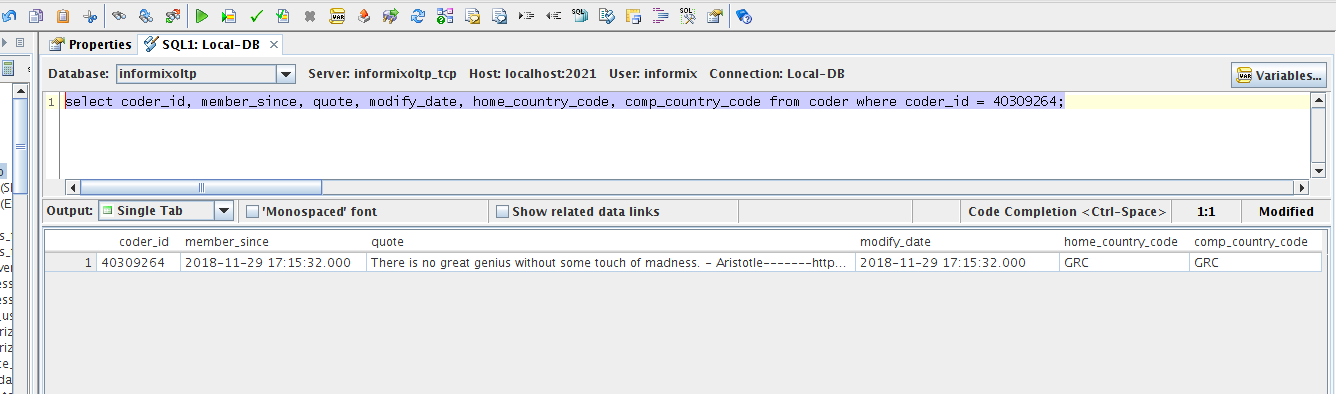


Now let’s check of the coder profile data is correctly inserted in Informix.

Connect to informixoltp database and execute the following SQL statements :

– *select coder\_id, member\_since, quote, modify\_date, home\_country\_code, comp\_country\_code from coder where coder\_id = 40309264;*

The result of this statement is shown below :

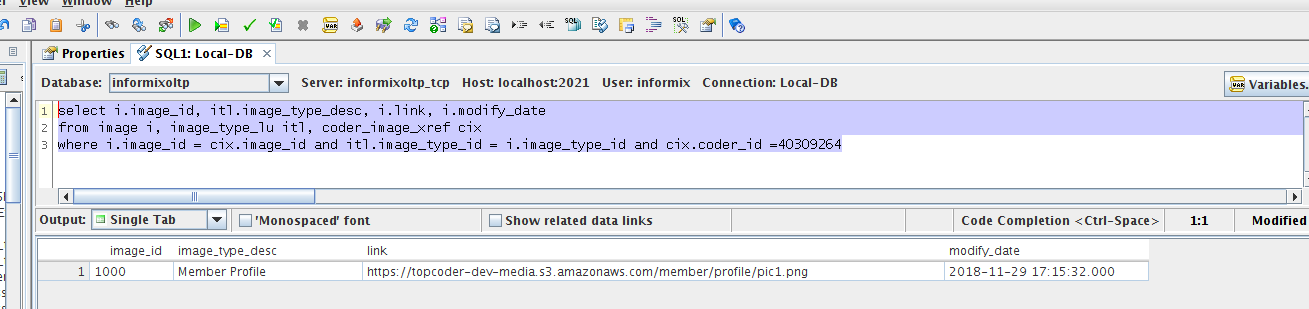


– *select i.image\_id, itl.image\_type\_desc, i.link, i.modify\_date*

*from image i, image\_type\_lu itl, coder\_image\_xref cix*

*where i.image\_id = cix.image\_id and itl.image\_type\_id = i.image\_type\_id and cix.coder\_id =40309264;*

The result of this statement is shown below :

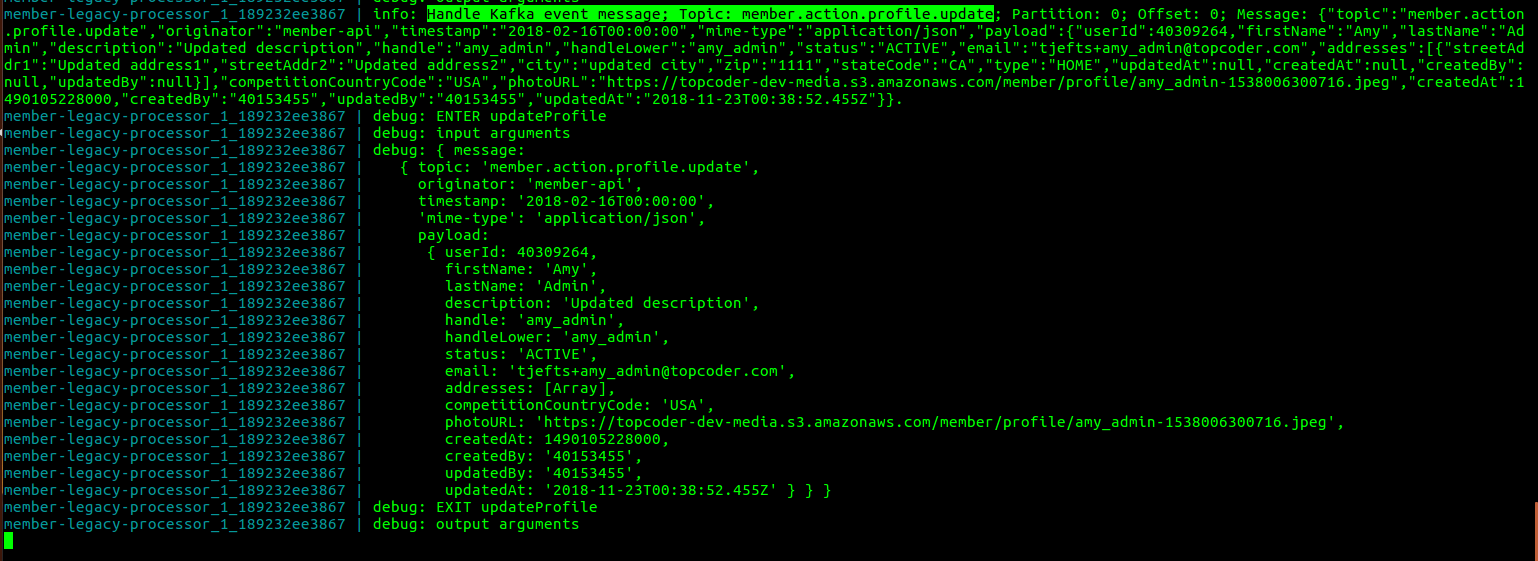


For testing the update profile messages, create a Kafka producer for update profile topic using the following command :

*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic member.action.profile.update*

A sample update profile message is provided under member-legacy-processor/docs/sample-messages/update-profile.json

After sending the message in the created producer, the processor will show the following logs :



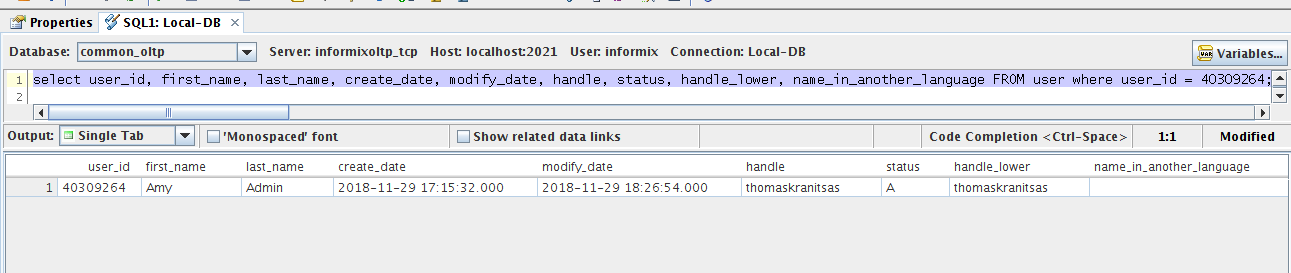
Re-check the database using the same SQL statements used for checking the profile creation.

The results of the checks are shown below for each SQL statement :

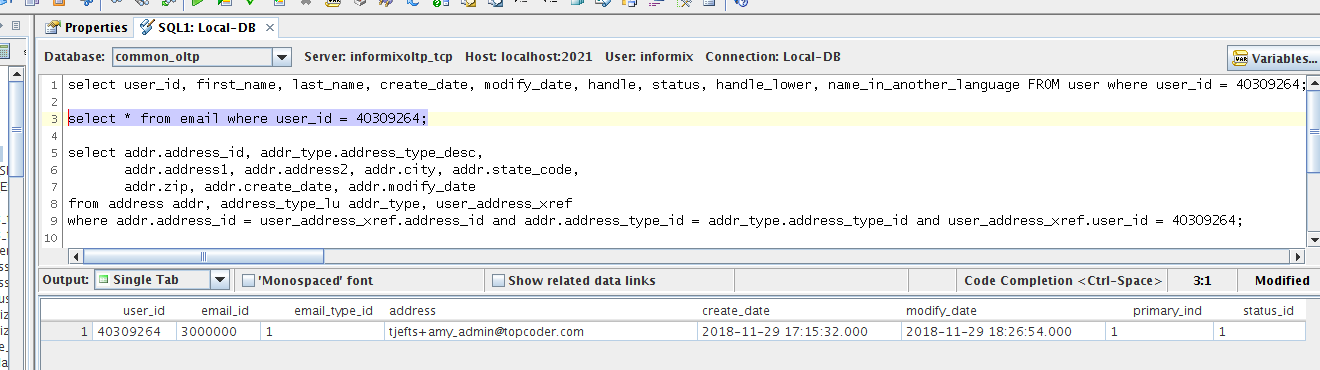
In common\_oltp database -

– *select user\_id, first\_name, last\_name, create\_date, modify\_date, handle, status, handle\_lower, name\_in\_another\_language FROM user where user\_id = 40309264;*

(notice the handle is not changed, this is not allowed by Topcoder policy)



– *select \* from email where user\_id = 40309264;*



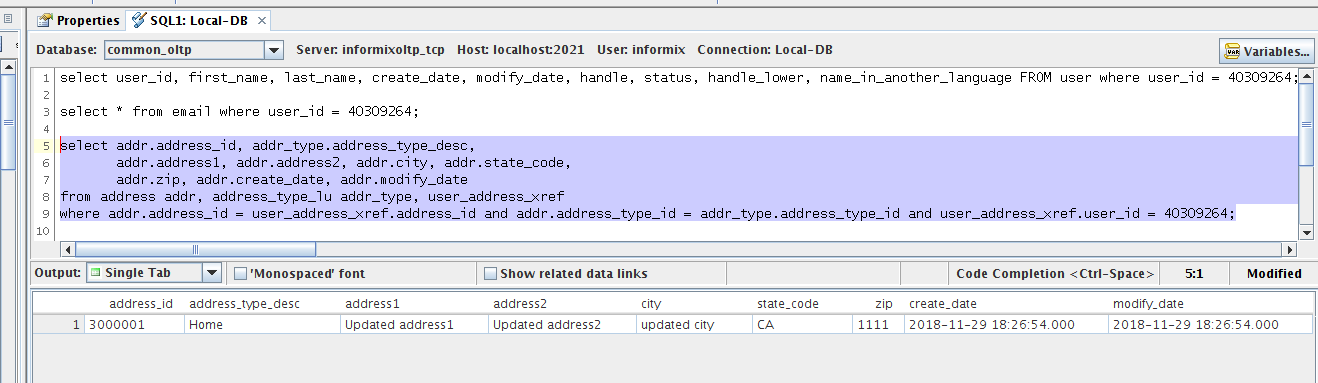
– *select addr.address\_id, addr\_type.address\_type\_desc,*

*addr.address1, addr.address2, addr.city, addr.state\_code,*

*addr.zip, addr.create\_date, addr.modify\_date*

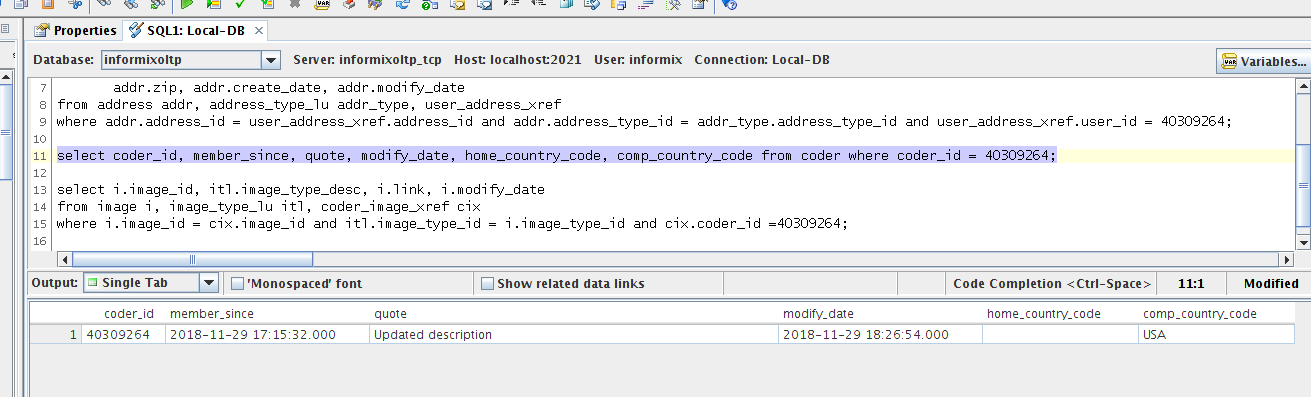
*from address addr, address\_type\_lu addr\_type, user\_address\_xref*

*where addr.address\_id = user\_address\_xref.address\_id and addr.address\_type\_id = addr\_type.address\_type\_id and user\_address\_xref.user\_id = 40309264;*



In informixoltp database :

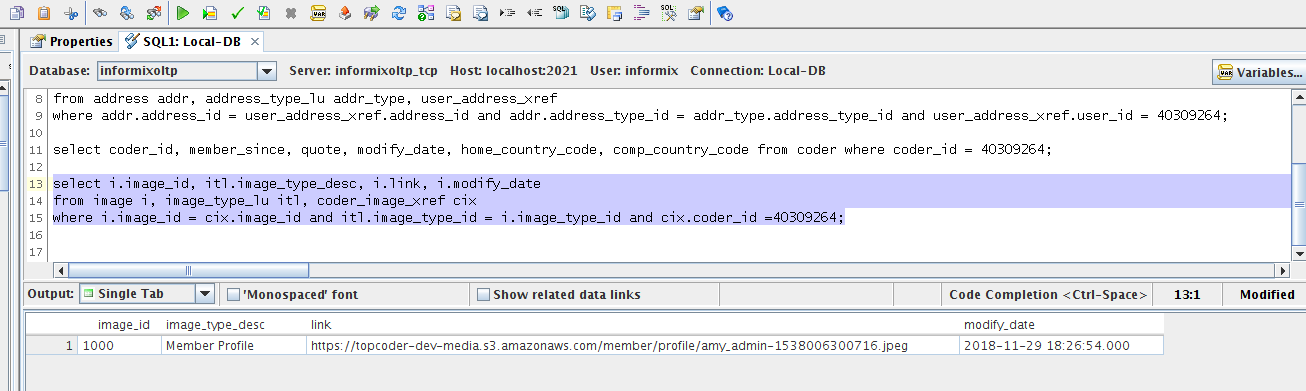
– *select coder\_id, member\_since, quote, modify\_date, home\_country\_code, comp\_country\_code from coder where coder\_id = 40309264;*



– *select i.image\_id, itl.image\_type\_desc, i.link, i.modify\_date*

*from image i, image\_type\_lu itl, coder\_image\_xref cix*

*where i.image\_id = cix.image\_id and itl.image\_type\_id = i.image\_type\_id and cix.coder\_id =40309264;*

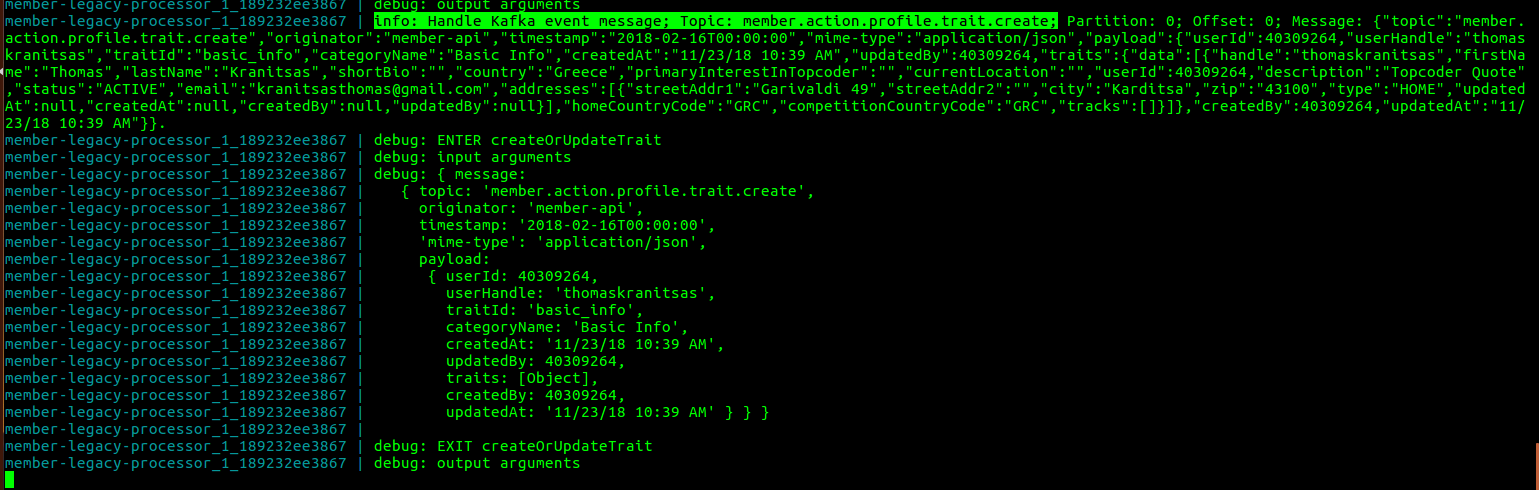


Create a Kafka producer for create trait topic with following command :

*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic member.action.profile.trait.create*

use docs/sample-messages/create-trait.json content as a message for testing.

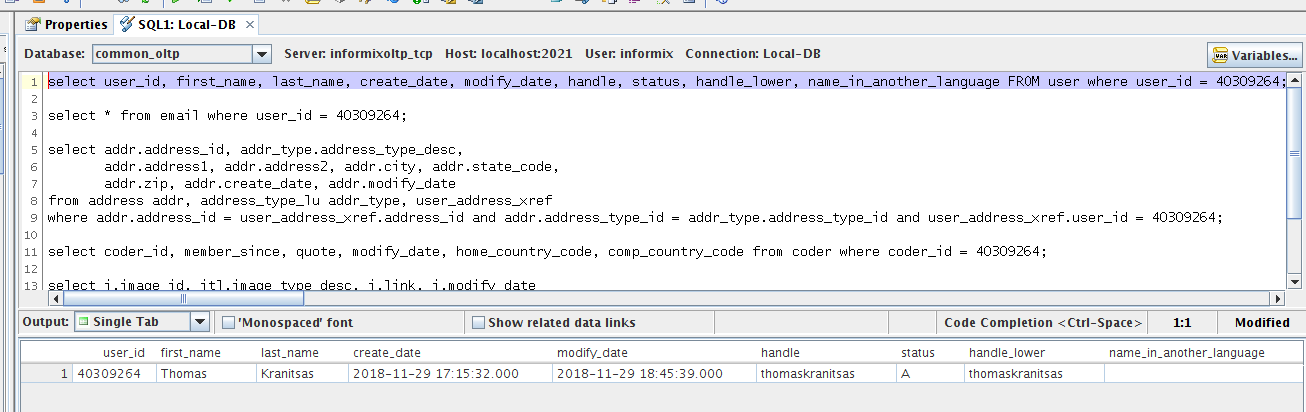
The processor logs will show the following :



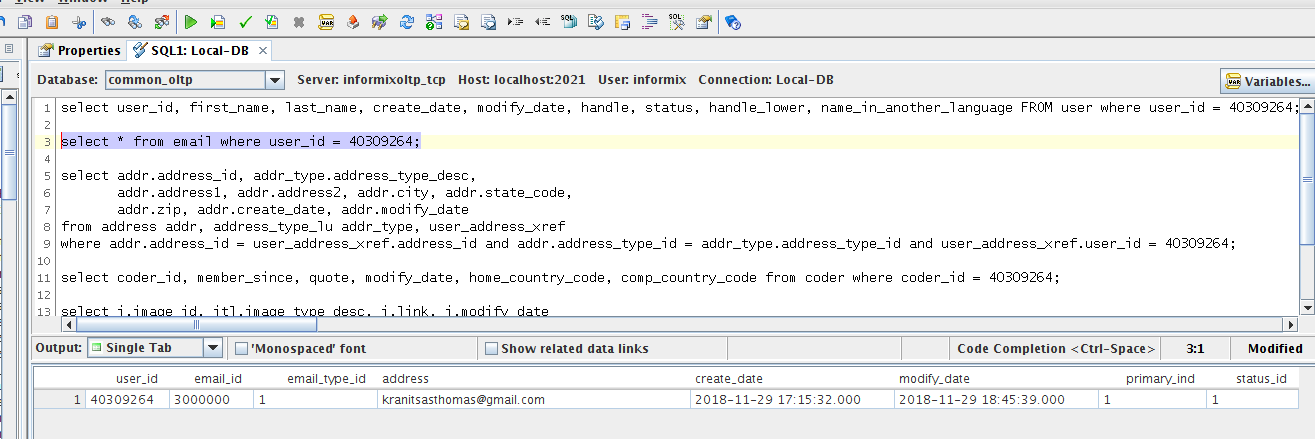
Double check the database using the same SQL statements used for profile creation/update and verify if the data is properly updated using the message content values.

In common\_oltp database :

– *select user\_id, first\_name, last\_name, create\_date, modify\_date, handle, status, handle\_lower, name\_in\_another\_language FROM user where user\_id = 40309264;*



– *select \* from email where user\_id = 40309264;*



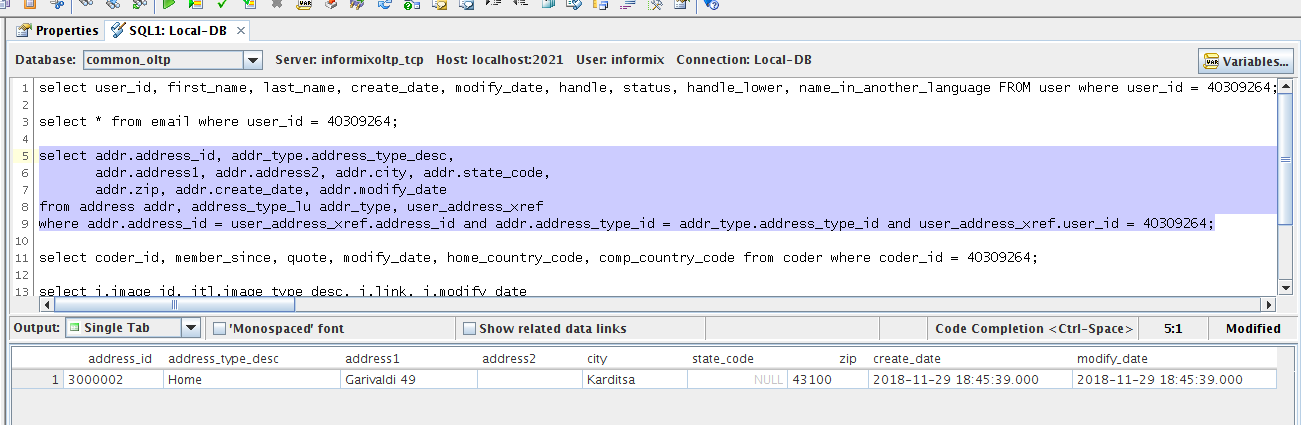
– *select addr.address\_id, addr\_type.address\_type\_desc,*

*addr.address1, addr.address2, addr.city, addr.state\_code,*

*addr.zip, addr.create\_date, addr.modify\_date*

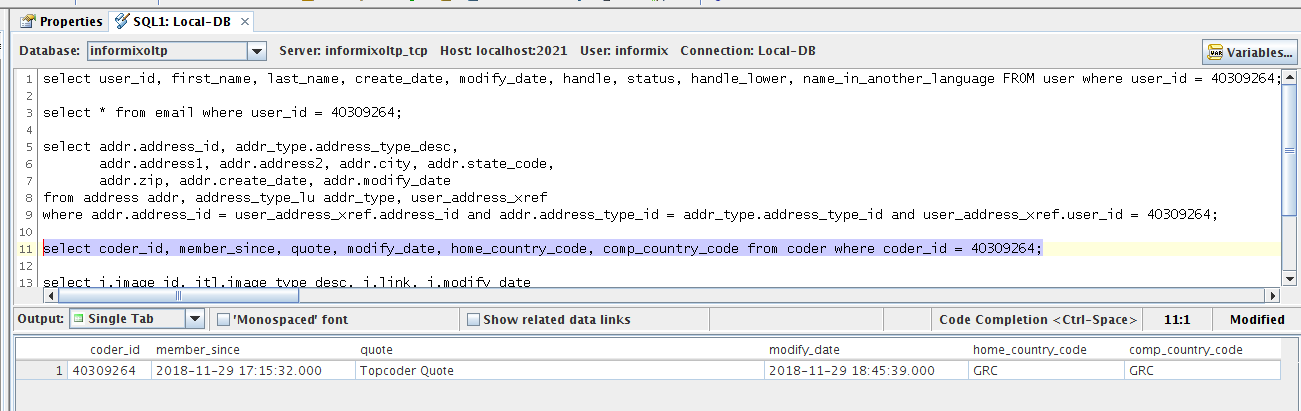
*from address addr, address\_type\_lu addr\_type, user\_address\_xref*

*where addr.address\_id = user\_address\_xref.address\_id and addr.address\_type\_id = addr\_type.address\_type\_id and user\_address\_xref.user\_id = 40309264;*



In informixoltp database :

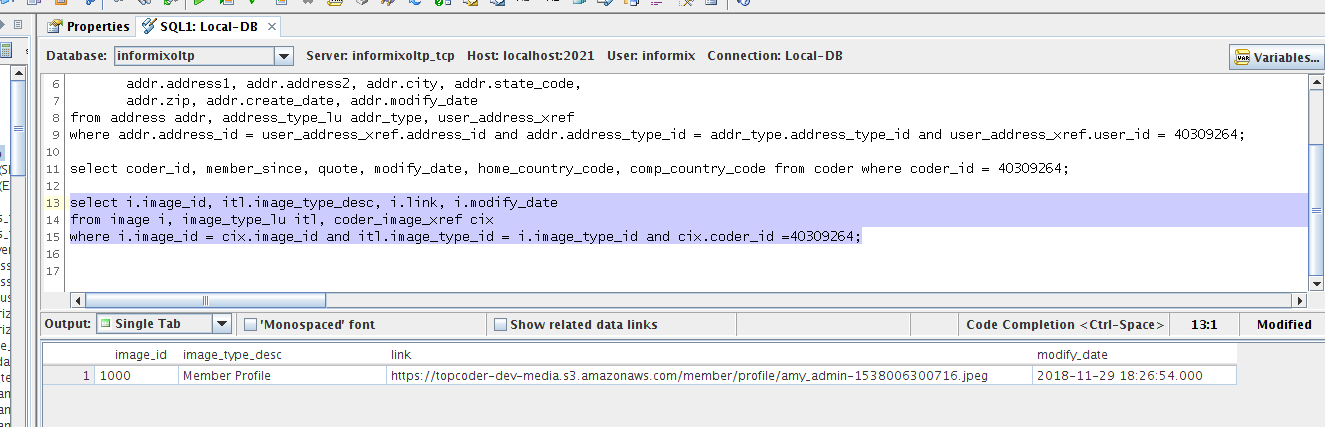
– *select coder\_id, member\_since, quote, modify\_date, home\_country\_code, comp\_country\_code from coder where coder\_id = 40309264;*



– *select i.image\_id, itl.image\_type\_desc, i.link, i.modify\_date*

*from image i, image\_type\_lu itl, coder\_image\_xref cix*

*where i.image\_id = cix.image\_id and itl.image\_type\_id = i.image\_type\_id and cix.coder\_id =40309264;*



Update trait can be tested in a similar manner as above using the corresponding message in the sample messages folder.

For email change verification, create the Kafka producer using the following command :

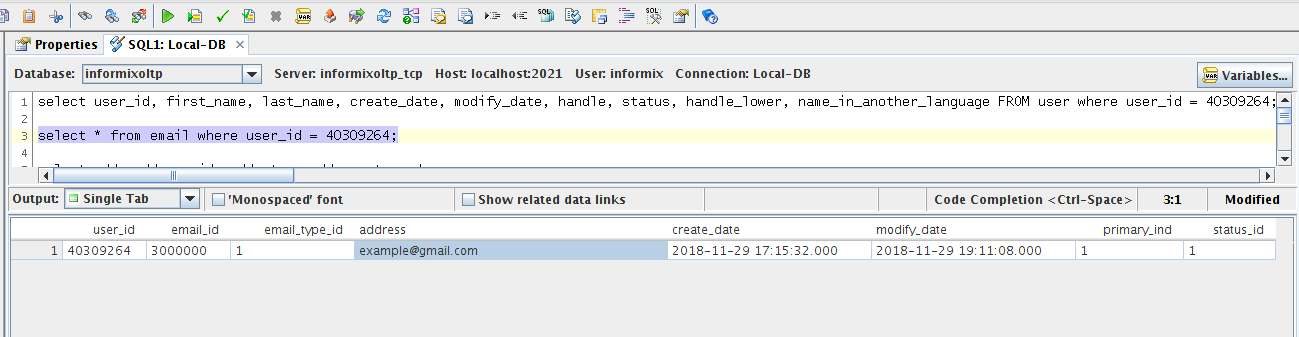
*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic member.action.email.profile.emailchange.verification*

and send the corresponding message for testing.

After the message processing, connect to informixoltp database and execute the following statement :

– *select \* from email where user\_id = 40309264;*

The result of the statement is shown in this screenshot (the email is changed and set to the one sent in the message ):



For photo update verification, create the Kafka producer using the following command :

*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic member.action.profile.photo.update*

use the content of the update-photo.json as a message to sent in the producer.

After the processing, connect to informixoltp database and execute the following SQL statement :

– *select i.image\_id, itl.image\_type\_desc, i.link, i.modify\_date*

*from image i, image\_type\_lu itl, coder\_image\_xref cix*

*where i.image\_id = cix.image\_id and itl.image\_type\_id = i.image\_type\_id and cix.coder\_id =40309264;*

The result is shown in this screenshot (note the updated link column) :

