# Troubleshooting External Systems Outbound (ESO)

Troubleshooting interfaces requires a methodical approach, knowledge of interface messaging, and a good understanding of the Cerner architecture. While no document can anticipate every issue, this document provides you with the basic methods to investigate some of the common problems encountered. It details some specific strategies and tools for troubleshooting outbound transactions. It also provides additional architecture information, which assists in the troubleshooting process.

System Integration Manager (SIManager.exe) now provides a module for troubleshooting outbound interfaces. This module provides the functionality for you to perform the folloing tasks:

- See a high-level view of the activity in the ESO queues (select Display Current CQM State).
- Search for a transaction from any of the ESO queues. From the CQM_FSIESO_QUE, you can search using very basic information, such as person, message type, ID, and so on.
- Export search results to a .CSV file to send to someone to assist with troubleshooting an issue.
- View a transaction at multiple points in the process. You can peek from the CQM_FSIESO_QUE, CQM_OENINTERFACE_QUE, and OEN_TXLOG.
- View the details from the FSIESO_QUE_DETAILS table associated with a particular transaction (View Details).
- See a visual trace of a transaction through the outbound process. This can be accessed by right-clicking on search results from any queue and selecting Overview. From the Overview, you can double-click on a queue and see the queue row information at the bottom of the screen.
- Use various reports for tracing a transaction (Queue Trace report), performance reports, daily audit reports, unprocessed transaction reports, trigger reports to understand the triggers that are on in a system, skipped message reports, and so on. You can send, save, and print these reports to assist with auditing or troubleshooting.
- View CMB_TEMP, script files, and so on from the front end without having to access the back end.
- View an interface tablespace report to see if the interface tables are getting full, reaching the maximum number of extents, or both.
- Replay a transaction through the queues to try and process again (right-click a queue row and select Replay Transaction).
- View the GET_*_SELECTION scripts that are running in production. You also can view the script template shipped with the current package you are running.
- View TCP/IP information and troubleshoot a TCP/IP connection.

# Overview of the Troubleshooting Process

1. Identify the problem.
2. Perform troubleshooting steps using the ESO Troubleshooting module in SI Manager and other available tools.
3. Resolve the problem or log a service request.

## Identify the Problem

You have just received an error. You should ask the following questions:

- Can you reproduce it?
- Are you the only user that receives the error?
- Do you receive the error on multiple computers?
- Can you reproduce the error in another environment?
- What type of installation does the PC have: local, network, or Citrix?
- What type of error is it?

Types of errors include script errors, illegal operation errors, and information errors. If you receive a script error, note the script name. If the error contains multiple tabs, copy the data on each tab, or print a report from the error if the print option is available. This information could be useful if you need to log a service request.

If the transaction did not post in the foreign system, you should ask these additional questions:

- Where in the process did the breakdown occur--- *Cerner Millennium*, Open Engine, or the foreign system?
- If it did not make it out of *Cerner Millennium*, where did the process break down?
    - Did the trigger drop to CQM?
    - Did the trigger make it through all the ESO servers?
    - Did the trigger make it to the appropriate communication server (comserver)?

## Perform Troubleshooting Steps

You have determined the error is reproducible. To reproduce an error, you must use the exact keystrokes that were used when the error was initially received. Begin troubleshooting by asking the following questions:

- If you alter the steps slightly, do you still receive the error?
- Is this the intended use of the interface, application, or tool?
- Did you check Cerner Solutions?
- Are all servers running?

If you anticipate logging a service request, please provide a screen print of the errors and the steps that you have taken to troubleshoot the issue, as this aids in troubleshooting.

## Resolve or Log a Service Request

Most of the time you can resolve your own issues using the ESO Troubleshooting module in SI Manager; however, you may require assistance to achieve resolution.

## Understanding the Troubleshooting Process

Understanding the troubleshooting process is described below.

## SolutionWorks and IRC Support

If you are unable to resolve an issue, SolutionWorksand the IRC(Immediate Response Center) are available to assist you.

- SolutionWorksis staffed Monday through Friday from 0700 to 1900 (Central Standard Time). You can contact SolutionWorksat (866) 221-8877|#\.
- The IRCis staffed 24 hours a day, 7 days a week. The IRChandles critical system issues or noncritical issues that occur when SolutionWorksis not staffed. You can contact the IRCat (816) 221-8877|#\.

For noncritical issues, you also can log a service request through eService on Cerner.com.

## ESO Processing Flow

Figure 1. ESO Processing Flow

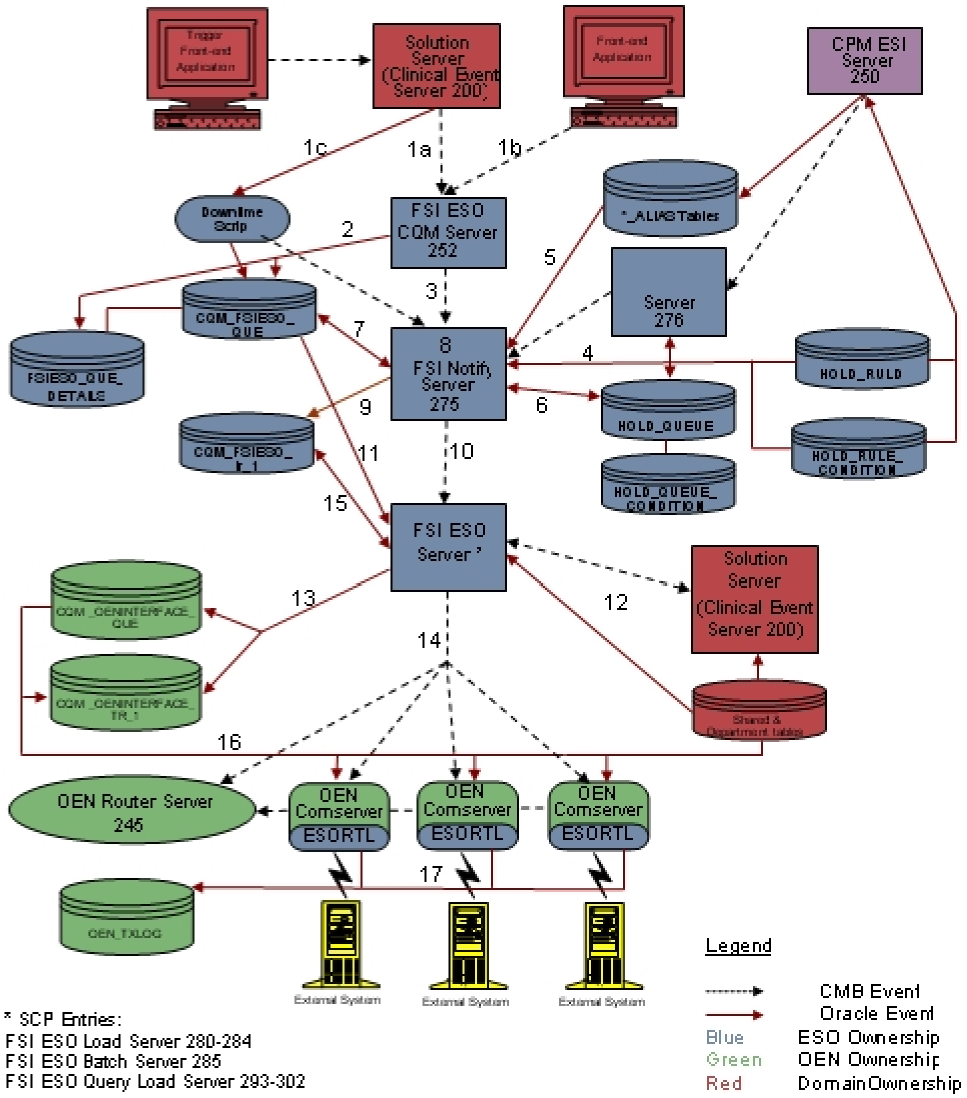


Figure 1, ESO Processing Flow

The numbered steps in the following flow chart correspond to the steps following the chart. These steps detail the ESO Processing Flow shown above.

1. The first step can occur one of three ways:
    1. An application sends a transaction to the solution server, and then the solution server sends the transaction to the FSI ESO CQM server (SCP 252) in the form of a CMB Insert Event (outbound trigger).
    2. The application sends the transaction directly to the Foreign System Interfaces ESO CQM server in the form of a CMB Insert Event.
    3. During Foreign System Interfaces ESO CQM server downtime, a CMB Notify event is sent to the Foreign System Interfaces Notify server (SCP 275) after writing the outbound trigger to the CQM_FSIESO_QUE table. This bypasses the Foreign System Interfaces ESO CQM server altogether while still authenticating the outbound trigger (Step 2). If the Foreign System Interfaces Notify server is running, it picks up the row in turn and process it. If the Foreign System Interfaces Notify server is down as well, when the Foreign System Interfaces Notify server starts up, it scans the CQM_FSIESO_QUE table for rows with a process status flag of 10 (Unprocessed Transaction). The Foreign System Interfaces Notify server recognizes the rows that were written while it was down and attempts to recreate the trigger.
2. The Foreign System Interfaces ESO CQM server writes the outbound trigger to the CQM_FSIESO_QUE table.

    Authentication of the outbound triggers is accomplished by selecting on the CQM_CONTRIBUTOR_CONFIG table using the *appname* and *contribalias* contained in the trigger. If there is no qualifying row, the outbound trigger is ignored.

    If a row was found on the CQM_CONTRIBUTOR_CONFIG table, then the authentication was successful and the outbound trigger is written to the CQM_FSIESO_QUE table. At the same time, the Foreign System Interfaces ESO CQM server extracts key information from the trigger and writes it to the FSIESO_QUE_DETAILS table. If an Oracle® error occurs while the Foreign System Interfaces ESO CQM server is writing to this or any table, it assumes its Oracle connection is faulty and cycles itself. The Foreign System Interfaces ESO CQM server authenticates the transaction and writes a row to the CQM_FSIESO_QUE table with a status flag of 10 (Unprocessed Transaction).

    To see these outbound triggers, set the Foreign System Interfaces ESO CQM server *dump_insert* property to one. Any outbound trigger sent to this server is printed to the .OUT or .ERR file in the cmb_temp directory. For more information on viewing the .OUT and .ERR files, see Server .OUT and .ERR Files.

    The Foreign System Interfaces ESO CQM server sends a CMB Notify event to the Foreign System Interfaces Notify server.

3. If the Foreign System Interfaces ESO CQM server is sending these CMB Notify events faster than the Foreign System Interfaces Notify server can process them, it begins queuing the CMB Notify events in a memory-based queue called the thread queue. If this queue becomes full, the Foreign System Interfaces Notify server dynamically switches from driving off the CMB Notify events to driving off the CQM_FSIESO_QUE table. That is, the Foreign System Interfaces Notify server switches back and forth as necessary between driving from the CMB Notify events and driving from the CQM_FSIESO_QUE table. There may be times when the Foreign System Interfaces Notify server is flooded with triggers, causing the thread queue to fill. When it reaches 90% of its capacity, it forces the Foreign System Interfaces Notify server to drive off the CQM_FSIESO_QUE table. When the activity slows down and the Foreign System Interfaces Notify server catches up, it switches back to driving off the CMB Notify events. Each server has a queue quota that is set on the server to determine when the thread queue is full. It is recommended that the queue quota for the Foreign System Interfaces Notify server not be set higher than 20,000 transactions.

> **Note**
>
> If Hold Transaction Processing is active, then Steps 4-7 are performed.

4. The Foreign System Interfaces Notify server checks to see if this type of outbound trigger qualifies to be put on hold. The Foreign System Interfaces Notify server obtains the hold conditions by selecting from the HOLD_RULE table using the class, type, and subtype from the trigger and joining to the HOLD_RULE_CONDITION table using the HOLD_RULE_ID. These are the conditions under which to hold this trigger (for example, order alias or encounter alias). These conditions must be met before the outbound trigger is released to go outbound.
5. The Foreign System Interfaces Notify server checks the respective alias tables to see if the hold conditions have been met. If the Foreign System Interfaces Notify server determined that this type of outbound trigger qualifies to be put on hold, it looks at the hold conditions from the HOLD_RULE_CONDITION table and checks the respective alias tables to see if the required aliases have been posted.
6. The Foreign System Interfaces Notify server writes a row to the HOLD_QUEUE table if the outbound trigger needs to be put on hold. If the Foreign System Interfaces Notify server finds that not all the hold conditions have been met, it puts the trigger on hold by writing outbound trigger information to the HOLD_QUEUE table. Additionally, the conditions for which it is on hold are written to the HOLD_QUEUE_CONDITION table.
7. If the outbound trigger needs to be put on hold, the Foreign System Interfaces Notify server sets the process status flag to 50 (Transaction On Hold) and sets the priority to 99 on the CQM_FSIESO_QUE table. (A priority of 99 means that the transaction already has been checked to see if needs to go on hold. A priority of 100 means that the transaction still needs to be checked.)

> **Note**
>
> If the outbound trigger has been put on hold, stop here. The Foreign System Interfaces Hold Release server (SCP 276) does all the releasing. Once the outbound trigger has been released, it picks up where it left off with Step 8.

8. The Foreign System Interfaces Notify server determines which Foreign System Interfaces ESO server to send a CMB Notify event. This also is partly determined by the Foreign System Interfaces CQM server. The Foreign System Interfaces CQM server fills out the SUBTYPE_DETAIL with a value of 1 to 5 for the Foreign System Interfaces ESO Load servers---1 for the Batch server, or 1 to 10 for the ESO Query Load servers. The table below shows in general terms how the Foreign System Interfaces Notify server determines to which Foreign System Interfaces ESO server the CMB Notify event is sent.

| **Server Name** | **Condition for Sending CMB Notify Event** | **SCP ID** | **Executable** |
|---|---|---|---|
| FSI ESO Server Load #1<br>FSI ESO Server Load #2<br>FSI ESO Server Load #3<br>FSI ESO Server Load #4<br>FSI ESO Server Load #5 | Unless the trigger is a PM Registry, PM Query, or batch transaction, one of these five servers receives the outbound trigger. If ESO Load Balancing is not installed, the trigger is routed to Foreign System Interfaces ESO Server Load #1 (SCP 280). If ESO Load Balancing is installed, see ESO Load Balancing or a detailed explanation of this process. | 280<br>281<br>282<br>283<br>284 | fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso |
| FSI ESO Server Batch #1 | This server receives the outbound trigger if it is a batch transaction. | 285 | fsi_srveso |
| FSI ESO Query Load #1<br>FSI ESO Query Load #2<br>FSI ESO Query Load #3<br>FSI ESO Query Load #4<br>FSI ESO Query Load #5<br>FSI ESO Query Load #6<br>FSI ESO Query Load #7<br>FSI ESO Query Load #8<br>FSI ESO Query Load #9<br>FSI ESO Query Load #10 | These work in the same manner as the normal Foreign System Interfaces ESO<br><br>Load servers. These servers are used only for PM Registry and PM Query triggers currently. | 293<br>294<br>295<br>296<br>297<br>298<br>299<br>300<br>301<br>302 | fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso<br>fsi_srveso |

9. The Foreign System Interfaces Notify server writes rows to the CQM_FSIESO_TR_1 table. The CQM_FSIESO_TR_1 table contains the triggers sent outbound. The number of rows written to this table is equal to the number of qualifying rows the Foreign System Interfaces Notify server found on the CQM_LISTENER_REGISTRY table in the previous step.
10. The Foreign System Interfaces Notify server sends a CMB Notify event to the Foreign System Interfaces ESO server. This can be any one of the Foreign System Interfaces ESO servers - SCP 280-284, 285, or 293-302 (see the table above).

Under optimal conditions, the Foreign System Interfaces ESO server drives off these CMB events and never looks at the CQM_FSIESO_TR_1 table. (Except for transactions that are put on hold because information from the CQM_FSIESO_TR_1 table is used for these transactions.) For instance, when the Foreign System Interfaces ESO server receives a CMB Notify event from the Foreign System Interfaces Notify server, it reads the CQM_FSIESO_QUE table, gets the compressed outbound trigger specified by that CMB Notify event, and processes it. For every row the Foreign System Interfaces Notify server writes to the CQM_FSIESO_TR_1 table, it also sends a CMB Notify event to a Foreign System Interfaces ESO server.

If the Foreign System Interfaces Notify server is sending these CMB Notify events faster than the Foreign System Interfaces ESO server can process them, it begins queuing these CMB Notify events in a memory-based queue called the thread queue. If this queue becomes full, the Foreign System Interfaces ESO server dynamically switches from driving off the CMB Notify events to driving off the CQM_FSIESO_TR_1 table. That is, the Foreign System Interfaces ESO server switches back and forth as necessary between driving from the CMB Notify events and driving from the CQM_FSIESO_TR_1 table. There may be times when the Foreign System Interfaces ESO server is flooded with triggers, causing the thread queue to fill. When it reaches 90% of its capacity, it forces the Foreign System Interfaces ESO server to drive off the CQM_FSIESO_TR_1 table. When the activity slows down and the Foreign System Interfaces ESO server catches up, it switches back to driving off the CMB Notify events.

11. The Foreign System Interfaces ESO server retrieves and uncompresses the outbound trigger on the CQM_FSIESO_QUE table. The CMB Notify event coming from the Notify server has the QUEUE_ID from the CQM_FSIESO_QUE table and the TRIGGER_ID from the row it wrote to the CQM_FSIESO_TR_1 table.
12. The Foreign System Interfaces ESO server processes the outbound trigger and creates an HL7 message. For the vast majority of the outbound triggers, this processing simply involves calling multiple scripts that go out to the various shared database tables, pull the required information, and format it into an HL7 message (the scripts are primarily involved in formatting the message). There are some triggers (currently only ORUs), however, that require the Foreign System Interfaces ESO server to call other solution servers (for example, the Clinical Event server) to get the information needed to format the HL7 message.
13. The Foreign System Interfaces ESO server writes the HL7 message to the CQM_OENINTERFACE_QUE table. The Foreign System Interfaces ESO server writes rows to the CQM_OENINTERFACE_TR_1 table based on the routing configurations for the OEN.ESO.Interface. (If the trigger is an HL7 version 3.x message, it is routed using the OEN.ESO.v3.Interface. If the ESO server is a Query Load server, it uses the OEN.ESO.Qry.Interface.)

    The HL7 message is compressed when it is written to this table, just as the outbound trigger was compressed on the CQM_FSIESO_QUE table. There is a field in the CQM_OENINTERFACE_QUE row that is used to indicate which outbound trigger (QUEUE_ID) and trigger row (TRIGGER_ID) created it. This is the CONTRIBUTOR_REFNUM field, and it is a concatenation of the QUEUE_ID from the CQM_FSIESO_QUE table and the TRIGGER_ID from the CQM_FSIESO_TR_1 table. For example, if the CONTRIBUTOR_REFNUM is 0999#######Q1111T2222, the QUEUE_ID from CQM_FSIESO_QUE is 1111, and the TRIGGER_ID from the CQM_FSIESO_TR_1 is 2222.

    If the HL7 message size exceeds the MAXBUFFERSIZE for the OEN Router, the Foreign System Interfaces ESO Server updates the process status flag for 70 (In Error) and the TRIGGER_STATUS_TEXT with the error message for the appropriate row on the CQM_FSIESO_TR_1 table.
14. The Foreign System Interfaces ESO server sends a CMB Notify event to the ESO comservers configured to receive the transaction. This CMB Notify event contains the QUEUE_ID, TRIGGER_ID and split transaction count from the CQM_OENINTERFACE_TR_1 rows the Foreign System Interfaces ESO server just wrote. Also, the Foreign System Interfaces ESO server sends a CMB Notify event to the OEN Router to update the statistics for the ESO comservers that the HL7 message was routed to.
15. The Foreign System Interfaces ESO server updates the process status flag to 90 (Processed Transaction) for the appropriate row on the CQM_FSIESO_TR_1 table. This is a critical step, because if it fails, the FSI ESO server receives an Oracle error and restarts. Because the Foreign System Interfaces ESO server has already processed the outbound trigger and written an HL7 message to the CQM_OENINTERFACE_QUE table, however, duplicate HL7 messages are created for the same outbound trigger.
16. The ESO comserver queries its queue and executes ESORTL. The Foreign System Interfaces ESO server places the HL7 object into the comserver's queue. The comserver then de-queues it and executes the ESORTL (ESO Runtime Library). The ESORTL searches for all format strings, such as ##CVA##, ##NULL##, and ##NOMEN##, inserted by the Foreign System Interfaces ESO server and replaces them with their aliases. The ESORTL uses the comserver's PACKESO trait to alias the coded values in the HL7 message. The PACKESO trait is actually the CONTRIBUTOR_SYSTEM_CD of the receiving foreign system. The ESORTL takes this CONTRIBUTOR_SYSTEM_CD and performs a select on the CONTRIBUTOR_SYSTEM table to get the CONTRIBUTOR_SOURCE_CD. The ESORTL then searches for all the null strings.

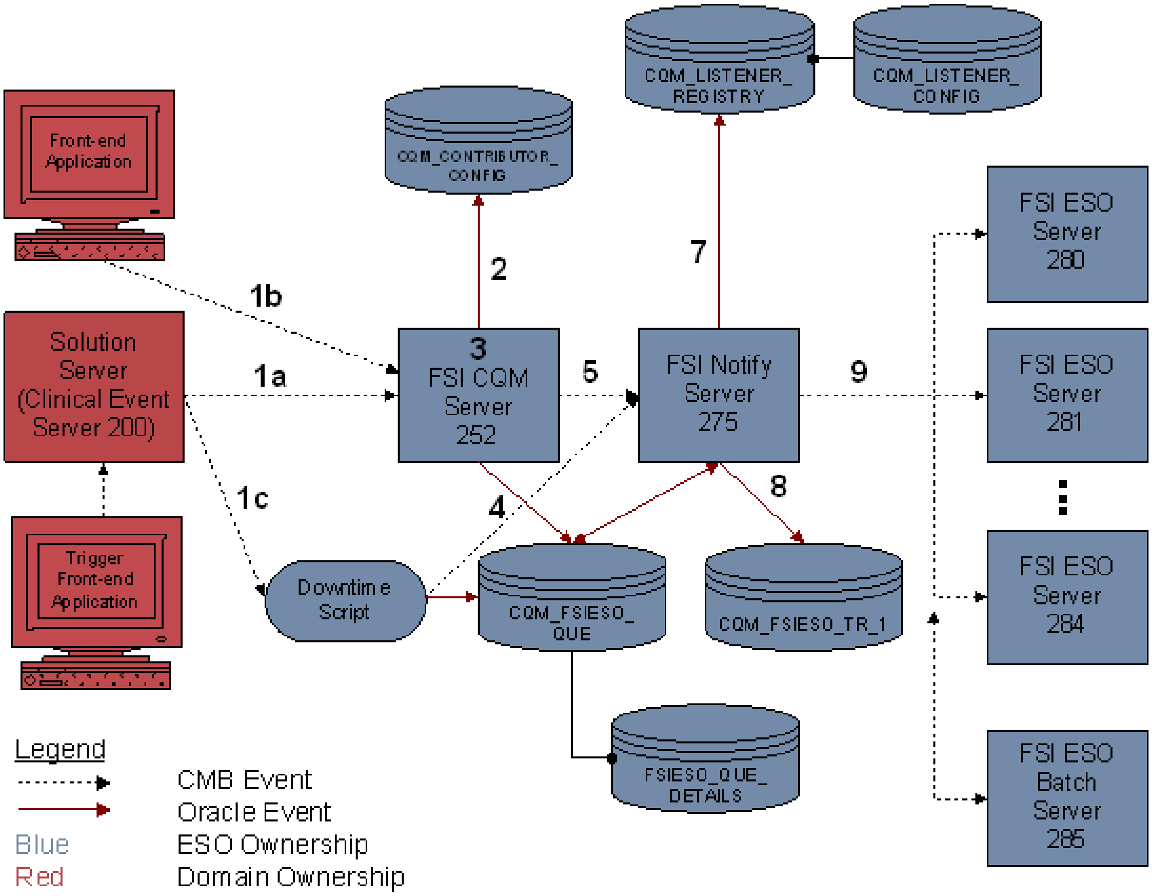
    The way the ESORTL processes the arguments after the format string depends on the type of format string. For example, a format string of ##CVA##,,11004 is recognized by the ESORTL as a code_value that needs to be aliased. Using the CONTRIBUTOR_SOURCE_CD and the CODE_VALUE (in this example, 11004), the ESORTL does a select on the CODE_VALUE_OUTBOUND table to get the outbound alias required by the foreign system. (Note that the ESORTL does not individually select the code value outbound alias as it encounters it, but rather caches them on start-up.) After the ESORTL has completed its processing, the MOD-OBJECT, MAP-FROM-LIBRARY, and finally the MOD-ORIGINAL scripts are executed.
17. The message is sent outbound and a copy of the outbound transaction is written to the OEN_TXLOG table (if enabled).

## Advanced ESO Concepts

Advanced ESO concepts are described below.

## ESO Load Balancing

The ESO load balancing process flow is shown below.



**Note**

The following steps are specifically directed toward ESO Load Balancing. There may be additional information for each step that is not shown here. See the ESO Processing Flow for a complete ESO data flow.

To determine which server receives the trigger, the Foreign System Interfaces Notify server gets a LISTENER_ID by selecting a row from the CQM_LISTENER_REGISTRY table using the class, type, subtype, and SUBTYPE_DETAIL contained in the outbound trigger. The LISTENER_ID is unique for each Foreign System Interfaces ESO server. A join is then performed on the CQM_LISTENER_CONFIG table using this LISTENER_ID. The qualifying row from this table indicates which Foreign System Interfaces ESO server to send the outbound trigger to, as well as any necessary parameters for that Foreign System Interfaces ESO server.

The following steps detail the ESO Load Balancing flow shown above.

1. The first step can occur in one of three ways:
    1. An application sends a transaction to the solution server, and then the solution server sends the transaction to the Foreign System Interfaces ESO CQM server (SCP 252) in the form of a CMB Insert Event (outbound trigger).
    2. The application sends the transaction directly to the Foreign System Interfaces ESO CQM server in the form of a CMB Insert Event.
    3. During Foreign System Interfaces ESO CQM server downtime, a CMB Notify event is sent to the Foreign System Interfaces Notify server (SCP 275) after writing the outbound trigger to the CQM_FSIESO_QUE table. This bypasses the Foreign System Interfaces ESO CQM server altogether, while still authenticating the outbound trigger (Step 2). If the Foreign System Interfaces Notify server is running, it picks up the row in turn and processes it. If the Foreign System Interfaces Notify server is down as well, when the Foreign System Interfaces Notify server starts up, it scans the CQM_FSIESO_QUE table for rows with a process status flag of 10 (Unprocessed Transaction). The Foreign System Interfaces Notify server recognizes the rows that were written while it was down and attempts to recreate the trigger.
2. The Foreign System Interfaces ESO CQM server authenticates the outbound trigger. Authentication of the outbound triggers is accomplished by selecting from the CQM_CONTRIBUTOR_CONFIG table using the *APPNAME* and *CONTRIBALIAS* contained in the outbound trigger. If there is no qualifying row, the outbound trigger is ignored.
   To see these outbound triggers, set the Foreign System Interfaces ESO CQM server *DUMP_INSERT* property to 1. Any outbound trigger that is sent to this server is printed to the .OUT or .ERR file in the cmb_temp directory. For more information on viewing the .OUT and .ERR files, see Server .OUT and .ERR Files.
3. The Foreign System Interfaces ESO CQM server performs a modulus operation on the PERSON_ID in the outbound trigger to determine which Foreign System Interfaces ESO server to route to. The algorithm the Foreign System Interfaces ESO CQM server uses actually is more complex than simply a modulus operation on the PERSON_ID, but essentially it can be looked at as such. By performing the modulus operation on the PERSON_ID, all the outbound triggers for a person are routed to Foreign System Interfaces ESO Load Server #1 (SCP 280). It is essential that all the transactions for a person are sent outbound in the same order as they were received. Therefore, by routing all the outbound triggers for a person to a single ESO server, their order is maintained. It is important to mention that currently a maximum of five Foreign System Interfaces ESO Load servers, one Foreign System Interfaces ESO Batch server, and ten Foreign System Interfaces ESO Query Load servers can be set up.

   The Foreign System Interfaces ESO CQM server looks for the following SCP properties on startup: the ESO_SERVER_COUNT is <# of ESO servers> and the BATCH_ESO_SERVER_COUNT is 1. From these properties, the Foreign System Interfaces ESO CQM server knows if ESO Load Balancing is turned on. If ESO Load Balancing is turned on, the Foreign System Interfaces ESO CQM server knows how many Foreign System Interfaces ESO servers it has available to route outbound triggers to, and if Batch Charges are implemented.

   If Batch Charges are implemented, Foreign System Interfaces ESO Server Batch #1 (SCP 285) is required. This results in all the charges being routed to the single Foreign System Interfaces ESO Batch server. This ensures that the order of the charges is maintained, specifically for the outbound header and trailer that indicate the beginning and end of the batch. If this order were not maintained, the outbound trailer could potentially reach the foreign system before all the charges. This would seem as if the foreign system received all the charges when in fact it did not.

4. The Foreign System Interfaces ESO CQM server writes the outbound trigger to the CQM_FSIESO_QUE table. When the Foreign System Interfaces ESO CQM server writes a row to this table, it extracts the class, type, and subtype from the outbound trigger and takes the value of the modulus operation (see the previous step) and write it to the SUBTYPE_DETAIL field. This means that by looking at the SUBTYPE_DETAIL column for a row on the CQM_FSIESO_QUE table you can see to which Foreign System Interfaces ESO server the outbound trigger was routed. Without ESO Load Balancing, the SUBTYPE_DETAIL field contains the PERSON_ID.
5. The Foreign System Interfaces ESO CQM server sends a CMB Notify event to the Foreign System Interfaces Notify server.

> **Note**
>
> If Hold Transaction Processing is active, Step 6 is executed. This step is not shown in the above illustration.

6. The system determines if this type of outbound trigger qualifies to be put on hold. For a detailed discussion of hold transaction processing, see Steps 4-7 in ESO Processing Flow. If the outbound trigger qualifies to be put hold, it is. When it is released, it continues where it left off with Step 7.
7. The Foreign System Interfaces Notify server explodes the outbound trigger and retrieves the parameters to the Foreign System Interfaces ESO server it is routing to. The Foreign System Interfaces ESO CQM server takes the class, type, subtype and SUBTYPE_DETAIL fields in the outbound trigger and matches them to a row or rows on the CQM_LISTENER_REGISTRY table. This step is referred to as exploding the outbound trigger. It is called this because the outbound trigger could potentially match on multiple rows on the CQM_LISTENER_REGISTRY table. A single outbound trigger would explode into multiple outbound triggers.

   After the Foreign System Interfaces Notify server has matched on a row or rows, it takes the LISTENER_ID and uses it to join to the CQM_LISTENER_CONFIG table. This returns a row from the CQM_LISTENER_CONFIG table that contains the binding name and the parameters that the Foreign System Interfaces ESO CQM server needs to route the outbound trigger to.
8. The Foreign System Interfaces Notify server writes rows to the CQM_FSIESO_TR_1 table. The CQM_FSIESO_TR_1 table contains the triggers sent outbound. The number of rows written to this table is equal to the number of qualifying rows the Foreign System Interfaces Notify server found on the CQM_LISTENER_REGISTRY table in the previous step.
9. The Foreign System Interfaces Notify server sends a CMB Notify event to the FSI ESO server. This can be any one of the Foreign System Interfaces ESO servers - SCP 280-284, 285, or 293-302.

   Under optimal conditions, the Foreign System Interfaces ESO server drives off these CMB events and never looks at the CQM_FSIESO_TR_1 table. (Except for transactions that are put on hold because information from the CQM_FSIESO_TR_1 table is used for these transactions.) For instance, when the Foreign System Interfaces ESO server receives a CMB Notify event from the Foreign System Interfaces Notify server, it reads the CQM_FSIESO_QUE table, gets the compressed outbound trigger specified by that CMB Notify event, and processes it. For every row the Foreign System Interfaces Notify server writes to the CQM_FSIESO_TR_1 table, it also sends a CMB Notify event to a Foreign System Interfaces ESO server.

   If the Foreign System Interfaces Notify server is sending these CMB Notify events faster than the Foreign System Interfaces ESO server can process them, it begins queuing these CMB Notify events in a memory-based queue called the thread queue. If this queue becomes full, the Foreign System Interfaces ESO server dynamically switches from driving off the CMB Notify events to driving off the CQM_FSIESO_TR_1 table. That is, the Foreign System Interfaces ESO server switches back and forth as necessary between driving from the CMB Notify events and driving from the CQM_FSIESO_TR_1 table. There may be times when the Foreign System Interfaces ESO server is flooded with triggers, causing the thread queue to fill. When it reaches 90% of its capacity, it forces the Foreign System Interfaces ESO server to drive off the CQM_FSIESO_TR_1 table. When the activity slows down and the Foreign System Interfaces ESO server catches up, it switches back to driving off the CMB Notify events.
10. See Step 11 in ESO Processing Flow for a continuation of the ESO processing flow.

# Troubleshooting Workshop

When troubleshooting outbound transactions, the best method is to step the transaction through the process using the ESO Troubleshooting module of SI Manager to determine where the breakdown occurred.

## Step 1: Gather the Essential Information

It is essential that at a minimum the following information be collected before troubleshooting a potential problem or logging a point with the IRC or SolutionWorks.

- What is the sites mnemonic?
- Which operating system is being used (AIX or OpenVMS)?
- In what domain is the problem occurring (for example, PROD, CERT)?
- What is the base ESO code version?
- Which Open Engine communication server is being used to send the data outbound?
- Which types of outbound interfaces are being implemented?
- Is ESO Load Balancing installed?

- If you have the transaction that is causing the issue, please provide a queue trace report from the ESO Troubleshooting module in SI Manager.

## Step 2: Check the Basics

Now that you have the essential information gathered, complete the following steps to begin the troubleshooting process:

1. Verify that all the ESO servers are running. To do this, access Olympus or SCP on the back end. Verify that all the servers identified in the ESO Processing Flow diagram are running. If any of them are stopped, try to restart them by right-clicking the server and starting it.
2. Verify that all the Open Engine communication servers are running. To do this, access the Openview (Openview.exe) Controller on the front end or SCP on the back end. Verify that the communication server of the system having the issue is running. If it is not running, try to restart it by right-clicking the server and starting it.
3. Review the Msgview for any errors that can help in determining the issue.
4. Verify that all the appropriate triggers are initialized in ESO_INIT_OUTBOUND---Option 3. This could be an issue if this is new functionality being added, or when an upgrade has occurred in the environment. To check the triggers, access SI Manager (ESO Configuration  Define ESO Interfaces), or on the back end, use ESO_INIT_OUTBOUND using *Discern Explorer* and select Option 2 (Define ESO Interfaces). Select the appropriate interface and press Enter. The list of possible triggers needed or used is viewable. If a trigger is not on that should be, in SI Manager, check the option for the trigger, or in ESO_INIT_OUTBOUND, select that trigger and then enter **A** for Activate.

## Step 3: Determine Where the Transaction Stopped in the ESO Process

1. Run the ESO Unprocessed report by using the reports view in the ESO Troubleshooting module in SI Manager.

   > **Caution**
   >
   > If the servers are in a backlogged state or heavily loaded, it is not recommended that you run this report because querying the tables for this information could cause more issues.

   This report can be helpful in finding where the breakdown is occurring. It provides a list of the process status flags on each table and the total count of each. This can help you locate where the process is backing up. The following table specifies the value of each status flag:

   | PROCESS_STATUS_FLAGValue | Meaning |
   |---|---|
   | 10 | Unprocessed transaction |
   | 30 | Transaction in process |
   | 50 | Transaction on hold |
   | 70 | Unable to process transaction |
   | 90 | Processed transaction |

2. Use the search feature in the ESO Troubleshooting module of SI Manager to search for a specific transaction and trace it through the outbound process. This can assist in determining how far the transaction was processed in the dataflow.

   There are many options to use when searching for a transaction. If you know the HL7 transaction that you are troubleshooting, you can look in the MSH-10 field. The queue ID of the CQM_FSIESO_QUE table is the number directly after the Q.

   Once you search for a transaction, you can right-click a row in the search results and look at the Overview screen. This screen gives you a visual diagram of the processing of the transaction through the ESO queues.

   From the Overview screen, you can double-click on the CQM_OENINTERFACE_QUE and see the table row for the queue row you are evaluating. Then, from the results at the bottom of the screen, you can right-click and select View Message to view the HL7 message before the custom scripts on the comservers have manipulated it.
3. From the Overview screen, you also can check the OEN_TXLOG record to ensure that the HL7 message is formatted according to the foreign system specification. If the transaction made it to the OEN_TXLOG, then it was sent to the foreign system. If ACK logging is turned on, the ACK (acknowledgement) received from the foreign system also should be logged here. Complete the following steps to check the OEN_TXLOG record:
   1. In the ESO Troubleshooting module of SI Manager, search for your transaction from the OEN_TXLOG search or one of the queues. Right-click the search results for your transaction and select Overview, or if searching from the OEN_TXLOG, select View Message.
   2. If using the Overview window, double-click the comserver that the transaction was routed to if it was successful. If it was not successful, the overview screen does not show a comserver routing.

> **Note**
>
> Transaction logging must be turned on. To turn on transaction logging, access OpenView (Openview.exe), click Process Config, select the Logging tab, and select Transaction Logging.

3. Compare the field values to those of the specification document to assure that the format complies.

With the information gained from executing these steps, you should be able to pinpoint where the breakdown occurred. The following sections provide troubleshooting steps once it is determined where the transaction processing stopped.

## If the Transaction Did Not Reach the CQM_FSIESO_QUE Table

1. Verify that the FSI ESO CQM (SCP 252) server is running by accessing Olympus or SCP and verifying that server 252 is in a running state. The SCP entry numbers are in the far-right Entry ID column. If the CQM server is not running, start it by right-clicking on the row for server 252 and selecting Start.
2. Cycle the Foreign System Interfaces ESO CQM server.
3. Use Olympus or SCP to verify that the solution server is in a running state.
4. Cycle the solution server.
5. Verify that the appropriate triggers have been turned on in the ESO Configuration module of SI Manager. If the triggers for the transaction have not been turned on, this may be another reason why the transaction is not accessing the CQM_FSIESO_QUE table. To check the triggers, complete the following steps:
    1. Access SI Manager and select the ESO Configuration module. Next, click Define ESO Interfaces.
    2. Select the appropriate interface for the outgoing transaction and press Enter. This displays a list of the possible triggers needed or used for the particular interface.
    3. If the trigger is not turned on, select the appropriate trigger and check the option.
    4. Save the trigger selection by clicking Save.

> **Caution**
>
> Do not turn on triggers in a production environment without first certifying the trigger in a test environment.

## If the Transaction Did Not Reach the CQM_FSIESO_TR_1 Table

1. Verify that the Foreign System Interfaces Notify server (SCP 275) is running by accessing Olympus or SCP and verifying that server 275 is in a running state. The SCP entry numbers are in the far-right Entry ID column. If the Foreign System Interfaces Notify server is not running, start it by right-clicking on the row for server 275 and selecting Start.
2. If the Foreign System Interfaces Notify server is running, verify that the status flag in the CQM_FSIESO_QUE table is equal to 90 (Completed) by searching for the transaction using the ESO Troubleshooting module in SI Manager. To do so, complete the following steps:
    1. Select the CQM_FSI_ESO_QUE search.
    2. Enter your search criteria and click Search.
    3. The search results are displayed at the bottom of the screen. From there, you can view the Process Status Flag to see if the transaction has been processed.
3. Cycle the Foreign System Interfaces Notify server.

## If the Transaction Did Not Reach the CQM_OENINTERFACE_QUE

1. Verify that the Foreign System Interfaces ESO Load servers (SCP 280--284) or the Foreign System Interfaces ESO Batch server (SCP 285) are running by accessing Olympus or SCP and verifying that the servers are in a running state. The SCP entry numbers are in the far-right Entry ID column. If the Foreign System Interfaces ESO Load servers are not running, start them by right-clicking on the appropriate rows and selecting Start.

> **Note**
>
> All of the Foreign System Interfaces ESO Load servers may not be running. The number of Foreign System Interfaces ESO Load servers running is based on ESO Load Balancing. Load Balancing is set up in ESO_INIT _OUTBOUND in *Discern Explorer* or in OpenView.

2. If the server is running, verify that the process status flag on the CQM_FSIESO_TR_1 table is Complete by viewing the CQM_FSIESO_TR_1 row in the search provided in the ESO Troubleshooting module of SI Manager.
3. If the process status flag is not Complete, see If the Transaction Did Not Reach the CQM_FSIESO_TR_1 Table.
4. Cycle the Foreign System Interfaces ESO Load servers or the Foreign System Interfaces ESO Batch server.

> **Caution**
>
> Do not change properties on a comserver or the OEN Router unless you know the consequences of changing them.

These properties can be modified in OEN_PROC_VIEW. To access this utility, enter the following command at the *Discern Explorer* prompt:

**OEN_PROC_VIEW go**.

This opens the Open Engine Personality File where you can change the router properties. If you change any of the properties, the Foreign System Interfaces ESO Load servers, Foreign System Interfaces ESO Batch Servers, Foreign System Interfaces ESO Query Load servers, and Foreign System Interfaces ESO Query Extract must be cycled for the changes to take effect.

5. Verify the routing.
6. Ensure that the transaction posted to the CQM_OENINTERFACE_QUE table by completing the following steps:
    1. Perform a search from one of the queues in the ESO Troubleshooting module of SI Manager.
    2. Right-click the search results for the transaction you are troubleshooting and select Overview. This option shows you, by showing the queue in green, if a transaction processed through the queue. The queue is gray if it did not process through.
    3. If the queue is green, double-click the CQM_OENINTERFACE_QUE to view the queue row at the bottom of the screen. There are various right-click options to view additional information.
7. In OpenView (Openview.exe), click Server Config and check the custom Route Script to determine if a custom route script is being used. Verify that the route script is directing transactions to the correct comservers. If custom routing is available, the router executes a Custom Route script that determines which ESO comservers receives the HL7 message. Otherwise, the comservers receive all transactions and it is the responsibility of the individual comserver to filter the transactions.

## If the Transaction Did Not Reach the Foreign System

1. In OpenView (Openview.exe), click Server Config and check the custom Route Script to determine if a custom route script is being used. Verify that the route script is directing transactions to the correct comservers. If custom routing is available, the router executes a Custom Route script that determines which ESO comservers receive the HL7 message. Otherwise, the comservers receive all transactions and it is the responsibility of the individual comserver to filter the transactions.
2. If the transaction did not make it to the OEN_TXLOG, verify that the outbound comserver is running. You can use the OpenView controller on the front end or SCP on the back end. Complete the following steps when using SCP:

    1. Find the process ID of your server by entering the following command at the SCP prompt:

        **dir -name *[part of the server name]***
    2. Access the name of the server, node, domain, and properties of the server by entering the following command at the SCP prompt:

        **show [id]**
        where the ID is the server ID
    3. To see if this server is running, enter the following command at the SCP prompt:

        **server -name *[part of the server name]***.
3. If the comserver is not running, start the comserver in SCP.

    1. Log in to SCP and enter the following command at the SCP prompt:

        **start [server_id]**
    2. Verify that the comserver is running by entering the following command at the SCP prompt:

        **server -name *[server name]***.

# Troubleshooting Audits

The audits for troubleshooting are described below.

## Msgview

All ESO servers (FSI_SRVCQM, FSI_SRVNOTIFY, and FSI_SRVESO) log messages to Msgview for failure events. Use the SCP *loglevel* property to specify the amount of logging.

Msgview logs critical errors for the Foreign System Interfaces CQM (SCP 252), Foreign System Interfaces ESO (SCP 280--285), and Foreign System Interfaces Notify (SCP 275) servers. Search for an event string of ESO_FAILURE. For the ESO server only, if *Discern Explorer* errors are noted in Msgview, use the *Discern Explorer* utility, SRVRTLVIEW, to clarify the error.

## Server .OUT and .ERR Files

Another source of information is the CMB output and error files. All Msgview logging is duplicated there. Use the SCP _debuglogging _property to specify the amount of logging. A value of 2 is the default. A value of 4 provides maximum logging. The debuglogging property settings are listed below.

| Debuglogging Property Setting | Output to CMB_TEMP file |
|---|---|

| | |
|---|---|
| 0 | Error level |
| 1 | Warning level |
| 2 | Audit level |
| 3 | Info level |
| 4 | Debug level |

There are two standard server error files, the .OUT file and the .ERR file. These files are located in the cmb_temp directory and are listed as cmb_[value]*01.out and cmb*[value]_01.err, where [value] is the CMB value of the server or interface being investigated.

The .OUT file is used to log informational messages about server activity. It keeps track of what the server did, which scripts it called, and so on. The .ERR file identifies errors in many cases, but most information for interfaces is logged to the .OUT file. Both of these files can be accessed from the back end and read using a standard operating system editor. These files also can be accessed using the ESO Troubleshooting module of SI Manager using the Display Files option. You can select the CMB_TEMP directory and select a file name, file extension (in this case, .OUT), or both.

On OpenVMS, these files remain locked while the server is running. To view the log files, the server needs to be cycled (stopped and started), at which point, it releases the lock on the old log files and creates new ones. Once the old log files are unlocked, you can use any standard operating system editor to view them.

## .RTL Files

The RTL files in the CCLUSERDIR should be monitored for *Discern Explorer* errors in the Foreign System Interfaces ESO server and in the comclient custom scripts. The Foreign System Interfaces CQM and Foreign System Interfaces Notify servers do not typically use .RTL file logging, because they do not use scripts.

# System Messages

For information on system messages across *Cerner Millennium*, including information on Msgview and middleware system messages, see System Messages Reference.

## External System Outbound System Messages

Some server system messages are generated as .MLG files and are viewable in MessageView. The server-specific standard .OUT and standard error (.ERR) files are in the cmb_temp directory, as indicated above.