

---

## Article

[Jeffrey Drumm](#) · Jan 7, 2020 3m read

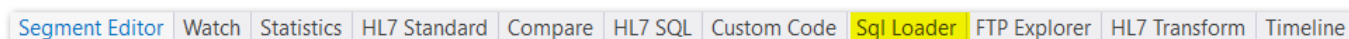
# Load Ensemble Messages into HL7 Spy v2020.1

I'm always on the lookout for tools that make the development and testing of my interfaces more efficient. A couple of years ago I came across HL7 Spy, from [Inner Harbour Software](#). It quickly became my go-to tool for running message comparison reports for interface engine migrations, message statistics gathering, and troubleshooting message receipt and delivery. It also offered enhanced functionality for things like fetching messages via sftp that other tools don't provide.

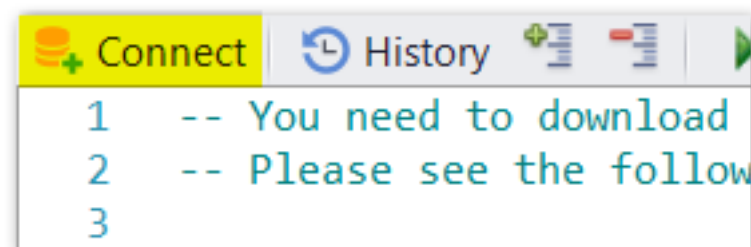
I've recently been working with HL7 Spy's author, Jon Reis, to enable support for fetching messages directly from the Ensemble message store. Its SQL Loader feature now has native Caché/IRIS support, and I've contributed a small server-side class to support the extraction of messages using it.

Version 2020.1 and higher, [downloadable from this page](#), includes this native support along with a sampling of SQL Loader queries that facilitate the retrieval of messages from Ensemble. This article provides a walk-through for configuring the latest release and shows some selected queries.

Once you've installed the beta, locate the SQL Loader tab at the bottom of the window and select it:



The bottom pane will change to an SQL editor window with a Connect button in the upper left corner:



Click Connect, then create a new connection profile by clicking the Add button, and filling in the form. Note: If your admin has implemented SSL/TLS security on the SuperServer port, include the string Ssl=True in the Additional Parameters text box (#8 in the image below); also, the default ports (#7) are 51773 for IRIS and 1972 for Caché:

The image shows the 'HL7Spy - Target Database Selection' dialog box. It has a left pane with a tree view showing '(none)' and 'Prod Env - HICG(Caché/IRIS) - prode...'. The right pane contains the following fields and controls:

- Profile Name:** 'Prod Env - HICG' (1)
- Category:** '(none)'
- Database Type:** 'Caché/IRIS' (2)
- User id:** 'jeff' (3)
- Password:** '\*\*\*\*' (4)
- Server:** 'prodenv' (5)
- Namespace:** 'HICG' (6)
- Port:** '51773' (7)
- Timeout (sec):** '10'
- Query:** '0' (secs. 0 = never)
- Additional Parameters:** 'Custom Connection String' (8)
- Buttons:** 'Test', 'OK', 'Cancel'
- Checkboxes:** 'Integrated Security' (unchecked), 'Store Password' (checked), 'Use Custom Connection String' (unchecked)

Next, click the Test button to verify that the connection works:

The image shows the 'HL7Spy - Target Database Selection' dialog box after clicking the 'Test' button. The 'Test' button is highlighted with a green box. Below it, a green box contains the text '✓ Connection Successful'. The 'Use Custom Connection String' checkbox is still unchecked. The 'OK' and 'Cancel' buttons are at the bottom right.

Finally, download the class [HICGHL7.xml](#), import it to your production namespace using Studio, and compile it. This class provides a SQL custom function called `HICG.GetMsg()` that takes a message body ID as its argument and populates a field in the returned result set with that ID's HL7 message.

That's it for installation and configuration. You can now take advantage of IRIS/Caché SQL to select messages based on values in `Ens.MessageHeader` or `EnsLib.HL7.Message`. A few examples:

1. Messages delivered to a Routing Process from a Service within a specified time frame

Click on each row in the bottom pane and HL7 Spy will display the related message in the upper pane.

HL7Spy Beta - Sql Loader

SQL Loader - Prod Env - HICG(Caché/IRIS) - prodenv\HICG

```
-- region 2 - Select by date/time range for messages between two hosts
SELECT SourceConfigName, TargetConfigName, TimeCreated, HICG.GetMsg(MessageBodyId) As Message FROM Ens.MessageHeader
WHERE
  MessageBodyClassName = 'EnsLib.HL7.Message'
  AND ID >= (SELECT TOP 1 ID FROM Ens.MessageHeader WHERE TimeCreated >='2020-01-03 00:00:00.000' ORDER BY TimeCreated ASC)
  AND ID <= (SELECT TOP 1 ID FROM Ens.MessageHeader WHERE TimeCreated <='2020-01-05 00:00:00.000' ORDER BY TimeCreated DESC)
  AND SourceConfigName = 'HL7SpyInbound'
  AND TargetConfigName = 'ADTRouter'
-- endregion
```

SourceConfigName	TargetConfigName	TimeCreated	Message
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.02	
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.03	
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.04	
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.05	
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.06	
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.07	

Segment Editor | Watch | Statistics | HL7 Standard | Compare | HL7 SQL | Custom Code | **Sql Loader** | FTP Explorer | HL7 Transform | Timeline

223 / 614

## 2. Suspended Messages

The screenshot displays the HL7Spy Beta - Sql Loader application. The top menu bar includes File, Home, and various tool icons. The main window is divided into three panes:

- Top Pane:** Contains a text editor with HL7 message data from 'AllMessages.txt'. The message includes fields like MSH, EVN, PID, PVL, DGI, OBX, and a description of pneumonia.
- Middle Pane:** Shows a SQL query in the 'SQL Loader - Prod Env - HICG(Cache/IRIS) - prodenv\HICG' environment. The query is designed to select suspended messages from the 'Ens.MessageHeader' table.
- Bottom Pane:** Displays the results of the SQL query in a table format, grouped by 'SourceConfigName' and 'TargetConfigName'.

**SQL Query:**

```

4  -- region 1 - Get Suspended messages
5  -- Status Values:
6  -- Created 1
7  -- Queued 2
8  -- Delivered 3
9  -- Discarded 4
10 -- Suspended 5
11 -- Deferred 6
12 -- Aborted 7
13 -- Error 8
14 -- Completed 9
15
16 SELECT SourceConfigName, TargetConfigName, TimeCreated, HICG.GetMsg(MessageBodyId) As Message FROM Ens.MessageHeader
17 WHERE
18     MessageBodyClassName = 'EnsLib.HL7.Message'
19     AND Status = 5
20 ORDER BY TargetConfigName ASC, TimeCreated ASC
21 -- endregion

```

**Query Results:**

	SourceConfigName	TargetConfigName	TimeCreated	Message
1	ADTRouter	ADTBucket	2020-01-06 09:49:23.40	
2	ADTRouter	ADTBucket	2020-01-06 09:49:23.41	
3	ADTRouter	ADTBucket	2020-01-06 09:49:23.43	
4	ADTRouter	ADTBucket	2020-01-06 09:49:23.51	
5	ADTRouter	ADTBucket	2020-01-06 09:49:23.52	

The bottom status bar shows the current segment editor is 'Sql Loader' and the total number of rows is 223 / 658.

### 3. Grouping

You can drag column headers to the grey bar at the top of the table pane, and records will be grouped/sub-grouped by the distinct values in those columns:

The screenshot shows the HL7Spy Beta - Sql Loader window. The SQL query is as follows:

```

23 -- region 2 - Select by date/time range for all inbound message destinations
24 SELECT SourceConfigName, TargetConfigName, TimeCreated, HICG.GetMsg(MessageBodyId) As Message FROM Ens.MessageHeader
25 WHERE
26     MessageBodyClassName = 'EnsLib.HL7.Message'
27     AND ID >= (SELECT TOP 1 ID FROM Ens.MessageHeader WHERE TimeCreated >= '2020-01-03 00:00:00.000' ORDER BY TimeCreated ASC)
28     AND ID <= (SELECT TOP 1 ID FROM Ens.MessageHeader WHERE TimeCreated <= '2020-01-05 00:00:00.000' ORDER BY TimeCreated DESC)
29     AND SourceConfigName = 'HL7SpyInbound'
30 -- endregion

```

The results pane shows the following data:

SourceConfigName	TargetConfigName	TimeCreated	Message
HL7SpyInbound	ADTRouter	2020-01-04 16:20:14.02	
HL7SpyInbound	ChargeRouter	2020-01-04 16:20:14.03	
HL7SpyInbound	ChargeRouter	2020-01-04 16:20:14.04	

#### 4. Selecting by both Message Header and Body Properties

The screenshot shows the HL7Spy Beta - (6) Sql Loader window. The SQL query is as follows:

```

54 -- region 6 - Get messages by message body name (derived from MSH:9)
55 SELECT
56     body.ID As BodyId,
57     head.SourceConfigName As Source,
58     body.DocType as DocumentType,
59     body.Name as BodyName,
60     body.FullSize as Size,
61     HICG.GetMsg(body.ID) As Message
62 FROM Ens.MessageHeader head
63 INNER JOIN EnsLib.HL7.Message body
64 ON head.MessageBodyId = body.ID
65 WHERE head.ID > 0
66 AND head.MessageBodyClassName = 'EnsLib.HL7.Message'
67 AND head.TargetConfigName = 'ADTRouter'
68 AND body.Name = 'ADT_A08'
69 -- endregion

```

The results pane shows the following data:

BodyId	Source	DocumentType	BodyName	Size	Message
5755	HL7SpyInbound	2.5.1:ADT_A01	ADT_A08	581	
5756	HL7SpyInbound	2.5.1:ADT_A01	ADT_A08	640	
5763	HL7SpyInbound	2.5.1:ADT_A01	ADT_A08	614	
5764	HL7SpyInbound	2.5.1:ADT_A01	ADT_A08	621	

#### Tips and Caveats

- Selecting messages by body (EnsLib.HL7.Message) properties seems to require a bit of a hack. You must provide a WHERE clause that starts with a selection by ID, even something as simple as ID > 0; in the case of a JOIN, that can be either the Header or Body ID.
- There's no reasonably performant option for selection by body content (other than the Identifier and Name properties, which are derived from MSH:10 and MSH:9 respectively). You should use IDs and other indexed properties such as SourceConfigName/TargetConfigName and Status to fetch the messages, then use HL7 Spy's powerful selection and filtering tools to find the specific message(s) you're looking for.
  - EDIT (10 January 2020): This is still generally a good idea, but I've incorporated the ability to do SearchTable queries as part of the message selection SQL query. That can speed things up ... see the comment below this article.
- Searching by a Date/Time range using TimeCreated is much faster if you use the method shown in example #1.

Finally ... I'll be updating this article as new discoveries are made and features added.

Thanks for reading 😊

[#HL7](#) [#Interoperability](#) [#Ensemble](#) [#InterSystems](#) [IRIS for Health](#)

---

Source URL: <https://community.intersystems.com/post/load-ensemble-messages-hl7-spy-v20201>