

Article

[Michael Breen](#) · Nov 9, 2016 5m read

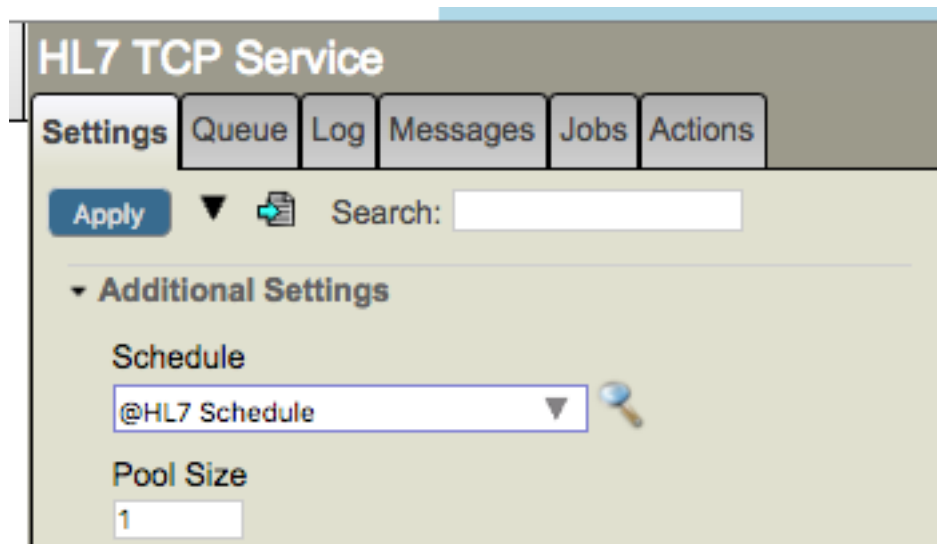
How the Ensemble Scheduler Works

The Ensemble Scheduler is used for automatically turning on and off business hosts at certain dates and times. You could use it if, for example, you wanted to only run a business host from 9am to 5pm every day. Conversely, if you want to trigger an event to occur at a specific time, for example, a job running at 1am to batch up and send off all the previous day's transactions in one file, we recommend other methods such as the Task Manager.

This article explains how the Ensemble Scheduler works. The screenshots below are from a test production with 2 scheduled components created to demonstrate how the Ensemble Scheduler can be used to manage the production's processes over time. Here's the Production Configuration showing two items: HL7 TCP Service and FTP Passthrough Operation.



Look at the Schedule by clicking on the little magnifying glass to the right of the schedule setting:



You can see that the HL7 TCP Service is scheduled to start at 9am and stop at 5pm:

NewSave AsDelete

Choose a schedule spec to work on: HL7 Schedule

Schedule Spec Editor

Raw String

START:*-*-*T09:00:00,STOP:*-*-*T17:00:00

PreviewSave SpecCancel

Description

Save Description

Deployable

Action Table

Action	Event Specification
START	*-*-*T09:00:00
STOP	*-*-*T17:00:00

Add Action

Detailed View of Current Schedule

September 2016

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Display Settings

Interval

30m

Start Time

6am

End Time

6pm

«	Day	Week	Month	»		
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
2016-09-18	2016-09-19	2016-09-20	2016-09-21	2016-09-22	2016-09-23	2016-09-24
6:00am	6:00am	6:00am	6:00am	6:00am	6:00am	6:00am
6:30am	6:30am	6:30am	6:30am	6:30am	6:30am	6:30am
7:00am	7:00am	7:00am	7:00am	7:00am	7:00am	7:00am
7:30am	7:30am	7:30am	7:30am	7:30am	7:30am	7:30am
8:00am	8:00am	8:00am	8:00am	8:00am	8:00am	8:00am
8:30am	8:30am	8:30am	8:30am	8:30am	8:30am	8:30am
9:00am	9:00am	9:00am	9:00am	9:00am	9:00am	9:00am
5:00pm	5:00pm	5:00pm	5:00pm	5:00pm	5:00pm	5:00pm
5:30pm	5:30pm	5:30pm	5:30pm	5:30pm	5:30pm	5:30pm

Likewise, if we look at the schedule for the FTP Passthrough Operation, we will see that it is also scheduled to start at 9am, but it stops at 3pm:

New Save As Delete

Choose a schedule spec to work on: FTP Schedule

Schedule Spec Editor

Raw String

START:*-*T09:00:00,STOP:*-*T15:00:00

Preview Save Spec Cancel

Description

Save Description

Deployable

☐

Action Table

Action	Event Specification
START	*-*T09:00:00
STOP	*-*T15:00:00

Add Action

Detailed View of Current Schedule

September 2016

S	M	T	W	T	F	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Display Settings

Interval 30m

Start Time 6am

End Time 6pm

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
2016-09-18	2016-09-19	2016-09-20	2016-09-21	2016-09-22	2016-09-23	2016-09-24
6:00am	6:00am	6:00am	6:00am	6:00am	6:00am	6:00am
6:30am	6:30am	6:30am	6:30am	6:30am	6:30am	6:30am
7:00am	7:00am	7:00am	7:00am	7:00am	7:00am	7:00am
7:30am	7:30am	7:30am	7:30am	7:30am	7:30am	7:30am
8:00am	8:00am	8:00am	8:00am	8:00am	8:00am	8:00am
8:30am	8:30am	8:30am	8:30am	8:30am	8:30am	8:30am
9:00am	9:00am	9:00am	9:00am	9:00am	9:00am	9:00am
3:00pm	3:00pm	3:00pm	3:00pm	3:00pm	3:00pm	3:00pm
3:30pm	3:30pm	3:30pm	3:30pm	3:30pm	3:30pm	3:30pm
4:00pm	4:00pm	4:00pm	4:00pm	4:00pm	4:00pm	4:00pm
4:30pm	4:30pm	4:30pm	4:30pm	4:30pm	4:30pm	4:30pm

The Ensemble scheduler starting and stopping a component is not the same as enabling or disabling the component. The Ensemble scheduler only starts and stops one or more operating system process (depending on Pool Size) for that component - it does not enable and disable the component. A component must be enabled for it to run according to its schedule. A disabled component will never run, regardless of its schedule.

To see if a scheduled component is started, select the component on the production configuration page and select the Jobs tab. If the process is started you will see a table showing its operating system process ID(s) in the Jobs panel on the right. Here, the Pool Size for the FTP Operation is 1, so there is only one process ID listed (14318 in the below example). If Pool Size were larger, you 'd see that many processes listed:

Legend Production Settings

Operations

FTP Passthrough Operation

FTP Passthrough Operation

Settings Queue Log Messages Jobs Actions

Go To Jobs

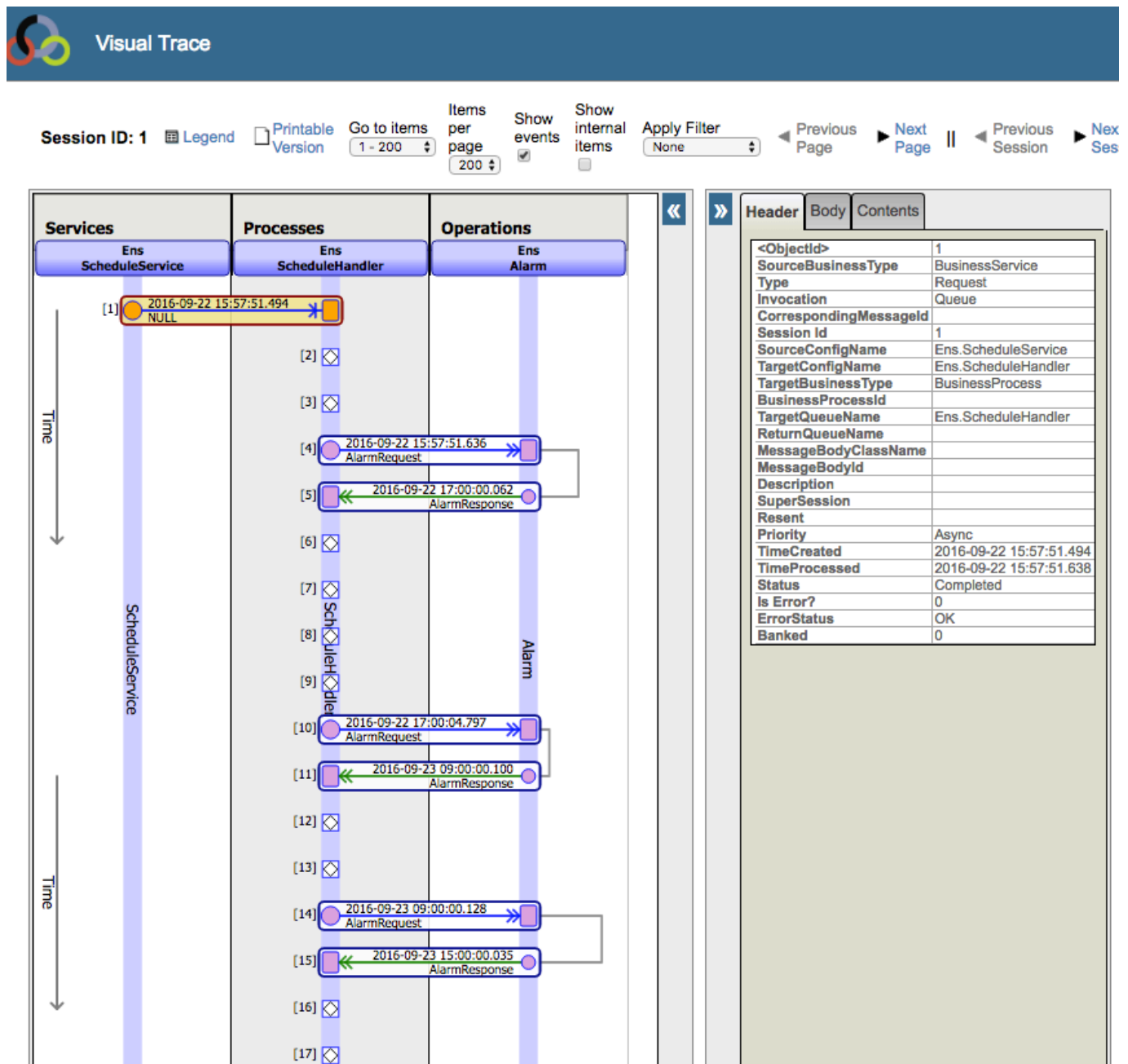
Abort Suspend Stop

Job	Status	Adapter	Retry	Message
14318	OK			

Note that the green color of the component shows whether it 's enabled or disabled, but doesn 't indicate whether it started. If the FTP Passthrough Operation was stopped by the Ensemble scheduler (for example, just after 3pm), you 'd still see the Operation showing a green circle, but there would be no processes listed under the Jobs tab:



To see how the Ensemble scheduler works “under the hood”, look at the Message Viewer and you’ll see a session that starts with `Ens.ScheduleService`. It sends a message to `Ens.ScheduleHandler` and from that point on, messages between `Ens.ScheduleHandler` and `Ens.Alarm` and back again. This back-and-forth keeps going for as long as the production is running.



This message exchange is the mechanism behind the Ensemble Scheduler.

- When the session starts, Ens.ScheduleService sends Ens.ScheduleHandler a message.
- Ens.ScheduleHandler calls `##class(Ens.Director).UpdateProduction()`.
- UpdateProduction looks at the state of the production and the schedule.
 - If the schedule says that a component is supposed to be stopped (not running) at that time, it kills the process(es) that component is running.
 - If the schedule says that a component is supposed to be started (running), it starts the appropriate number of processes (as specified by Pool Size).
- After UpdateProduction() completes, all the components should have the correct number of processes running (according to the schedule). Ens.ScheduleHandler has completed starting and stopping processes

After the call to UpdateProduction(), Ens.ScheduleHandler determines the next time it needs to run. The scheduler:

1. Collects the list of scheduled START and STOP times and

2. Finds the next scheduled transition time (whether it be a START or a STOP).
3. Sends a message to Ens.Alarm
4. Waits for a response

In the above example, Ens.ScheduleHandler collected these times:

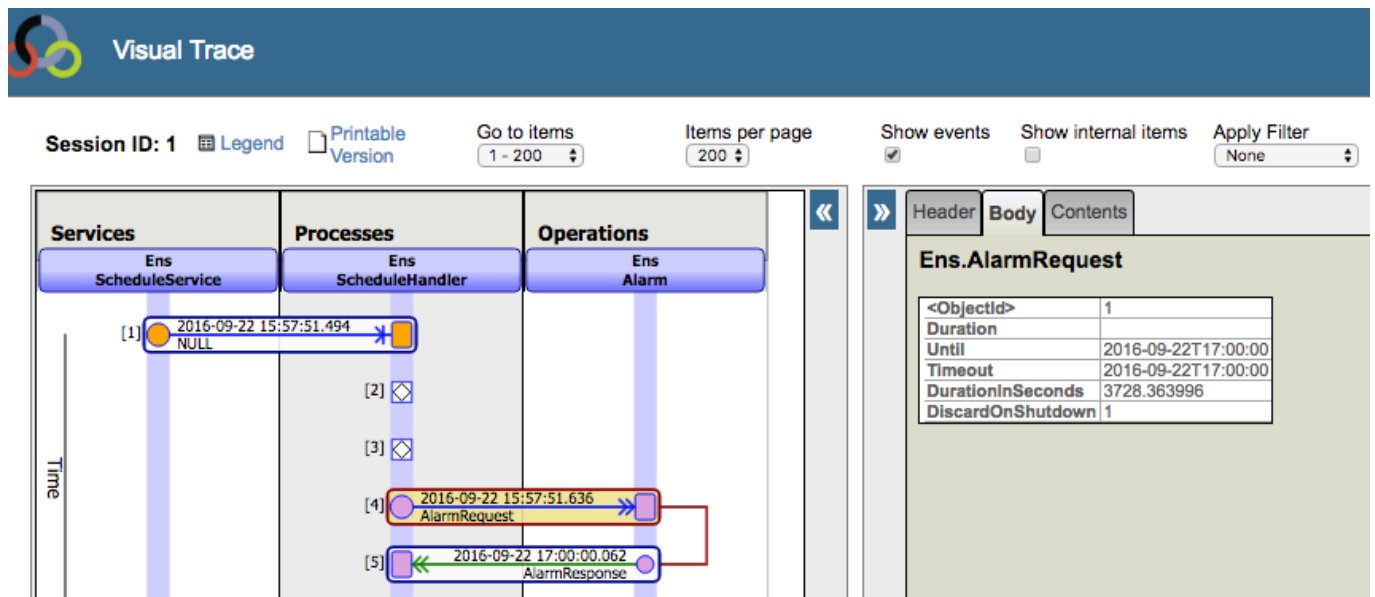
9:00 am : HL7 TCP Service START

9:00 am : FTP Passthrough Operation START

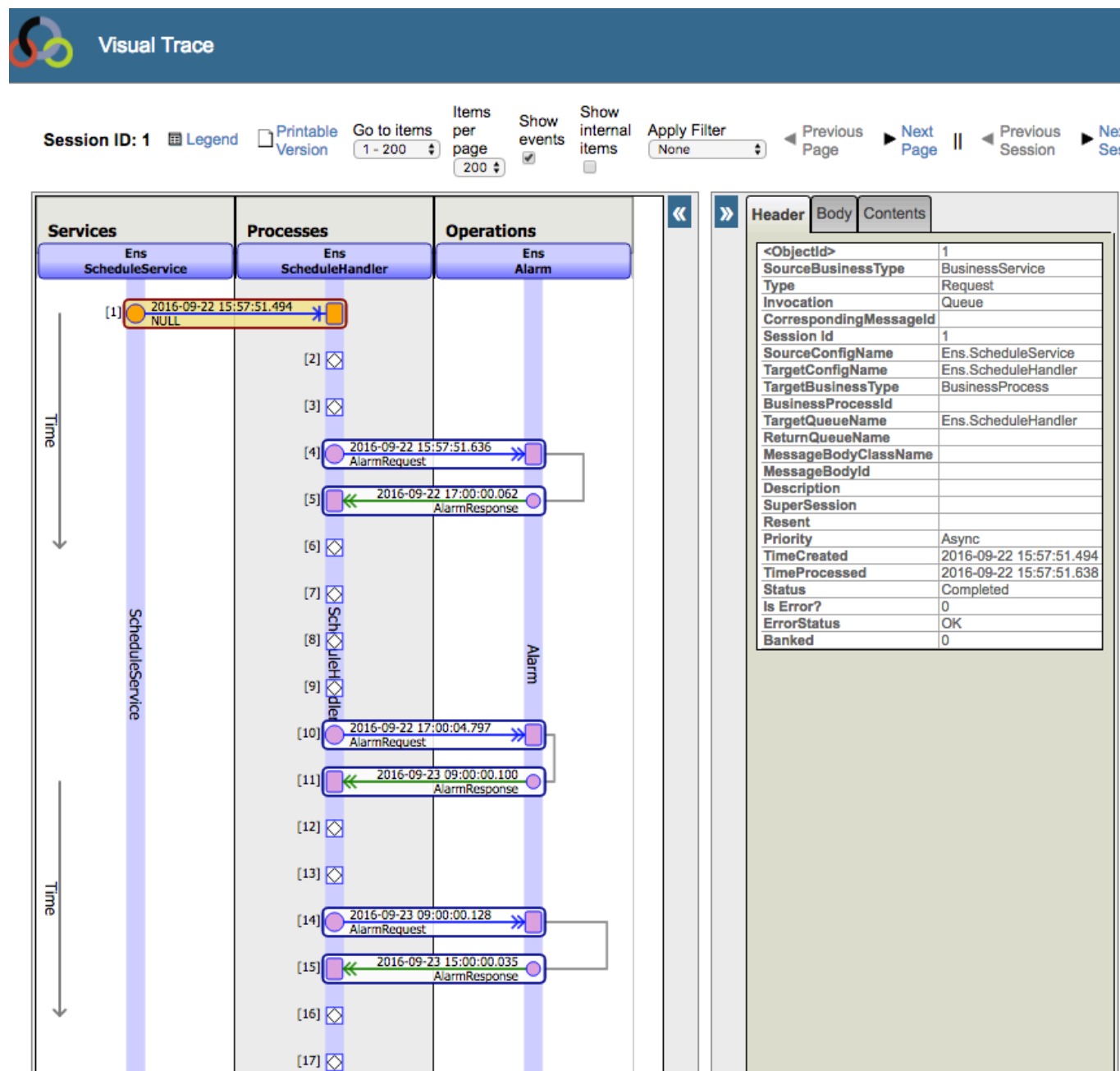
3:00 pm : FTP Passthrough Operation STOP

5:00 pm : HL7 TCP Service STOP

If you look carefully at the Visual Trace above you can see the timestamp that shows when Ens.ScheduleHandler was running: 15:57:51, a.k.a. 3:57pm. At that time, the next scheduled time is 5pm, so it sends a message to Ens.Alarm which essentially says "Wake me back up at 5pm" :



Ens.Alarm waits until 5pm and then sends its response. Above the "Alarm Response" you can see the timestamp: 17:00:00 which is the time Ens.Alarm responded. This response message triggers Ens.ScheduleHandler to call UpdateProduction() and the process starts over. It starts and stops the appropriate processes. In this case, since it's 5pm, the HL7 TCP Service is scheduled to be stopped.



If you click the Trace items marked by the white diamonds labeled [6], [7], [8] and [9] you 'd see in the right panel:

[6] Production 'User.SchedulerTest' updating...

[7] Stopping job '9545' (HL7 TCP Service)

[8] ConfigItem 'HL7 TCP Service' (EnsLib.HL7.Service.TCPService) stopped in job 9545

[9] Production 'User.SchedulerTest' updated.

Header	Body	Contents
ID:	21	
Type:	Info	
Text:	Production 'User.SchedulerTest' updating...	
Logged:	2016-09-22 17:00:00.065	
Source:	Ens.ScheduleHandler	
Session:	1	
Job:	9480	
Class:	Ens.Director	
Method:	UpdateProduction	
Trace:	(none)	
Stack:		

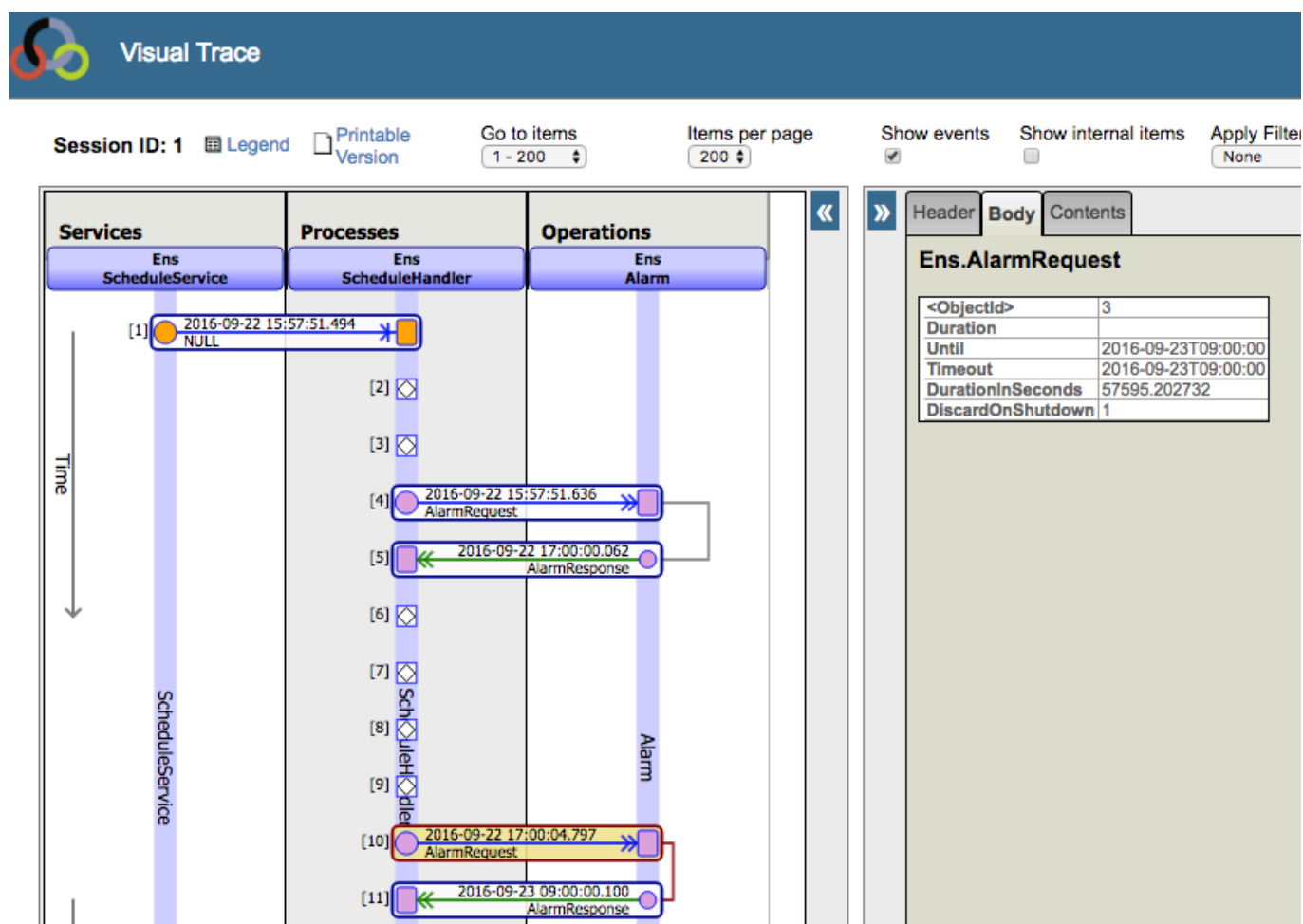
Header	Body	Contents
ID:	22	
Type:	Info	
Text:	Stopping job '9545' (HL7 TCP Service)	
Logged:	2016-09-22 17:00:00.075	
Source:	Ens.ScheduleHandler	
Session:	1	
Job:	9480	
Class:	Ens.Job	
Method:	Stop	
Trace:	(none)	
Stack:		

Header	Body	Contents
ID:	23	
Type:	Info	
Text:	ConfigItem 'HL7 TCP Service' (EnsLib.HL7.Service.TCPService) stopped in job 9545	
Logged:	2016-09-22 17:00:04.796	
Source:	Ens.ScheduleHandler	
Session:	1	
Job:	9480	
Class:	Ens.Director	
Method:	actualizeProductionDifferences	
Trace:	(none)	
Stack:		

If there were more processes to stop, you ' d see a pair of white trace diamonds similar to [7] and [8] for every process the scheduler stops.

Next, Ens.ScheduleHandler looks at the scheduled START and STOP times and determines that 9:00am is the next time it needs to take action, so it sends a message to Ens.Alarm requesting it wake Ens.ScheduleHandler up at 9:00am.

The sequence of steps below, where it starts processes instead of stopping them, is largely the same as above, with the exception of a few missing trace elements as described below.



Ens.Alarm waits until 9:00am and then sends back a response. Ens.ScheduleHandler sees that it 's after 9:00am and that it needs to start both the HL7 TCP Service and the FTP Passthrough Operation. The Visual Trace doesn ' t show Trace elements showing that the components are being started because the components themselves do the logging as opposed to Ens.ScheduleHandler, but you can see it in the Event Log:

Info	30	2016-09-23 09:00:00.345		24441	FTP Passthrough Operation	ConfigItem 'FTP Passthrough Operation' (EnsLib.FTP.PassthroughOperation) started in job 24441
Info	28	2016-09-23 09:00:00.134		24442	HL7 TCP Service	ConfigItem 'HL7 TCP Service' (EnsLib.HL7.Service.TCPService) started in job 24442
Info	26	2016-09-23 09:00:00.127	1	9480	Ens.ScheduleHandler	Production 'User.SchedulerTest' updated.
Info	25	2016-09-23 09:00:00.104	1	9480	Ens.ScheduleHandler	Production 'User.SchedulerTest' updating...

The messages continue to pass back and forth between the Ens.ScheduleHandler and the Ens.Alarm until the production stops (or until someone removes the schedules).

Since the scheduler uses Ensemble messages, you must configure the purge task so that it does not purge the scheduler message. The safest way to avoid accidentally purging scheduler messages is to always select the Keep Integrity option in the Ens.Util.Tasks.Purge task (see screenshot below). Keep Integrity specifies that the task should only purge messages that are " complete". Since the scheduler session is never complete, it should not be purged, but setting " Keep Integrity " to false will override this safeguard. For more information on the " Keep Integrity " setting, see <https://community.intersystems.com/post/why-keepintegrity-important-when-purging-healthshareensemble-data>

Task Scheduler Wizard

This wizard helps to you schedule a task for execution by the Task Manager or to edit the details of a previously scheduled task. For user-defined tasks you must first create a new subclass of the %SYS.Task.Definition class which will then be selectable as a 'Task type'.

Task name: * Message Purge

Description:

Namespace to run task in: ENSEMBLE

Task type: * Ens.Util.Tasks.Purge

BodiesToo ☒

KeepIntegrity ☒

NumberOfDaysToKeep 7

TypesToPurge All Types

Task priority: Priority Normal

Run task as this user: UnknownUser

Open output file when task is running? No

Output file: Browse...

Suspend task on error? No

Reschedule task after system restart? No

If you have any questions or problems with the scheduler, please call the WRC at 617-621-0700 or email us at support@intersystems.com

[#System Administration](#) [#Ensemble](#)

Source URL: <https://community.intersystems.com/post/how-ensemble-scheduler-works>