



Technical New Design
Episys Message Bus Handler
19.004.02

Last Updated: 2/8/21 8:28 AM

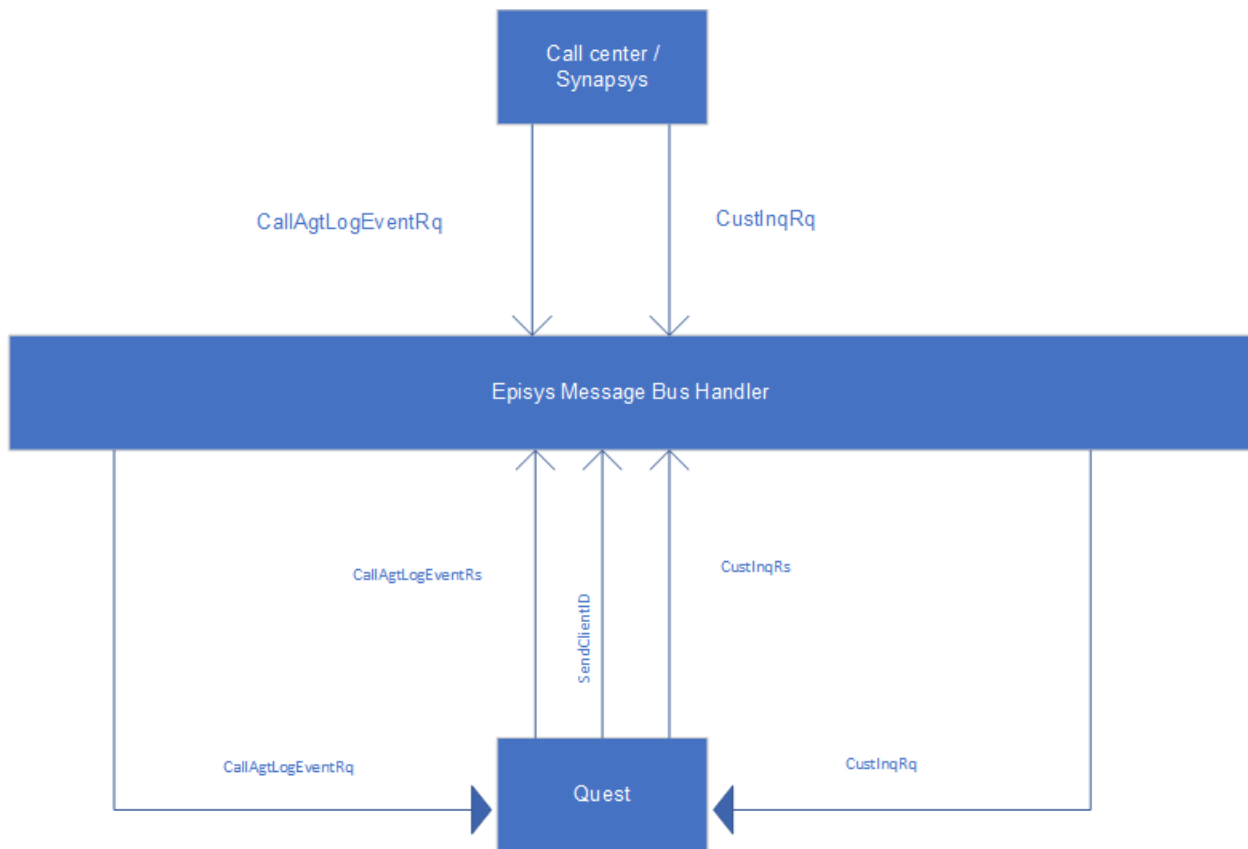
Table of Contents

1 Project Summary	3
1.1 Current Behavior.....	3
1.1.1 CallAgtLogEventRq_XPType.....	4
1.1.2 CustInqRq_XPType	5
1.1.3 SendClientId.....	5
1.1.4 Sequence Diagram	6
1.2 New Behavior	6
1.2.1 Message Queue	6
1.2.2 CallAgtLogEventRq_XPType.....	6
1.2.2.1 Connect	6
1.2.2.2 Disconnect	7
1.2.3 CustInqRq_XPType	7
1.2.3.1 Scenario 1.....	7
1.2.3.2 Scenario 2.....	8
1.3 Storage Map and ClientID	9
1.4 Work Area	9

1 Project Summary

EpisysMessagehandler is used to handle all the messages to and from all Quest instances in Xperience. Mainly it handles three messages, the first two messages are published by Call center and the last message is published by Quest.

- CallAgtLogEventRq_XPType
- CustInqRq_XPType
- SendClientID



1.1 Current Behavior

EpisysMessagehandler maintains “Persistent Storage” map in the form of dictionary, which saves the information of partition, SYM and Quest version available in the SYM. On successful connection, the connection information (Partition, SYM and Quest version) will be written into storage map and the same information will be read when “CallAgtLogEvent” message triggered from Call center to Quest. These two messages normally come one after another within 20-30 seconds time frame and CustInq message enters loop and will wait for 2 ½ minutes to finish CallAgtLogEvent message.

1.1.1 CallAgtLogEventRq_XPType

EpisysMessageBusHandler verifies the existing persistent storage map,

- ➔ If the hostId, corresponding sym numbers present then it takes the Quest version.
- ➔ Verifies if any of the instance of that Quest is already opened,
 - If NOT, check whether the Quest is installed in machine,
 - If Quest version is installed, launches an instance.
 - If Quest version is not installed, launches the highest Quest version from Quest Map. (It won't navigate any work area further. Because the result of host and sym, client version is different than CustInq message expected host and sym)

Once host is connected successfully, Quest sends CallAgtLogEventRs_XpType response to EpisysMessageBusHandler with "Success+ClientID".

Connect to host example:

```
XperienceContext.Current.MessageBus.Connect();

var xPHdrCType = new XPHdr_CType
{
    AuditUsrId = "episysuser",
    InstRtId = "021320171", //Inst num
    InstEnv = "232"
};

var xPMsgRqHdrCType = new XPMsgRqHdr_CType
{
    XPHdr = xPHdrCType
};

CallAgtLogEventRq_XPType request = new CallAgtLogEventRq_XPType()
{
    HostId = "sdcaqas8p37",
    XPMsgRqHdr = xPMsgRqHdrCType,
    LogEventType = eventType // "CONNECT"/"DISCONNECT"
};

MessageOptions msgOptions = new MessageOptions();
msgOptions.DestinationIds.Add("Episys Quest");

var res =
XperienceContext.Current.MessageBus.Send<CallAgtLogEventRq_XPType>(request,
msgOptions).WaitForResponse(TimeSpan.FromSeconds(40));
```

1.1.2 CustInqRq_XPType

Prior to routing the message, EpisysMessageBusHandler ensures that we've established a successful connection to the correct host and SYM (Quest sends "Success + ClientID" on successful connection).

- ➔ If the connection has been established, it route the message.
- ➔ If the connection has NOT been established, it won't route the message.

CustInq route message example:

```
var XPHdr = new XPHdr_CType()
{
    InstEnv = "232",
    InstRtId = "021320171",
    ConsumerProd = "sdcaqas8p37"
};

var XPMsgRqHdr = new XPMsgRqHdr_CType()
{
    XPHdr = XPHdr
};

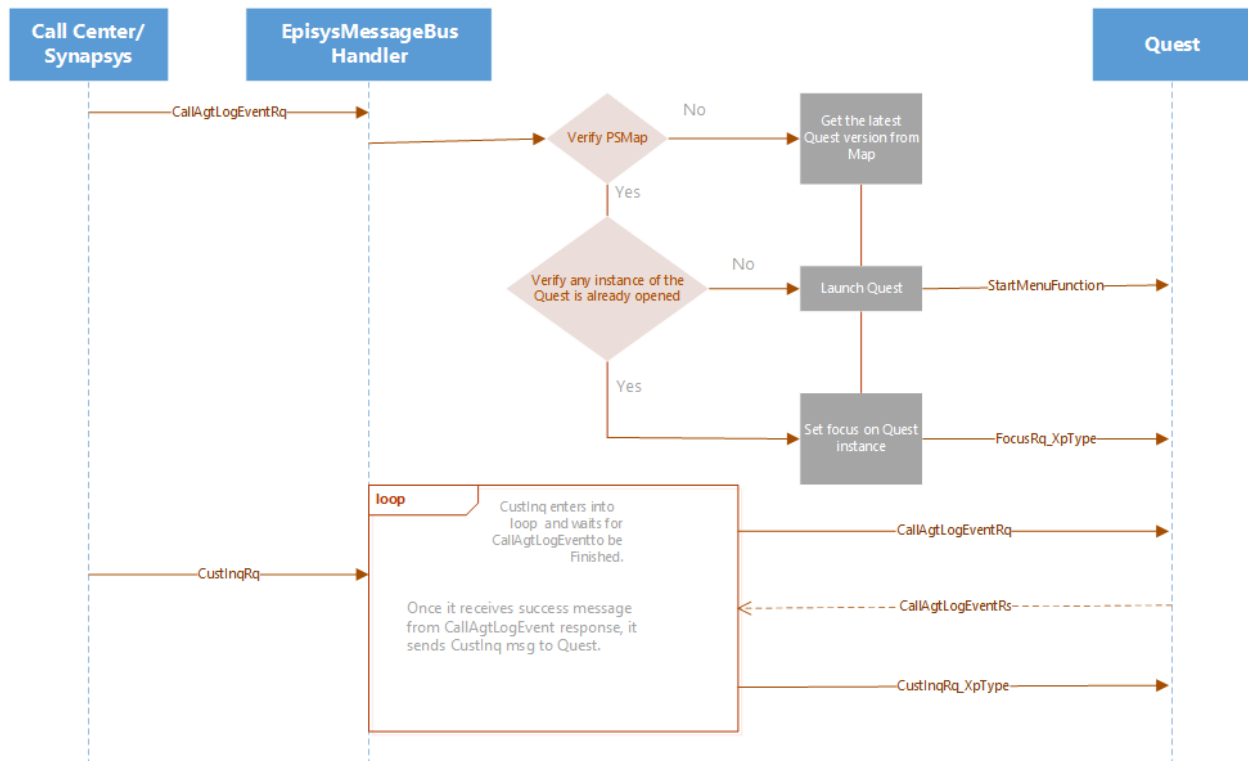
CustInqRq_XPType request = new CustInqRq_XPType()
{
    CustId = "0000000265",
    XPMsgRqHdr = XPMsgRqHdr,
    ActContextElem = workArea //"AccountManager", "MemberManager", "TellerTran"
};

MessageOptions msgOptions = new MessageOptions();
msgOptions.DestinationIds.Add("Episys Quest");
var res = XperienceContext.Current.MessageBus.Send(request, msgOptions);
```

1.1.3 SendClientId

This message will be published by Quest. This is call back function which will be fired back when new instance of Quest is created or closed. Quest publishes this message with ClientID. This ClientID will be saved in "questInstanceMap".

1.1.4 Sequence Diagram



1.2 New Behavior

EpisysMessageBusHandler mainly supports two types of messages and these two messages will be standalone.

- ➔ CallAgtLogEventRq_XPType
- ➔ CustInqRq_XPType

1.2.1 Message Queue

If CallAgtLogEvent and CustInq message comes one after another immediately, the second message will be saved in message queue. Once get the first message response back, the second message will be implemented.

1.2.2 CallAgtLogEventRq_XPType

1.2.2.1 Connect

EpisysMessageBusHandler will handle all the messages which Quest is sending back while connecting/disconnecting the host.

- Success (Implemented in existing behavior)
- Failed due to another Quest in Sym
- Host is already connected
- Failed to Connect to host

1.2.2.2 Disconnect

EpisysMessageBusHandler gets the ClientID from persistent storage map based on host and sym number. It uses the same ClientID to disconnect the host.

Note: User can disconnect the host using CallAgtLogEvent message, even if Quest launches manually.

1.2.3 CustInqRq_XPType

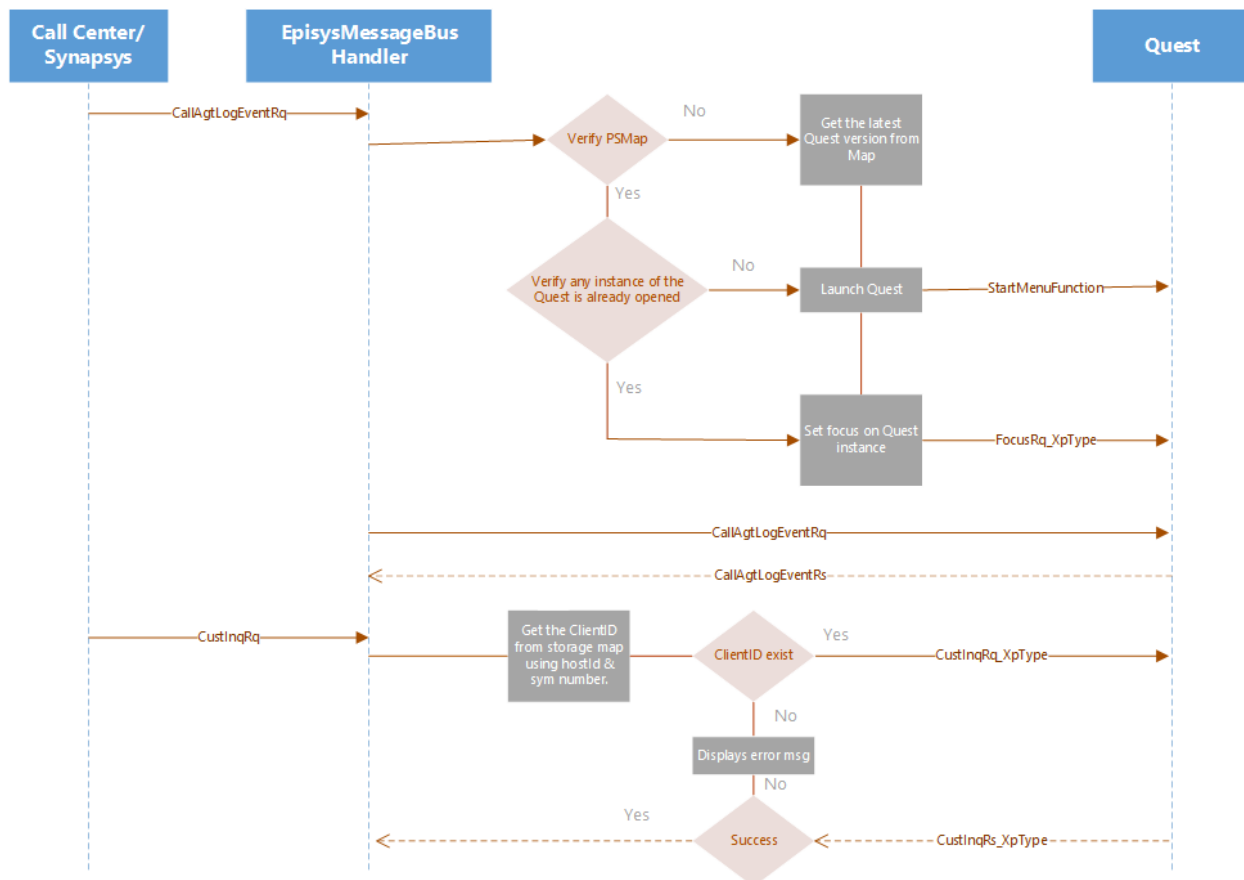
The CustInq message will work independently.

- ➔ Using hostId and sym number it gets the ClientID from storage map. It uses the same ClientID to send CustInq message to Quest.
- ➔ Verifies the ClientID from storage map
 - If ClientID exist, route the CustInq and return the CustInqRs_XpType response back. If response has "Fail" message, displays the error message.
 - If the ClientID doesn't not found in the storage map, below are two scenarios, will implement any one. (Note: In team meeting will discuss and decide the best scenario)

1.2.3.1 Scenario 1

- ➔ If the ClientID is not found in the storage map, EpisysMessageBusHandler display the error message.

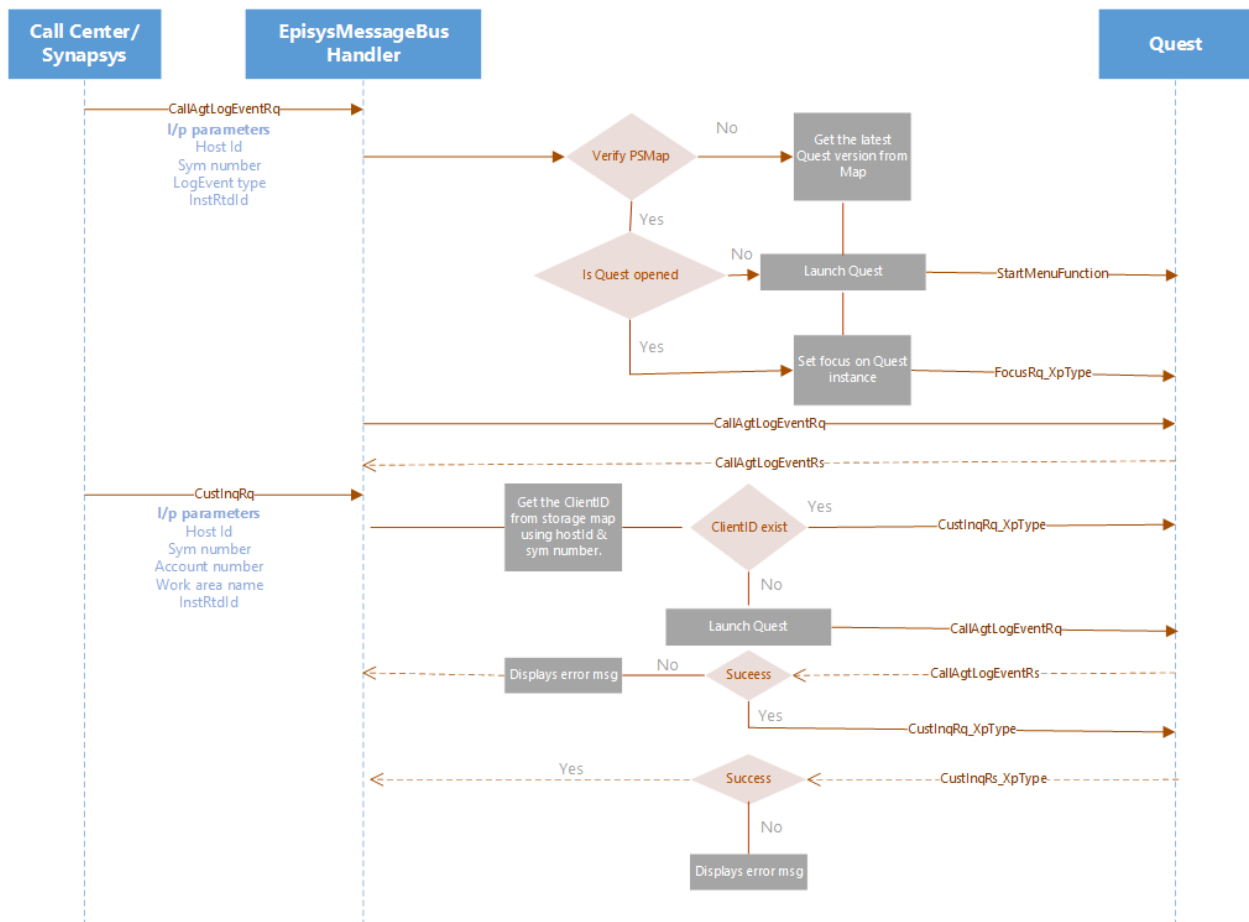
1.2.3.1.1 Sequence diagram 1



1.2.3.2 Scenario 2

- ➔ If the ClientID is not found in the storage map,
 - Check whether the Quest is installed in machine,
 - If Quest version is installed, launch an instance, sends CallAgtLogEvent message and Quest sends response back.
 - If EpisysMessageBusHandler get the success response, route the CustInq message.
 - If not, display error message
 - If Quest version is not found, display appropriate message.

1.2.3.2.1 Sequence diagram 2



1.3 Storage Map and ClientID

EpisysMessageBusHandler maintains three storage maps (dictionaries).

→ Persistent Storage Map

It contains hostId, Sym number and Quest version.

Based on host and sym number, persistent storage map returns the quest version.

→ Quest Instance Map

Data comes from each new instance of Quest is started/closed on Xperience.

It maintains ClientID and Quest version as key, values .

Quest instance map will return ClientID based on Quest version returned by persistent storage map.

→ Quest Map

Data comes from all the msi installed quest products on the machine.

It contains Quest version and path of the SFW.exe.

1.4 Work Area

EpisysMessageBusHandler supports below work areas.

Tellertran,
AccountManager,
ApplicationProcessing,
CollectionPackage.