

ServiceControl Explorer User Manual

Last Updated: 12/16/2018

Overview

ServiceControl Explorer (SCE) is a WinForms application that allows you to make sense of messages stored in NServiceBus' ServiceControl. You will be able to efficiently and effectively search for messages in meaningful ways that will allow you to extract answers about "how", "what", and "when" something happened inside your NServiceBus deployment. Think of SCE as your swiss-army knife to understanding how your NServiceBus system works.

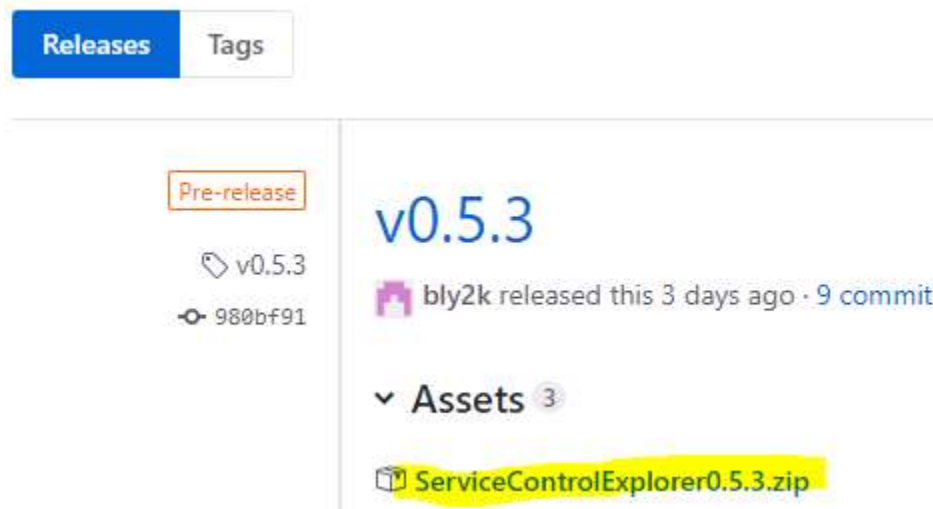
Software download

SCE releases are available for download here:

<https://github.com/bly2k/ServiceControlExplorer/releases>

You only need to download the zip containing the binaries and extract it to anywhere on your local machine. There is no need to download and/or compile the source code.

For example, below is the binary download for SCE 0.5.3:



Prerequisites

- You will need to first have Elasticsearch and Java installed. This is a one-time installation. After you install Elasticsearch, you will likely not need to install it again unless for any reason you need to swap drives or upgrade Elasticsearch in the future.

Note: It is possible to run and use SCE without installing Elasticsearch. When this happens, you can only connect and search messages on a remote ServiceControl instance. This is less powerful compared to

enabling local indexing of the messages into Elasticsearch but is nonetheless a viable option. For more details on this, see the section labeled “Disabling Local Indexing”.

Installation steps

Check/Install Java:

1. Open command prompt and run: `java -version -d64`
2. Check output from above and make sure there are no errors. Also Java version ***must be 1.8 or higher*** (for Elasticsearch)

```
C:\Users\bly>java -version -d64
java version "10.0.2" 2018-07-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)
```

3. Run: `echo %JAVA_HOME%`
4. Check that above JAVA_HOME environment variable is set and pointed to Java 1.8 (or higher) installation folder

```
C:\Users\bly>echo %JAVA_HOME%
C:\Java\jdk-10.0.2
```

5. If both above are good, skip the next steps and proceed below to installing Elasticsearch
6. Download and install Java/JDK for Windows x64 from here:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>
7. During installation, note the Java/JDK path you are installing to
8. Setup your Windows environment ***System*** variables as follows:

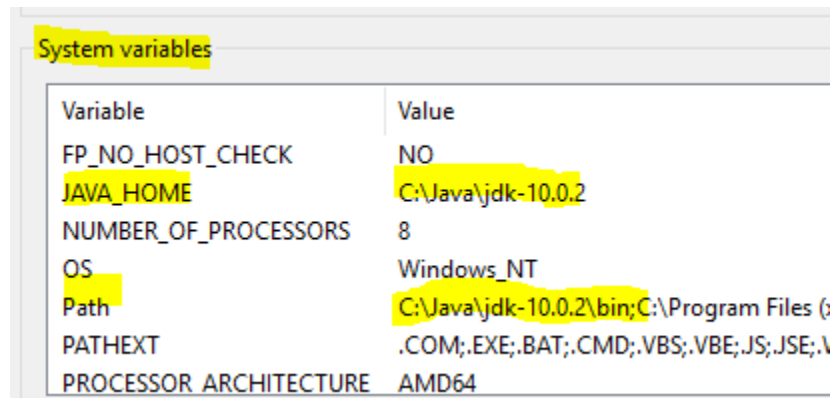
Note: In Windows there are two sets of variables, one for User and another for System. We want to change the System variables, and NOT the User variables.

Add the Java/JDK installation ***bin*** path to the Path ***System*** variable (make sure it is first in the Path list)

Example: On my system, I installed Java/JDK to `c:\java\jdk-10.0.2`, so I will add `c:\java\jdk-10.0.2\bin` as the ***first*** item in my Path system environment variable

Define (or change) the ***System*** variable named JAVA_HOME and set it to the ***root*** JDK installation path

Example: On my system, I set JAVA_HOME=c:\java\jdk-10.0.2

A screenshot of the Windows 'System variables' dialog box. The title bar is 'System variables'. It contains a table with two columns: 'Variable' and 'Value'. The variables listed are FP_NO_HOST_CHECK (NO), JAVA_HOME (C:\Java\jdk-10.0.2), NUMBER_OF_PROCESSORS (8), OS (Windows_NT), Path (C:\Java\jdk-10.0.2\bin;C:\Program Files (...)), PATHEXT (.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.VB...), and PROCESSOR_ARCHITECTURE (AMD64).

Variable	Value
FP_NO_HOST_CHECK	NO
JAVA_HOME	C:\Java\jdk-10.0.2
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Java\jdk-10.0.2\bin;C:\Program Files (...)
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.VB...
PROCESSOR_ARCHITECTURE	AMD64

9. Close and ***reopen*** command prompt and rerun: `java -version -d64`, and make sure Java now as expected (version 1.8 or higher)
10. Run: `echo %JAVA_HOME%`, and make sure this is pointed correctly as configured above (pointing to Java/JDK root installation path)

Install Elasticsearch 6.4.2

1. Download Windows Zip from here: <https://www.elastic.co/downloads/past-releases/elasticsearch-6-4-2>
2. Unzip to a local folder (note where the Elasticsearch bin folder is)
3. Open ***Admin*** command prompt and run (from the Elasticsearch bin folder): `elasticsearch.bat`
4. Make sure there are no errors and that Elasticsearch does not terminate early (i.e. command window will remain open)

```
C:\WINDOWS\system32\cmd.exe

] [WWss3eJ] adding template [.watch-history-9] for index patterns [.w
atcher-history-9*]
[2018-12-01T10:52:57,994][INFO ][o.e.x.s.t.n.SecurityNetty4HttpServer
Transport] [WWss3eJ] publish_address {127.0.0.1:9200}, bound_addresses
{127.0.0.1:9200}, {:::1}:9200}
[2018-12-01T10:52:57,994][INFO ][o.e.n.Node ] [WWss3eJ]
started
[2018-12-01T10:52:58,041][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.triggered_watches] for index patterns [.
triggered_watches*]
[2018-12-01T10:52:58,134][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.watches] for index patterns [.watches*]
[2018-12-01T10:52:58,228][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.monitoring-logstash] for index patterns
[.monitoring-logstash-6-*]
[2018-12-01T10:52:58,384][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.monitoring-es] for index patterns [.mon
itoring-es-6-*]
[2018-12-01T10:52:58,478][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.monitoring-alerts] for index patterns [.
monitoring-alerts-6]
[2018-12-01T10:52:58,556][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.monitoring-beats] for index patterns [.
monitoring-beats-6-*]
[2018-12-01T10:52:58,634][INFO ][o.e.c.m.MetaDataIndexTemplateService
] [WWss3eJ] adding template [.monitoring-kibana] for index patterns [.
monitoring-kibana-6-*]
[2018-12-01T10:52:58,869][INFO ][o.e.l.LicenseService ] [WWss3eJ]
license [14493ab0-af8f-4fa6-be4d-336dc1717dec] mode [basic] - valid
```

5. Open browser and verify that the Elasticsearch node is responding: <http://localhost:9200/>

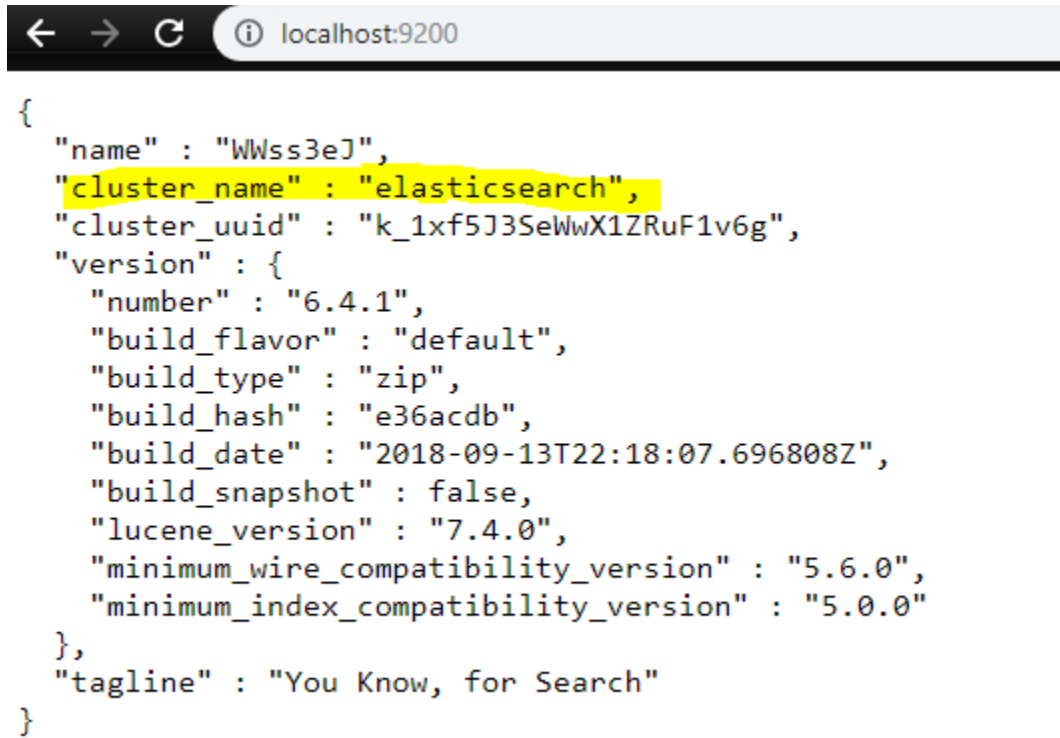


```
{
  "name" : "WWss3eJ",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "k_1xf5J3SeWwX1ZRuF1v6g",
  "version" : {
    "number" : "6.4.1",
    "build_flavor" : "default",
    "build_type" : "zip",
    "build_hash" : "e36acdb",
    "build_date" : "2018-09-13T22:18:07.696808Z",
    "build_snapshot" : false,
    "lucene_version" : "7.4.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

6. **Terminate** the running Elasticsearch instance (Ctrl+C) and make sure it shuts down cleanly (i.e. no Elasticsearch node/instance should be running before you proceed)
7. Install Elasticsearch as a service. Run: elasticsearch-service.bat install
8. Open windows services and find and then start the Elasticsearch service

Downloaded Maps Manager	Windows service for application access to downloaded maps...	Running	Automatic (D...
Elasticsearch 6.4.2 (elasticsearch-service-x64)	Elasticsearch 6.4.2 Windows Service - https://elastic.co		Automatic
Embedded Mode	The Embedded Mode service enables scenarios related to Ba...		Manual (Trig...
EMCIServ	Client Service	Running	Automatic

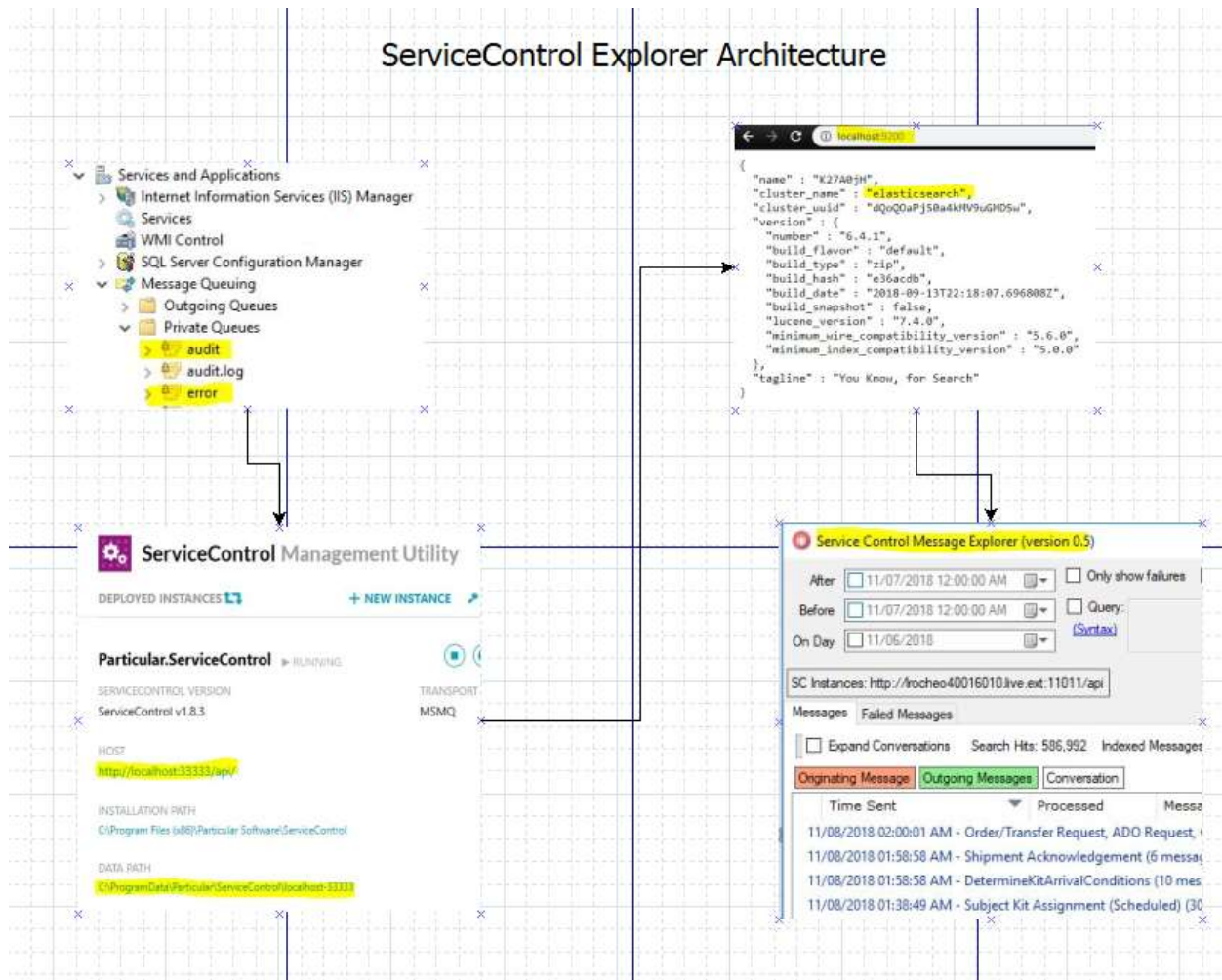
9. Open browser and verify that the Elasticsearch node is responding: <http://localhost:9200/>



```
{
  "name" : "WWss3eJ",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "k_1xf5J3SeWwX1ZRuF1v6g",
  "version" : {
    "number" : "6.4.1",
    "build_flavor" : "default",
    "build_type" : "zip",
    "build_hash" : "e36acdb",
    "build_date" : "2018-09-13T22:18:07.696808Z",
    "build_snapshot" : false,
    "lucene_version" : "7.4.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

ServiceControl Explorer Architecture

The SCE architecture is very simple and can be summarized in the below diagram:



The above is explained as follows:

- 1) From the top left, an NSB deployment sends all messages to a destination MSMQ audit/error queue
- 2) From the bottom left, ServiceControl monitors the audit/error queue and pulls down the messages into its internal storage
- 3) From bottom right, SCE calls the ServiceControl APIs to download the (JSON) messages from its internal storage.

Important Note: When SCE downloads the messages, this operation is non-blocking (i.e. think of this as a background task operation). In addition, the download is only as fast/good as your network infrastructure. SCE can download the messages as fast your SC/network infrastructure allows but your experience here will ultimately be constrained by how good your infrastructure is.

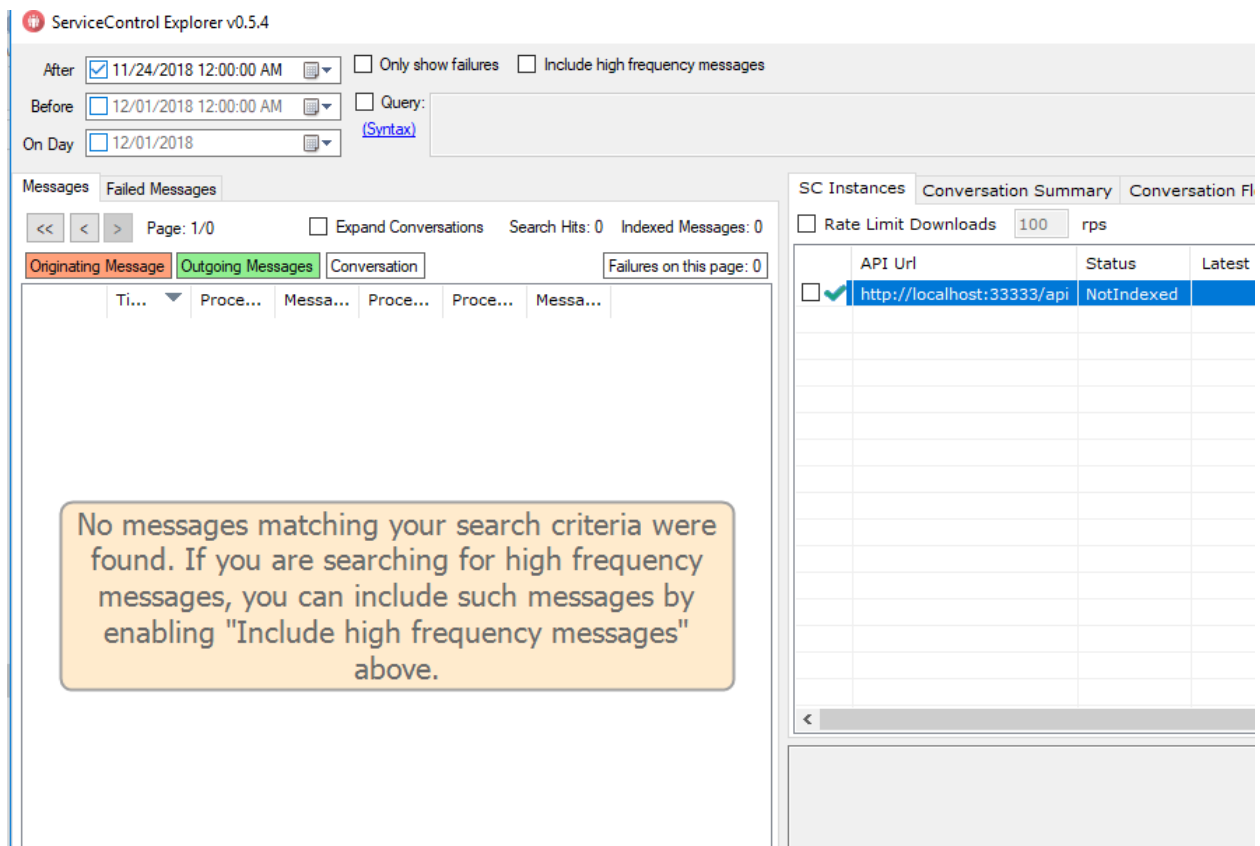
Downloads can also be throttled if needed, which will be described elsewhere in this guide.

- 4) From the top right, SCE stores the downloaded JSON messages into Elasticsearch
- 5) Once a message is download and indexed into Elasticsearch, it is immediately (within ~1 second) available for search and analysis.

Note: It is possible to do simple keyword searches against the ServiceControl data from SCE without downloading the messages down into Elasticsearch. However, the search that can be performed this way is less powerful than if you download all the data locally first. For more details on this, see the section labeled “Remote Messages”

Starting ServiceControl Explorer

After you download and unzip the SCE binaries to your local machine, run ServiceControlExplorer.exe. You will initially get this screen:



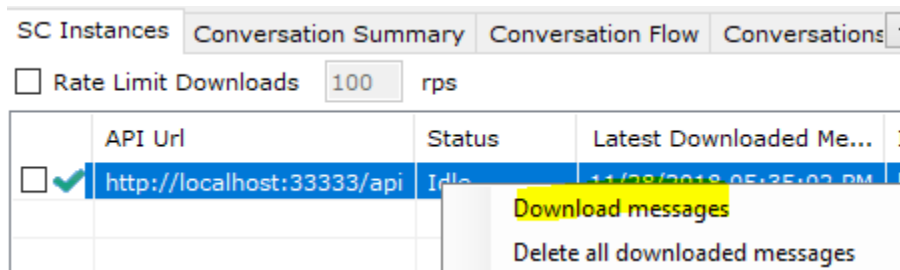
This means that:

1. You do not have any messages yet downloaded from any ServiceControl instances into your local Elasticsearch node.

2. On the right side, a default a localhost ServiceControl instance is preconfigured (pre-defined) for you. This localhost instance has an auto-download capability, meaning it will download messages from that ServiceControl instance often without user-intervention. This is a convenience for local development. If you do not have a local ServiceControl instance on your machine, you will see an error when it auto-downloads. You can just safely ignore the error in this case – this is OK and will not interfere with the functionality of SCE.

Downloading messages

Downloading messages from an instance is easy – just right click an instance and select “Download messages” as follows:



SCE will then start downloading the recent most messages (audit and errors) from the selected ServiceControl instance. Again, this operation is non-blocking and will not stop you from doing search/analysis on the download messages. As soon as a single downloaded message is indexed into Elasticsearch, you can search and analyze it immediately.

Again, note that the localhost instance has auto-download capability so unless you want to download localhost messages in real time, localhost messages should always be fairly up-to-date to within the last minute or so.

A few points to note:

1. If for any reason the download fails, you will see an error icon/status. To see the error, just right click on the instance and select “Show error” for more details of what has gone wrong.
2. The first download will pull down all messages from the instance. Depending on your infrastructure and message volume, this can take a while.
3. Succeeding downloads (after the first full download completes successfully) will only download new messages that have not yet been downloaded into SCE.
4. A download in progress can be stopped at any time (just right click on the instance and select “Stop download”. A stopped (or failed) download is resumable. What this means is if you stop (or fail) at the 50% mark for example, you can resume the download from that point instead of starting from the beginning again.
5. If for any reason you want to redownload everything again, you can “Delete all downloaded messages” for an instance, and then start from the beginning again.

Partial Downloads

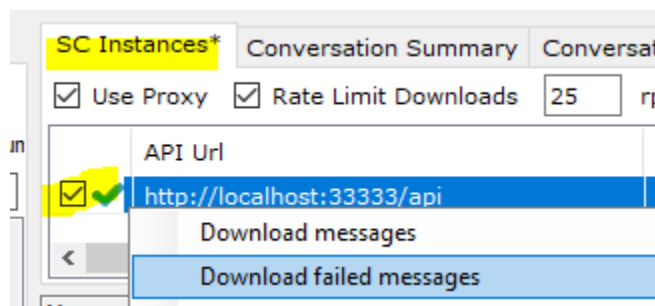
By default, SCE will download all messages (audit and errors) from a ServiceControl instance. If this is not desirable, it is possible to selectively download only specific messages. This is useful for example in the following scenarios:

1. Your instance might contain a lot of messages but you are only interested in looking at a small subset of messages (for example: a specific conversation(s), or messages with a specific date value, a series of correlated conversations, etc.)
2. You want to be able to pull the messages down selectively first and then analyze them locally offline using the rich capabilities of SCE.
3. You only want to pull failed messages and nothing else

Partial downloads can be done in any of the following ways:

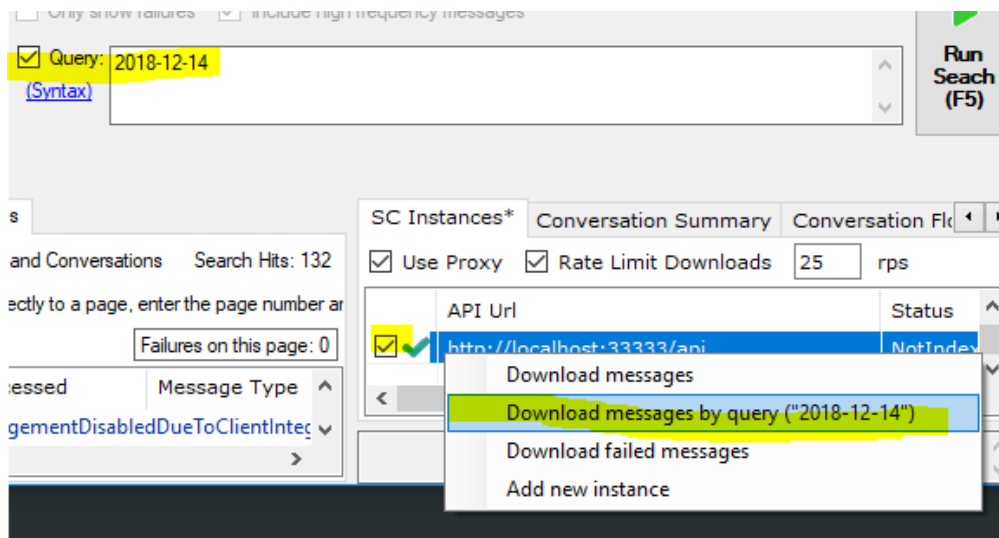
1. Download only failed messages

Highlight any instance from the SC Instances list, right click, and select “Download failed messages”:



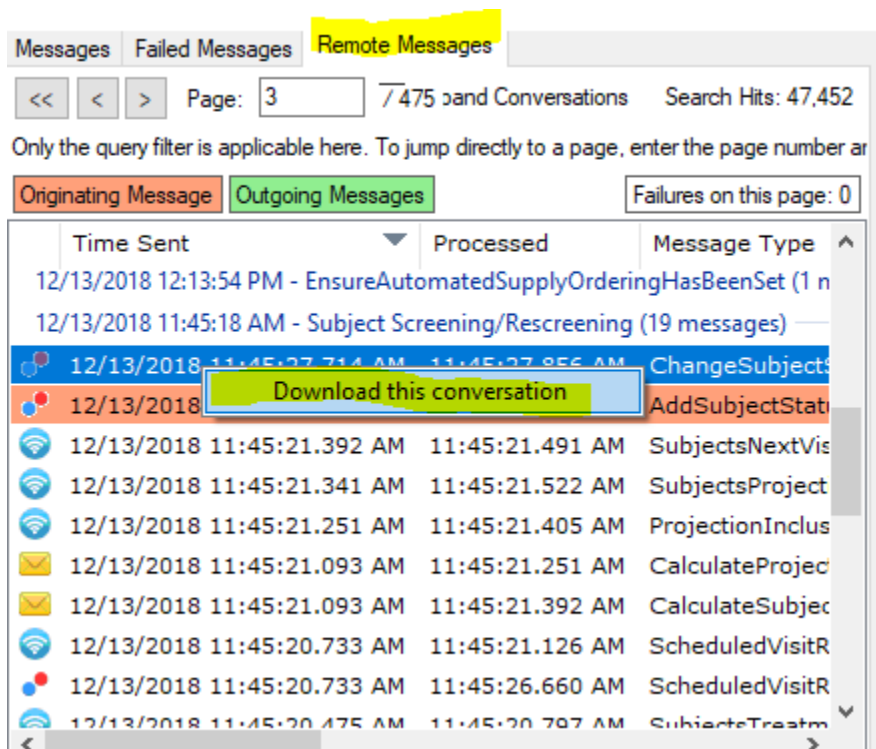
2. Download messages by query

You will first need to run a search query against a remote instance (see section labeled “Remote Messages” for more details). Once you get the desired search results from the remote search, highlight the currently selected instance, right click, and select “Download messages by query”:



3. Download messages in a single conversation

You will first need to run a search query against a remote instance (see section labeled “Remote Messages” for more details). Once you get the desired search results from the remote search, highlight any single message in the results, right click, and select “Download this conversation”:

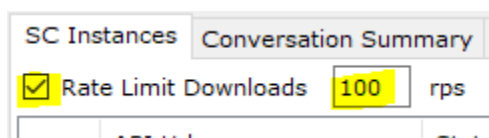


Important Note: For a single instance, only 1 active download can occur at any given time. If a download is currently in progress for an instance, you will not be able to start another download until either the active download is stopped, fails, or successfully completes.

Throttling Downloads

If you find that SCE is downloading “too fast”, you can throttle the download. This functionality is only available for non-localhost ServiceControl instances. The steps below will describe how to accomplish this:

1. If you want to throttle an existing/in-progress download, first stop the download.
2. Check the “Rate Limit Downloads” option and supply a reasonable value for the rps (HTTP requests per second) textbox:



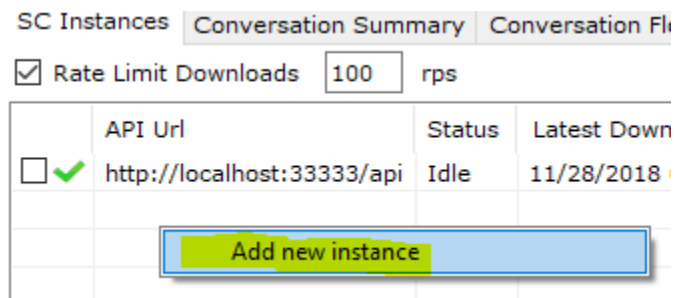
The screenshot shows the 'SC Instances' tab with the 'Conversation Summary' sub-tab selected. The 'Rate Limit Downloads' checkbox is checked, and the 'rps' value is set to 100.

3. Start or resume your download.

Rate limited downloads apply to all downloads that you start after changing the rate limit. For example, if you download from more than one instance after changing the rate limit, it will then be applied simultaneously to all active downloads across the instances. In addition, changing the rate limit while an existing download is in progress has no effect on the existing download in progress. If you want the rate limit to apply to an existing download, you must first stop the download, change the rate limit, and resume the download.

Adding a New Service Control Instance

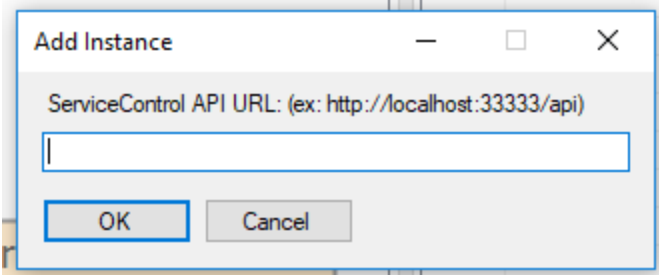
You can connect/download to as many ServiceControl instances as you like. To add a new instance, just right click anywhere on the SC Instances tab and select “Add new instance”. You will then be prompted for the instance API URL and then SCE will quickly ping it and if successful, the instance will be added. Once added, an instance is persistent across SCE restarts.



The screenshot shows the 'SC Instances' tab with the 'Conversation Summary' sub-tab selected. The 'Rate Limit Downloads' checkbox is checked, and the 'rps' value is set to 100. Below this is a table with one instance:

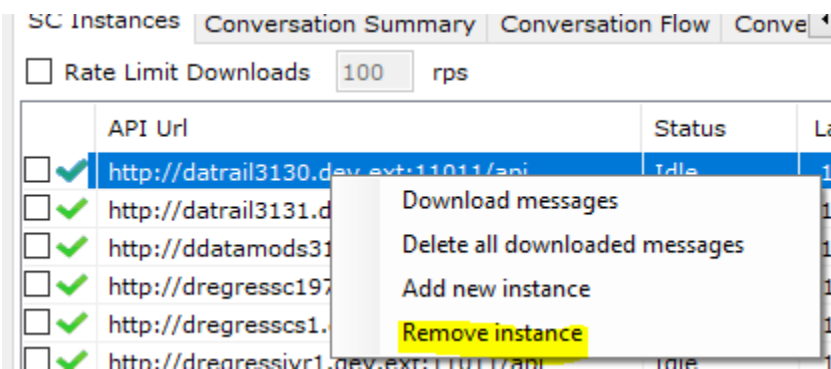
	API Url	Status	Latest Down
<input type="checkbox"/> ✓	http://localhost:33333/api	Idle	11/28/2018

Below the table is a button labeled 'Add new instance'.



Removing an Existing Service Control Instance

You can remove an existing ServiceControl instance if you no longer want to work with it. To remove an instance, just right click on it and select “Remove instance”:



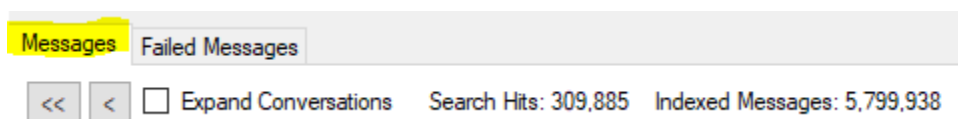
Note that removing an instance will simply remove it from the SC Instances list. The downloaded messages for that instance (if any) will remain intact inside Elasticsearch.

Searching Messages

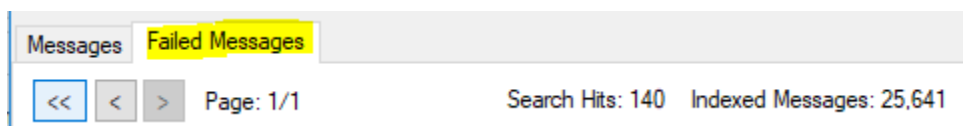
Note: Unless otherwise noted, all dates/times shown in SCE are local to the machine that runs SCE.

There are 2 sets of messages available for search:

1. Audit Messages – these will all be shown in the Messages tab



2. Error Messages – these will all be shown in the Failed Messages tab

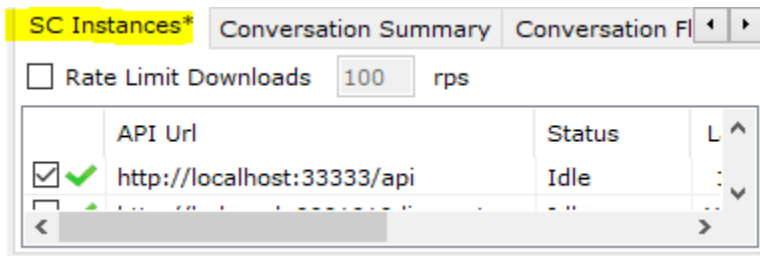


Depending on your workflow, you will likely be working mostly with one of the above set of messages.

Message Search (Audit Messages)

You have the ability to search messages by

1. ServiceControl instance



2. Date/Range (against the message time_sent property)

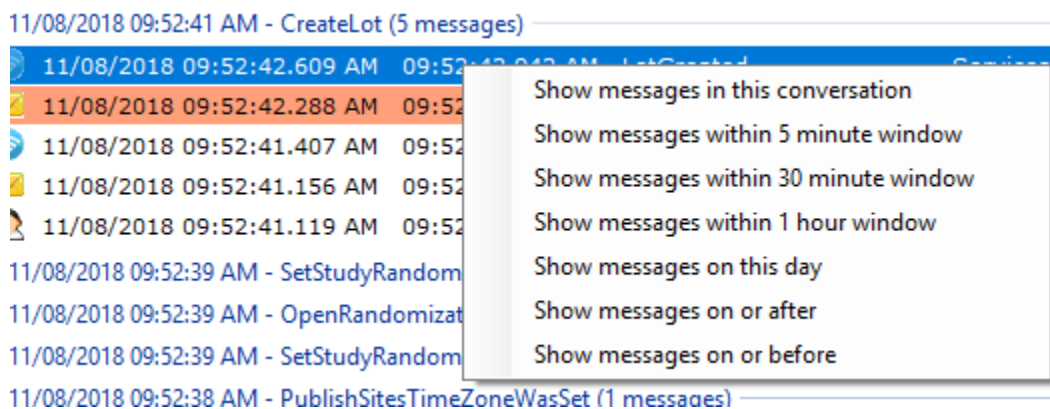
After ☒ 11/24/2018 12:00:00 AM

Before ☐ 12/01/2018 12:00:00 AM

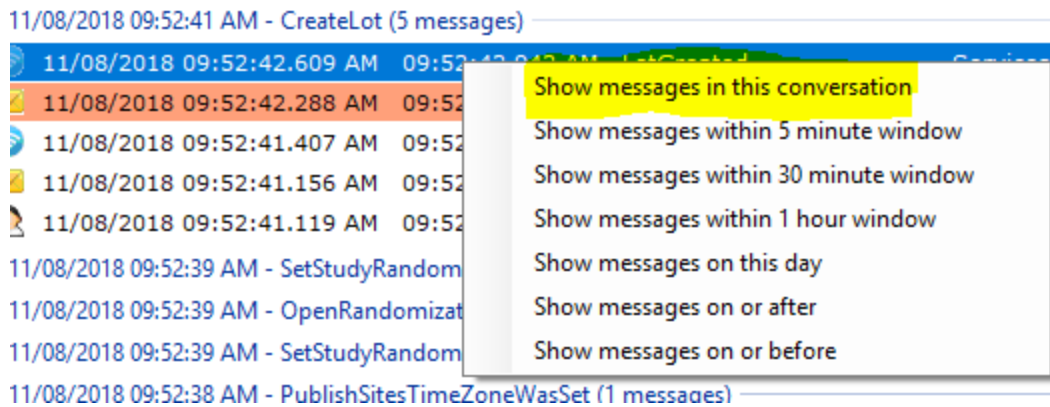
On Day ☐ 12/01/2018

Note: When SCE runs, the "After" date filter is initialized to 7 days prior to the current day, meaning, by default, you will only see messages within the past 7 days. It is important to keep this filter on all the time for performance reasons when dealing with a lot of messages.

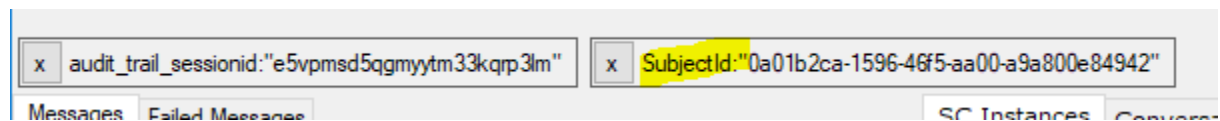
3. Time range window surrounding any single message:



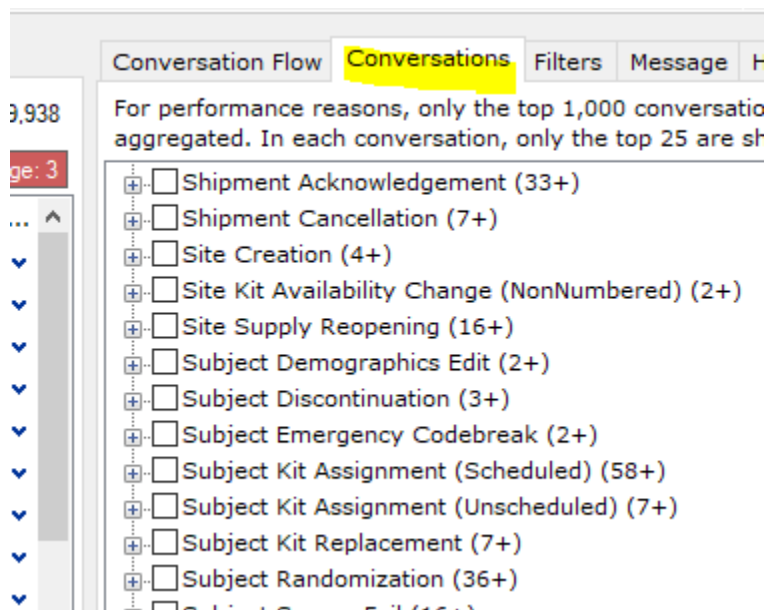
4. Single conversation



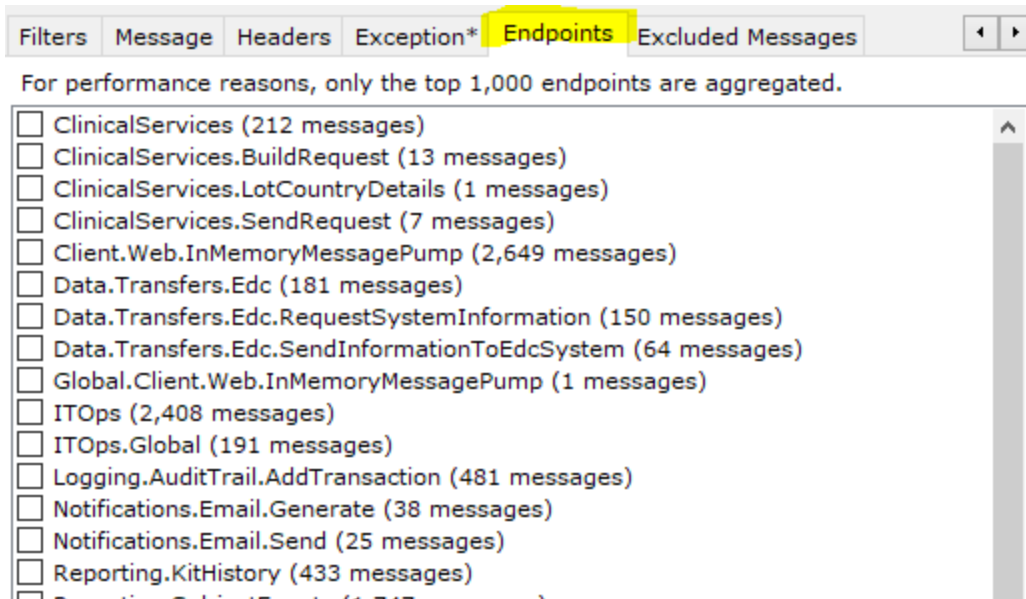
5. Message property filters (can also filter by instance, or by endpoint)



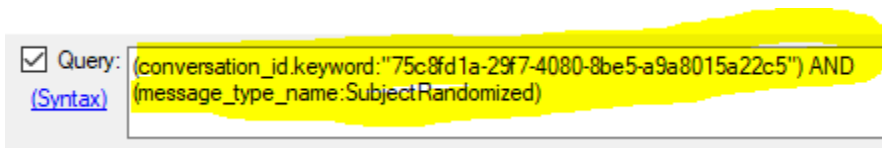
6. Conversation type (more on this later)



7. Endpoint

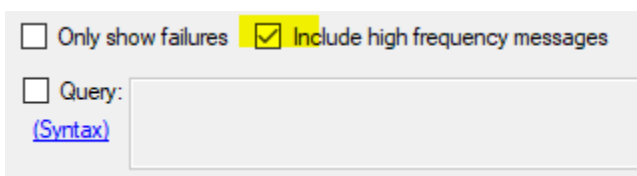


8. Any arbitrary boolean/full-text query (for advanced users)



In addition, high-frequency/"noise" messages are excluded by default. These generally are messages that recur very often and do not necessarily provide real value for most daily searches. However, there are some things you can do with high-frequency messages:

1. You can enable/include these messages in your search



2. You can customize this list if absolutely needed. For daily usage, this should not be necessary.

Message Search Results

Message search results are automatically grouped into NServiceBus conversations. This is where the power of SCE begins. A conversation typically equates to an event that your system did (or more specifically, an event that is either currently still in progress or that has already completed).

Conversations are easy to understand, discuss, and describe to people who new or are not very familiar with NServiceBus. Examples of conversations are:

- 1) A subject was randomized
- 2) A scheduled visit was completed
- 3) A drug order/transfer requisition was initiated

If we narrow down messages first by date/time (when) and then by conversation (what), then we can efficiently find out about any event that has occurred in our system. From here, we can start to investigate other things such as what other conversations are related to the current conversation at hand and intelligently piece together a story/timeline of what has happened, when it happened, and how it happened. This is the essence of SCE.

In the below example, you can see multiple conversations: Site Creation (9 messages), Seeding (1 message), SaveAutomatedSupplyOrderingStatus (2 messages), etc.:

The screenshot shows the NServiceBus Messages console interface. At the top, there are tabs for 'Messages' and 'Failed Messages'. Below the tabs, there are navigation buttons (back, forward, search), a page indicator 'Page: 1/32', a checkbox for 'Expand Conversations', and search statistics 'Search Hits: 31,238' and 'Indexed Messages: 37,876'. Below this, there are three tabs: 'Originating Message' (highlighted in orange), 'Outgoing Messages' (highlighted in green), and 'Conversation'. To the right of these tabs is a box that says 'Failures on this page: 0'. The main area displays a list of conversations, each with a timestamp, a description, and a count of messages. The list is as follows:

Time	Process	Message Count	Process	Process	Message Count
11/05/2018 06:45:14 PM	Site Creation	9 messages			
11/05/2018 06:44:28 PM	Seeding	1 messages			
11/05/2018 06:39:02 PM	SaveAutomatedSupplyOrderingStatus	2 messages			
11/05/2018 06:38:41 PM	SeedEdcWebServicePassword	1 messages			
11/02/2018 10:29:40 AM	Subject Randomization	38 messages			
11/02/2018 10:29:35 AM	Randomization Treatment Reservation	3 messages			
11/02/2018 10:28:01 AM	SaveEdcStatus	27 messages			
11/02/2018 10:27:35 AM	Subject Screening/Rescreening	25 messages			
11/02/2018 10:25:40 AM	Subject Randomization	36 messages			
11/02/2018 10:25:33 AM	Randomization Treatment Reservation	3 messages			
11/02/2018 10:25:22 AM	Subject Screening/Rescreening	26 messages			
11/02/2018 10:25:07 AM	Shipment Acknowledgement	11 messages			
11/02/2018 10:24:46 AM	Order/Transfer Request	107 messages			

At a high-level without looking at any specific individual message, you can clearly see what events are happening chronologically inside your system. In addition, scanning from the top of the results tab you can see:










1. There are 32 pages of messages (that match the current search criteria). By default, each page consists of 1,000 messages.
2. You are currently on page 1 (of 32)
3. There are 31,238 messages that match the current search criteria (thus the 32 pages)
4. There are a total of 37,876 messages indexed for this instance.

Note: Keep in mind that by default, high-frequency messages are excluded from your search unless explicitly enabled as described above.

Note: The conversation groupings displayed are automatically derived from the NServiceBus messages. The actual label of the conversations are predefined in SCE and is user-configurable. If a conversation is predefined in SCE, then the predefined conversation label will be displayed. Otherwise SCE will pick the first message in the conversation and use its message type name as the label for the conversation.

Note: If a conversation spans more than 1 day (for example, a conversation that triggers a saga timeout at a later date), it is broken out separately by days. This is to avoid confusion of when a specific message in a conversation was actually sent out.

Expanding the first conversation (Site Creation) produces this:

Originating Message		Outgoing Messages		Conversation: Site Creation (9 messages)		Failures on this page: 0	
Time Sent	Processed	Message Type	Processing Endp...				
11/05/2018 06:45:14 PM - Site Creation (9 messages)							
	11/05/2018 06:45:16.882 PM	06:45:16.927 PM	SitesSupplierScheduledSupply...	Services.Developer	U		
	11/05/2018 06:45:16.811 PM	06:45:16.843 PM	SitesDoNotShipIntervalsEvalu...	Services.Developer	U		
	11/05/2018 06:45:16.754 PM	06:45:16.901 PM	SitesSupplierEvaluated	Services.Developer	U		
	11/05/2018 06:45:16.151 PM	06:45:16.754 PM	DetermineIfSiteHasASupplier	Services.Developer	U		
	11/05/2018 06:45:16.129 PM	06:45:16.811 PM	DetermineIfSiteHasDoNotShip...	Services.Developer	U		
	11/05/2018 06:45:14.967 PM	06:45:16.698 PM	SiteAdded	Services.Developer	U		
	11/05/2018 06:45:14.967 PM	06:45:16.460 PM	SiteAdded	ITOps	U		
	11/05/2018 06:45:14.828 PM	06:45:14.953 PM	PublishSitesTimeZoneWasSet	Services.Developer	U		
	11/05/2018 06:45:14.796 PM	06:45:14.967 PM	PublishSiteAdded	Services.Developer	U		

From the above:

1. You can see the 9 individual messages that comprise the conversation
2. Highlighting the first chronological message in the conversation (PublishSiteAdded), you can see that it is processed by a handler that in turn sends out an outgoing message (SiteAdded). The SiteAdded message is then handled in 2 endpoints (ITOps and Services.Developer)










Highlighting the SiteAdded message/handler in Services.Developer, we then get this:

Originating Message

Outgoing Messages

Conversation: Site Creation (9 messages)

Failures on this page: 0

Time Sent	Processed	Message Type	Processing Endp...	
11/05/2018 06:45:14 PM - Site Creation (9 messages)				
 11/05/2018 06:45:16.882 PM	06:45:16.927 PM	SitesSupplierScheduledSupply...	Services.Developer	U
 11/05/2018 06:45:16.811 PM	06:45:16.843 PM	SitesDoNotShipIntervalsEvalu...	Services.Developer	U
 11/05/2018 06:45:16.754 PM	06:45:16.901 PM	SitesSupplierEvaluated	Services.Developer	U
 11/05/2018 06:45:16.151 PM	06:45:16.754 PM	DetermineIfSiteHasASupplier	Services.Developer	U
 11/05/2018 06:45:16.129 PM	06:45:16.811 PM	DetermineIfSiteHasDoNotShip...	Services.Developer	U
 11/05/2018 06:45:14.967 PM	06:45:16.698 PM	SiteAdded	Services.Developer	U
 11/05/2018 06:45:14.967 PM	06:45:16.460 PM	SiteAdded	ITOps	U
 11/05/2018 06:45:14.828 PM	06:45:14.953 PM	PublishSitesTimeZoneWasSet	Services.Developer	U
 11/05/2018 06:45:14.796 PM	06:45:14.967 PM	PublishSiteAdded	Services.Developer	U

From the above:

1. You can see that SiteAdded (in Services.Developer) is processed by a handler that then sends out 2 more messages (DetermineIfSiteHasASupplier and DetermineIfSiteHasDoNotShipIntervals).
2. You can see that SiteAdd (in Services.Developer) originated from a handler that processed PublishSiteAdded.

You can repeat the above process by highlighting any message in the conversation and you can tell where you came from (originating message) and where you are going next (outgoing messages). This allows you to “single-step” and correlate messages individually by causation.

When you are highlighting a single message, you will be able to see the message body/payload on the bottom right section:

Originating Message

Outgoing Messages

Conversation: Site Creation (10 m)

Failures on this page: 0

Time Sent	Processed	Message Type
11/05/2018 06:45:14 PM - Site Creation (9 messages)		
11/05/2018 06:45:16.882 PM	06:45:16.927 PM	SitesSupplierScheduledSupply
11/05/2018 06:45:16.811 PM	06:45:16.843 PM	SitesDoNotShipIntervalsEvalu
11/05/2018 06:45:16.754 PM	06:45:16.901 PM	SitesSupplierEvaluated
11/05/2018 06:45:16.151 PM	06:45:16.754 PM	DetermineIfSiteHasASupplier
11/05/2018 06:45:16.129 PM	06:45:16.811 PM	DetermineIfSiteHasDoNotShip
11/05/2018 06:45:14.967 PM	06:45:16.698 PM	SiteAdded
11/05/2018 06:45:14.967 PM	06:45:16.460 PM	SiteAdded

API Url

☒ http://localhost:33333/api

Message Id: 4d79d960-e12d-4377-8e52-a98f01877638

Time Sent: 11/05/2018 06:45:16.129 PM

Processed: 11/05/2018 06:45:16.811 PM

Body:

```
{
  "SiteId": "f83f1d1a-802c-4046-a4ed-a98f01350dd2",
  "CountryId": "e3ef53f2-d302-b22a-64a7-6e0ce65be769"
}
```

You can also quickly see the payloads of any unselected message by hovering over the message quickly in the messages list. SCE will show quick tooltip with a snippet of the message payload.

Conversation Flow

The conversation messages list gives you a “flat” chronological display of messages. It is also possible to display a higher-level “flow” display of messages. When a single conversation is highlighted, you can switch to the Conversation Flow tab to see the “flow” level visualization:

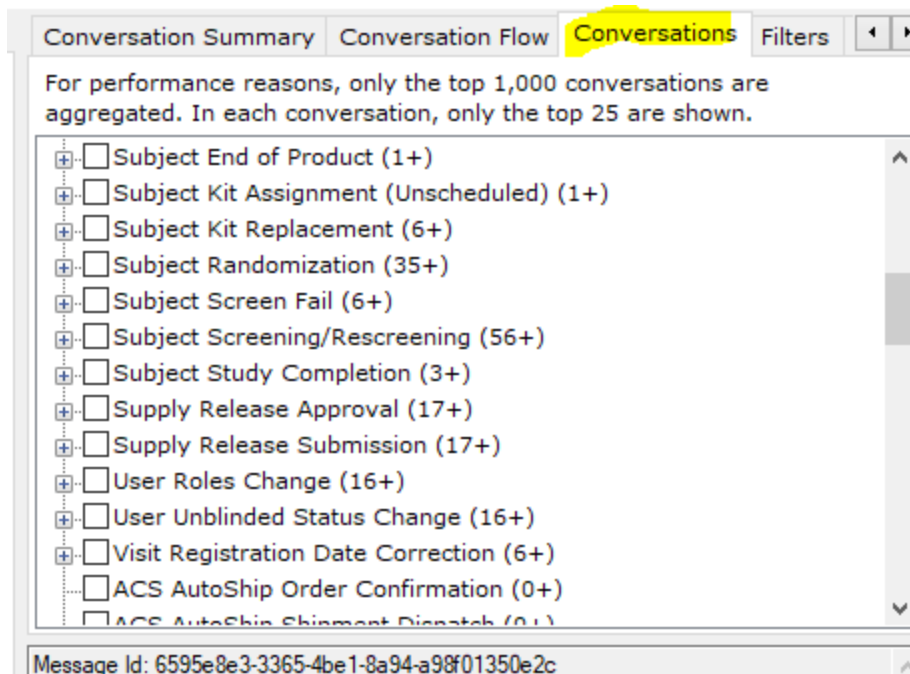
Similar to the messages list, you can see the message-level causation chain (originating and outgoing messages), but this time, you see the flow in a hierarchical manner with more context. The flow is useful to be able to:

1. Quickly see from a high level how the entire conversation structure looks like
2. Quickly determine what endpoints are involved/touched in the entire conversation
3. Quickly understand how messages are sent out in terms of overall sequential/parallel timing.
This is very useful to evaluate and/or debug potential timing issues when handlers are expecting messages to be sent/completed in a specific “logical” order.
4. Quickly see/evaluate critical payload properties of messages

Note: A very important thing to note here is that the conversation flow will always show all the messages in the conversation. For example, although you typically search/filter and find only a subset of messages **per** conversation, highlighting the conversation/message in the message search results will always pull up the entire conversation on the Conversation Flow tab. This is a very powerful feature that will allow you to quickly understand and diagnose context surrounding any individual message.

Conversations Aggregation

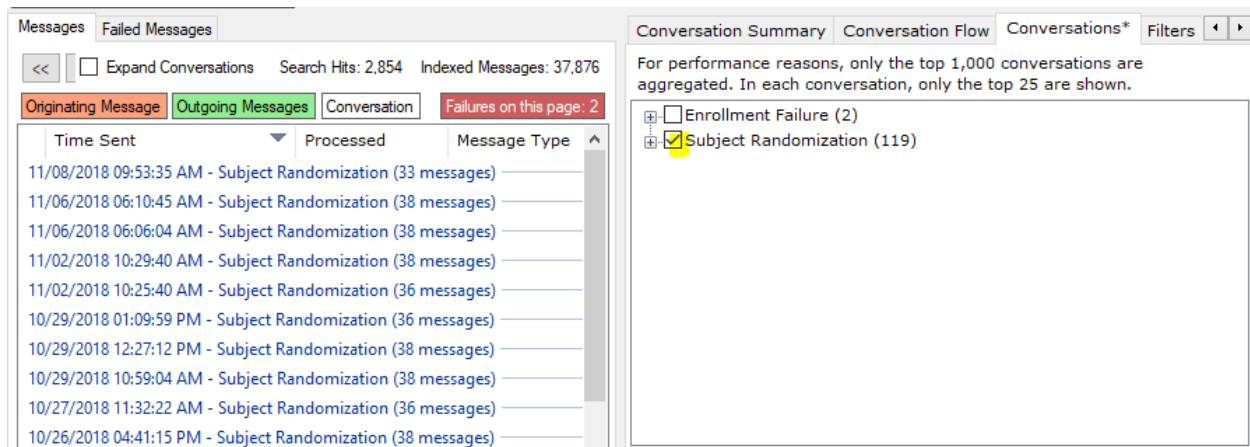
SCE aggregates the different predefined conversation types and provides a count of each type. You can see this from the Conversations tab:



From the above, you can see:

1. Multiple conversations were found from the matching search results. From the top, they are: Subject End of Product, Subject Kit Assignment, Subject Kit Replacement, etc.
2. For each conversation type, the aggregated number of conversations is displayed next to it. So for example, from above, there are at least 35 Subject Randomizations (in the message search results)
3. For each aggregation, SCE picks the top 25 conversations. If you expand the conversation type, you can see up to 25 individual conversations.

Checking a conversation will then allow you to filter and only show messages that are about that conversation. For example, clicking Subject Randomization produces this:



From above, you can see:

1. We are now narrowed down to only Subject Randomization conversations
2. In the messages results, you can see that only messages relating to Subject Randomizations are shown.

Note: It is possible to predefine conversations that contain messages that overlap. In the above example, you can see that Enrollment Failure is another predefined conversation. More importantly, you can tell that of the 119 Subject Randomizations (in the current search results), 2 of those conversations are also about Enrollment Failures.

Note: If there are multiple conversation types selected/checked, it will match all conversations that are either any of the selected conversation types (i.e. this is an OR filter condition). If you select a specific conversation (i.e. one of the children of a conversation type), it will narrow down only to that one conversation – in this case, the single selected conversation takes precedence and no other conversations will be shown in the search results.

Property Filtering

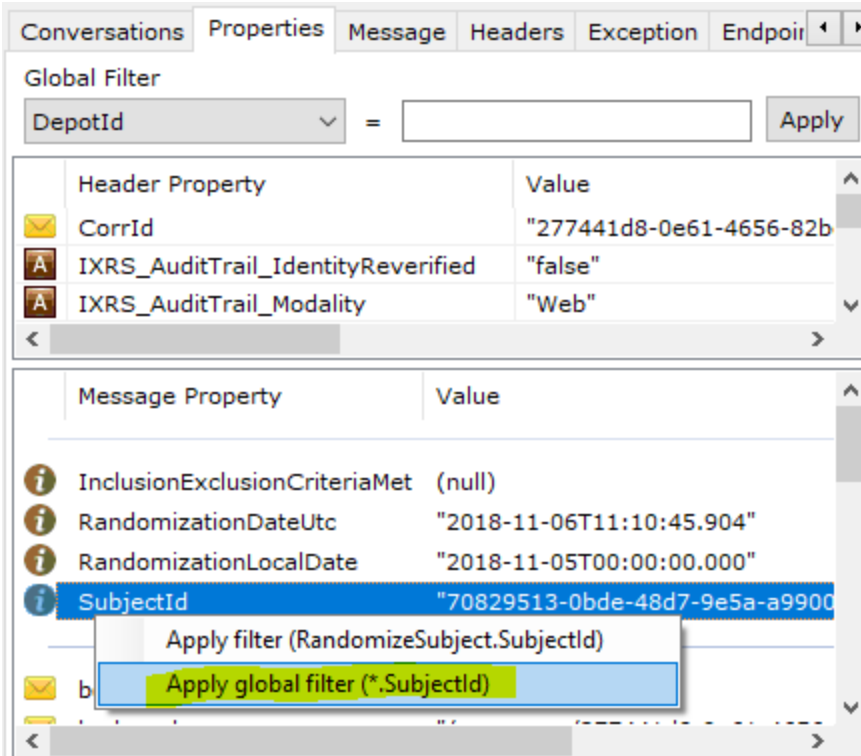
Property filtering is another powerful feature in SCE. Given any message, you can quickly narrow down your search criteria using any of the property values of the selected message. In the below example, I am currently highlighted on a RandomizeSubject message:

The screenshot displays the SCE interface with the 'Messages' tab selected. The left pane shows a list of messages, with 'RandomizeSubject' highlighted. The right pane shows the 'Properties' tab, which is divided into 'Header Property' and 'Message Property' sections.

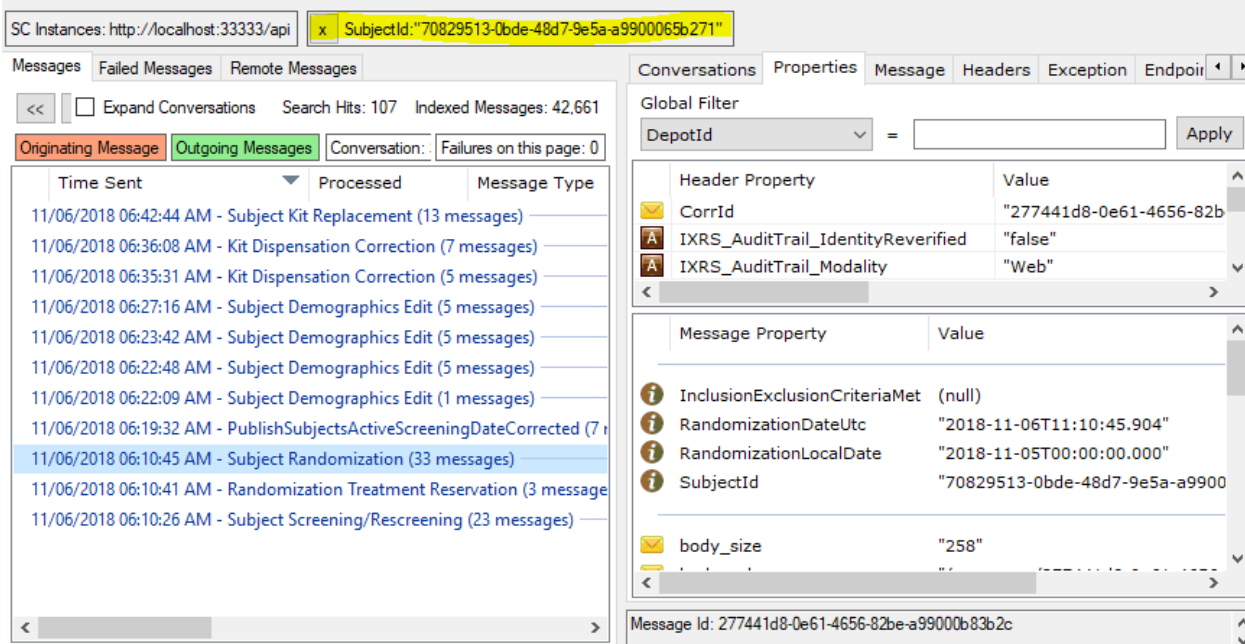
Header Property	Value
CorrId	"277441d8-0e61-4656-82b"
IXRS_AuditTrail_IdentityReverified	"false"
IXRS_AuditTrail_Modality	"Web"

Message Property	Value
InclusionExclusionCriteriaMet	(null)
RandomizationDateUtc	"2018-11-06T11:10:45.904"
RandomizationLocalDate	"2018-11-05T00:00:00.000"
SubjectId	"70829513-0bde-48d7-9e5a-a9900"
body_size	"258"

On the right is the Properties tab showing all the header and message/payload properties of this specific message. If I want to find all other messages (regardless of message type) that contain the exact same value as the above SubjectId – meaning my intention is to find what else has been happened to this specific subject, I can right click and select Apply global filter:



Which then narrows down my results to only messages/conversations about this specific subject:



From above, you can see:

1. Message search results are now returning only messages that contain the SubjectId that was filtered from the above action.

2. The SubjectId filter that is currently in-effect is shown on the top of the message search results.
3. Any time you wish to remove the current SubjectId filter in-effect, you can just click the [x] button next to it and the message search will automatically re-execute without that filter.

Using the above technique, you can also filter on any message header property. For example the below filter will allow you to narrow down your search to only to messages that originated from a specific UserId:

The screenshot shows the 'Message' tab in the search results. At the top, there is a 'Global Filter' section with a dropdown menu set to 'DepotId' and an 'Apply' button. Below this, a table lists header properties and their values. The first row is highlighted in blue, and a yellow tooltip is visible over it.

Header Property	Value
IXRS_AuditTrail_OriginatingUserId	"8cfc3987-c65c-b3b2-2e1f-f250e1c99eee"
IXRS_AuditTrail_Reason	(null)
IXRS_AuditTrail_ReferenceNumber	(null)
IXRS_AuditTrail_Sessionid	"unqp2lfawad1l0mufp3tipel"
IXRS_OriginatingAtUtc	"2018-11-06T11:10:45.000"
NServiceBus_ContentType	"application/json"

A yellow tooltip is displayed over the first row, containing the text: "Apply filter (headers.IXRS_AuditTrail_OriginatingUserId)".

Below the header properties table, there is another table for message properties:

Message Property	Value
InclusionExclusionCriteriaMet	(null)
RandomizationDateUtc	"2018-11-06T11:10:45.904"
RandomizationLocalDate	"2018-11-05T00:00:00.000"
SubjectId	"70820513-0bdc-48d7-8c53-a0000065b371"

In other words, you can use the above filter to ask SCE to show you everything that has been done in the system by this specific user.

Note: Multiple filters can be applied. As you apply them, they will all appear in the filter section above the messages. All filters are ANDed when executed. This is by design. The purpose of filters is to quickly narrow down your search based on information about a specific message/conversation that you are currently looking at.

Exceptions

Failed messages in the message search results are explicitly and visibility highlighted as follows:

Messages		Failed Messages
<<		<input type="checkbox"/> Expand Conversations Search Hits: 3 Indexed Messages: 37,876
Originating Message	Outgoing Messages	Conversations
Failures on this page: 3		
Time Sent	Processed	Message
✖ 10/26/2018 11:50:38 AM - Subject Randomization (1 messages)		
✖ 10/26/2018 11:39:27 AM - Subject Randomization (1 messages)		
✖ 10/26/2018 08:08:34 AM - SendEdcMessage (1 messages)		

From the above, you can see:

1. On this page, 3 of them are failed messages and you can see their corresponding conversations in context.

Highlighting one of the failed messages and switching to the Exception tab, you can quickly see the details about the failed message (as well as the relevant source code lines automatically extracted for you):

Messages

Failed Messages

<<

☐ Expand Conversations

Search Hits: 3

Indexed Messages: 37,876

Originating Message

Outgoing Messages

Conversations

Failures on this page: 3

Time Sent	Processed	Message
✖ 10/26/2018 11:50:38 AM - Subject Randomization (1 messages)		
✖ 10/26/2018 11:50:38.634 AM - Subject Randomization (1 messages)	11/29/2018 01:41:45.547 PM	Ki
✖ 10/26/2018 11:39:27 AM - Subject Randomization (1 messages)		
✖ 10/26/2018 08:08:34 AM - SendEdcMessage (1 messages)		

Filters

Message

Headers

Exception*

Endpoints

Excluded Messages

Source:

System.InvalidOperationException, NServiceBus.Core

Message:

No handlers could be found for message type: IXRS.TreatmentDispensing.KitsDispensed

Source lines:

LoadHandlersConnector.cs:line 31
DataBusReceiveBehavior.cs:line 63
Async.ClearSynchronizationContextBehavior.<Invoke>d__0.MoveNext()
DeserializeLogicalMessagesConnector.cs:line 33
InvokeAuditPipelineBehavior.cs:line 18

Stacktrace:

System.InvalidOperationException: No handlers could be found for message type: I
at NServiceBus.LoadHandlersConnector.<Invoke>d__1.MoveNext() in C:\BuildAge
at NServiceBus.DataBusReceiveBehavior.<Invoke>d__1.MoveNext() in C:\BuildAge
at IXRS.Async.ClearSynchronizationContextBehavior.<Invoke>d__0.MoveNext()

Message Id:

2c2a34b7-079a-4888-b81c-a9850104d09d

Time Sent:

10/26/2018 11:50:38.634 AM

Processed:

11/29/2018 01:41:45.547 PM

Body:

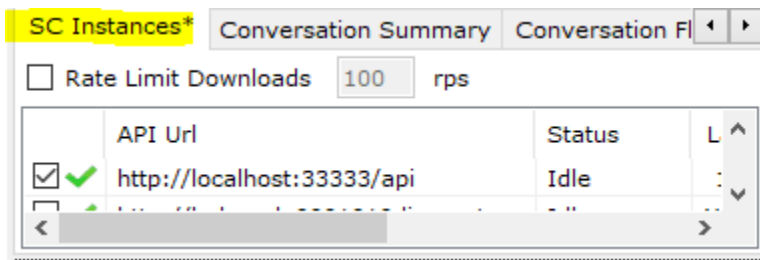
{
"SiteId": "5a0c60a3-bda5-465f-8743-a98400eebe35",
"SubjectId": "73ac60c5-09fa-4406-931e-a98500c2ac2c",
"VisitId": "6668535b-44e2-4cdb-f7c-bccd5a5c86b8",
"KitIds": [
"fc915b74-57e3-422c-b1a7-a98400eea4f6"
]
}

Note: It is very important that you look into failed messages and find the underlying cause and resolve them as soon as possible. Ignoring failed messages can possibly mean that they will go to production and eventually cause bigger problems down the line. You do not want this to happen. Ignoring failed messages will result in poor quality software.

Failed Message Search (Error Message)

You have the ability to search failed messages by

1. ServiceControl instance



2. Date/Range (against the message time_sent property)

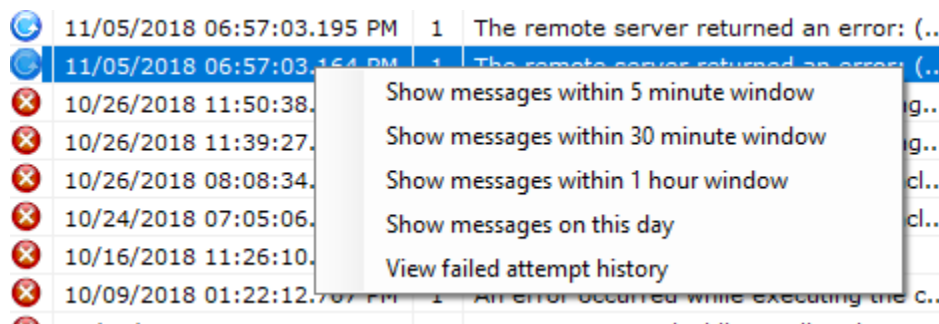
After ☒ 11/24/2018 12:00:00 AM

Before ☐ 12/01/2018 12:00:00 AM

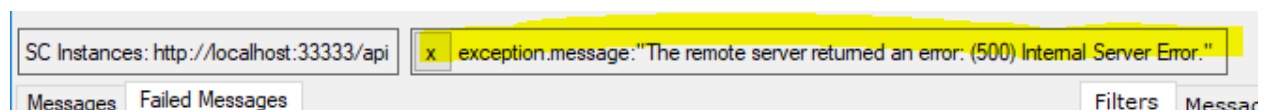
On Day ☐ 12/01/2018

Note: When SCE runs, the “After” date filter is initialized to 7 days prior to the current day, meaning, by default, you will only see messages within the past 7 days. It is important to keep this filter on all the time for performance reasons when dealing with a lot of messages.

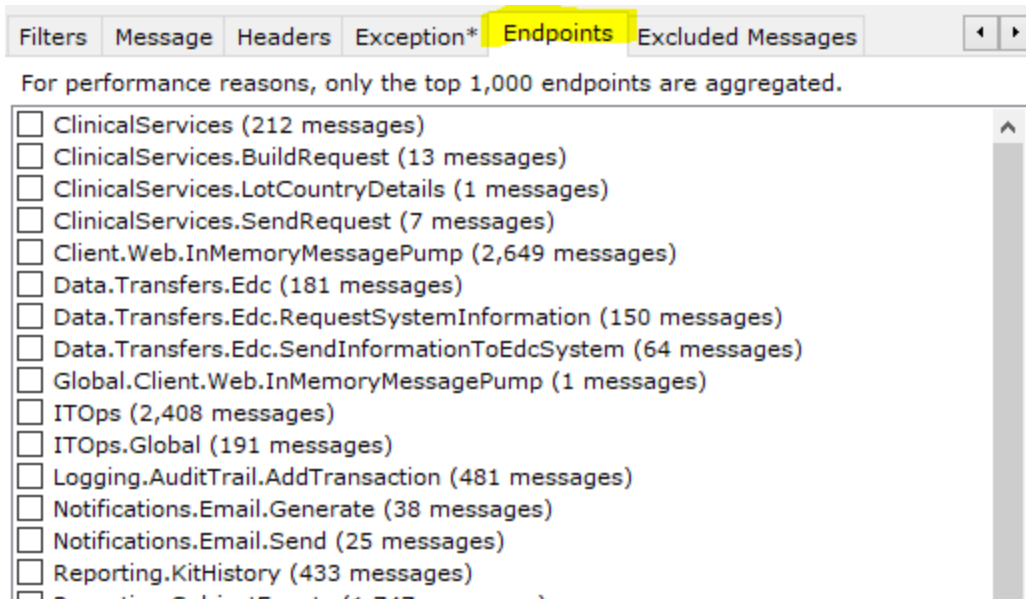
3. Time range window surrounding any single message:



4. Message property filters (can also filter by instance, or by endpoint)



5. Endpoint



6. Any arbitrary boolean/full-text query (for advanced users)

☒ Query:
[\(Syntax\)](#)

Failed Message Search Results

Failed message search results are much simpler and more straightforward than the messages:

Messages					
Failed Messages					
<< < > >> Page: 1/1			Search Hits: 682 Indexed Messages: 682		
	Time Sent	\..	Exception	Message Type	Processing E
✓	11/25/2018 04:18:35.014 PM	3	ProcessStockInformationFile failed.	ProcessStockInfor...	Roche
✓	11/25/2018 11:50:38.330 AM	4	ProcessStockInformationFile failed.	ProcessStockInfor...	Roche
🔄	11/05/2018 06:57:03.195 PM	1	The remote server returned an error: ...	PushCustomPromp...	ITOps
🔄	11/05/2018 06:57:03.164 PM	1	The remote server returned an error: ...	AssociateCorePro...	ITOps
✗	10/26/2018 11:50:38.634 AM	7	No handlers could be found for messa...	KitsDispensed	Data.Transfe
✗	10/26/2018 11:39:27.682 AM	4	No handlers could be found for messa...	KitsDispensed	Data.Transfe
✗	10/26/2018 08:08:34.994 AM	1	Valid values are between 1 and 13, in...	SendEdcMessage	Data.Transfe
✗	10/24/2018 07:05:06.646 AM	1	Valid values are between 1 and 13, in...	SendEdcMessage	Data.Transfe
✗	10/16/2018 11:26:10.454 AM	1	Sequence contains no elements	VoidKitsDispensed...	Services.Dev
✗	10/09/2018 01:22:12.707 PM	1	An error occurred while executing the ...	VisitRegistrationDa...	Services.Dev
✗	10/08/2018 04:22:36.879 PM	1	An error occurred while sending the r...	RefreshPromptsAn...	ITOps
✗	10/08/2018 04:21:44.982 PM	1	An error occurred while executing the ...	RandomizeSubject...	Services.Dev
✗	10/08/2018 07:03:22.813 AM	1	An error occurred while sending the r...	RefreshPromptsAn...	ITOps
✗	09/21/2018 07:05:34.614 AM	1	Calling 'Read' when the data reader is...	VisitRegistrationDa...	Services.Dev
✗	09/18/2018 12:06:56.846 PM	1	Sequence contains no elements	CalculateSubjectsN...	Services.Dev
✗	09/16/2018 08:49:49.184 AM	1	Sequence contains no elements	KitsAssignedAtSch...	ClinicalServi

Each failed message is shown with a corresponding status visualization (unresolved, retried, or resolved). In addition, highlighting a single message will show the raw payload as well as the detailed exception information as follows:

Messages

Failed Messages

<< < > >>

Page: 1/1

Search Hits: 682 Indexed Messages: 682

	Time Sent	\..	Exception
✓	11/25/2018 04:18:35.014 PM	3	ProcessStockInformationFile fa
✓	11/25/2018 11:50:38.330 AM	4	ProcessStockInformationFile fa
🔄	11/05/2018 06:57:03.195 PM	1	The remote server returned ar
🔄	11/05/2018 06:57:03.164 PM	1	The remote server returned ar
✗	10/26/2018 11:50:38.634 AM	7	No handlers could be found for
✗	10/26/2018 11:39:27.682 AM	4	No handlers could be found for
✗	10/26/2018 08:08:34.994 AM	1	Valid values are between 1 and
✗	10/24/2018 07:05:06.646 AM	1	Valid values are between 1 and
✗	10/16/2018 11:26:10.454 AM	1	Sequence contains no element
✗	10/09/2018 01:22:12.707 PM	1	An error occurred while execut
✗	10/08/2018 04:22:36.879 PM	1	An error occurred while sendin
✗	10/08/2018 04:21:44.982 PM	1	An error occurred while execut
✗	10/08/2018 07:03:22.813 AM	1	An error occurred while sendin
✗	09/21/2018 07:05:34.614 AM	1	Calling 'Read' when the data r
✗	09/18/2018 12:06:56.846 PM	1	Sequence contains no element
✗	09/16/2018 08:49:49.184 AM	1	Sequence contains no element
✗	09/16/2018 08:49:49.183 AM	1	Sequence contains no element
✗	09/16/2018 08:49:49.182 AM	1	Sequence contains no element
✗	09/16/2018 08:49:48.180 AM	1	Sequence contains no element
✗	09/16/2018 08:49:48.179 AM	1	Sequence contains no element
✗	09/16/2018 08:49:46.175 AM	1	Sequence contains no element
✗	09/16/2018 08:49:46.175 AM	1	Sequence contains no element
✗	09/16/2018 08:49:43.170 AM	1	Sequence contains no element
✗	09/11/2018 09:29:19.184 AM	1	An error occurred while sendin

Filters

Message

Headers

Exception*

Endpoints

Excluded Messages

Source: System.InvalidOperationException, NServiceBus.Core

Message: No handlers could be found for message type: IXRS.TreatmentDispensing.KitsDispensed

Source lines: LoadHandlersConnector.cs:line 31
DataBusReceiveBehavior.cs:line 63
Async.ClearSynchronizationContextBehavior.<Invoke>d__0.MoveNext()
DeserializeLogicalMessagesConnector.cs:line 33
InvokeAuditPipelineBehavior.cs:line 18

Stacktrace: System.InvalidOperationException: No handlers could be found for message type at NServiceBus.LoadHandlersConnector.<Invoke>d__1.MoveNext() in C:\Build\ at NServiceBus.DataBusReceiveBehavior.<Invoke>d__1.MoveNext() in C:\Build\ at IXRS.Async.ClearSynchronizationContextBehavior.<Invoke>d__0.MoveNext

Message Id: 2c2a34b7-079a-4888-b81c-a9850104d09d
Time Sent: 10/26/2018 11:50:38.634 AM
Body: {"id": "6a486283-7046-b9ba-20a2-cc22dae73332", "message_type": "IXRS.TreatmentDispensing.KitsDispensed", "time_sent": "2018-10-26T15:50:38.63449Z", "is_system_message": false, "exception": {"exception_type": "System.InvalidOperationException", "message": "No handlers could be found for message type: IXRS.TreatmentDispensing.KitsDispensed", "source": "NServiceBus.Core"}}

Similar to messages, you can apply property filters to failed messages like so:

Messages
Failed Messages
Remote Messages

<< < > >>
Page: 1 / 380
☐ Expand Conversations
Search Hits: 37,967

Note: Only a query filter is applicable here. All other filters are ignored.

Originating Message
Outgoing Messages

Failures on this page: 0

Time Sent	Processed	Message Type
12/05/2018 09:25:44 AM	- CreateShipmentMonitorUploadedEvent (3 messages)	
12/05/2018 09:23:39 AM	- EnsureAutomatedSupplyOrderingHasBeenSet (1 messages)	
12/05/2018 09:23:39 AM	- DepotSupplyLevelEvaluationIsDue (2 messages)	
12/05/2018 09:23:39 AM	- TimeoutState (2 messages)	
12/05/2018 09:23:39 AM	- EnsureConfiguredBlockAssignmentStrategyExists (1 messages)	
12/04/2018 04:16:59 PM	- CreateShipmentMonitorUploadedEvent (3 messages)	
12/04/2018 04:12:16 PM	- CreateShipmentMonitorUploadedEvent (3 messages)	

From here, you will be able to:

1. Perform simple keyword searches against the messages
2. Perform paging as well as jump directly to any page
3. Contextually view the conversation flow of any selected conversation

Since the messages are queried remotely ad-hoc and for performance reasons, this feature will not be as rich compared to having all the messages downloaded up front. In other words, search is limited to keywords only and the results that will be shown in SCE are limited to small pages at a time (default of 100 messages per page). In addition, the conversation flow will only show the limited messages that are downloaded within a page at a time.

Disabling Local Indexing

If you do not wish to download/index any ServiceControl messages into Elasticsearch, you can completely disable local indexing. To do this, shutdown SCE, edit the ServiceControlExplorer.json file and change the property `localIndexingEnabled=false`, and restart SCE. From this point on, no local message searches/indexing is allowed. The only thing you can do is run searches against a remote ServiceControl instance.