Artículo
Kurro Lopez · 15 dic, 2021 Lectura de 6 min

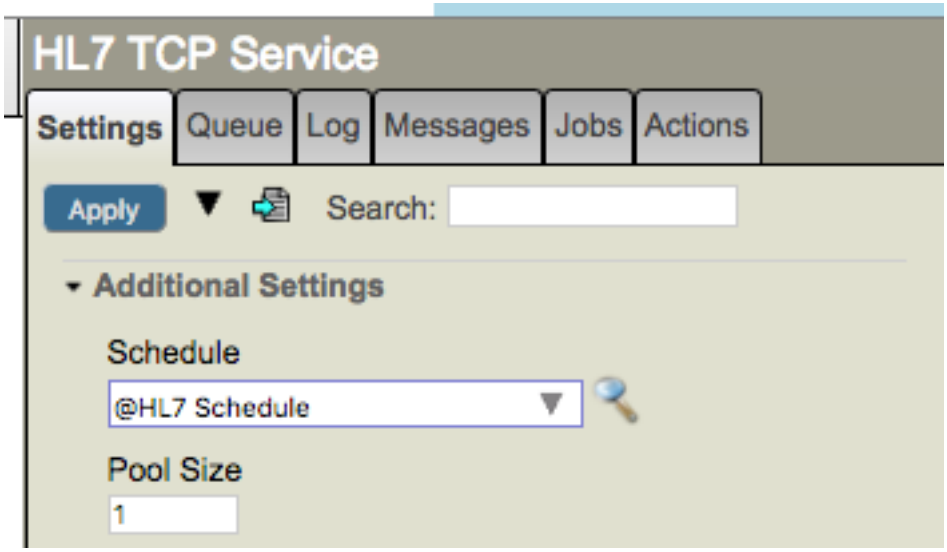
Como funciona la planificación de Ensemble

El Programador de Ensemble se utiliza para encender y apagar automáticamente los hosts en determinadas fechas y horas. Podrías usarlo si, por ejemplo, solo quisieras ejecutar un host de negocios de 9:00 a 17:00 todos los días. Por el contrario, si desea activar un evento para que ocurra en un momento específico, por ejemplo, un trabajo que se ejecuta a la 01:00, para agrupar y enviar todas las transacciones del día anterior en un archivo, recomendamos otros métodos como el Administrador de tareas.

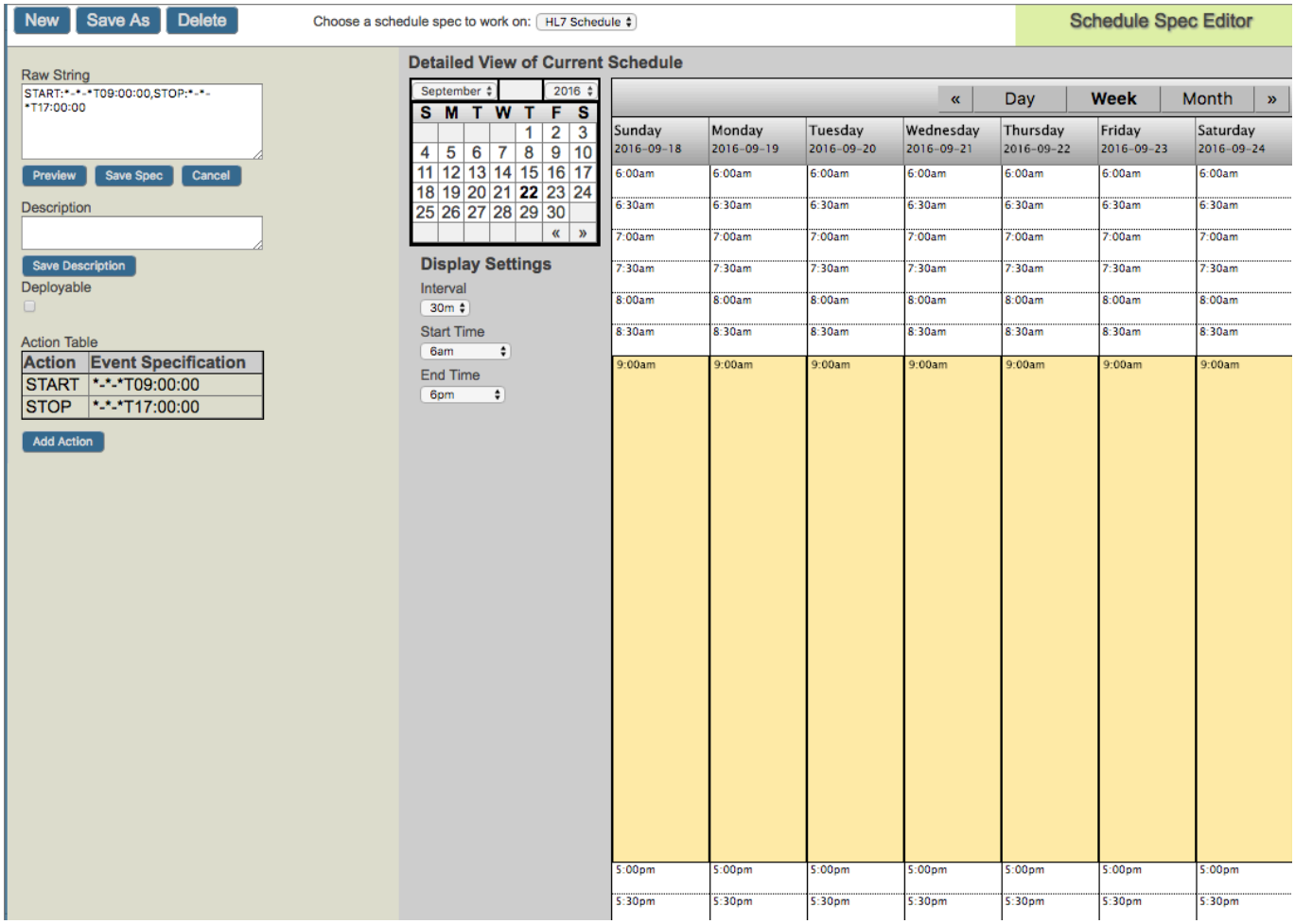
Este artículo explica cómo funciona el Programador de Ensemble. Las capturas de pantalla a continuación son de una producción de prueba con 2 componentes programados creados para demostrar cómo se puede usar el Programador de Ensemble para administrar los procesos de producción a lo largo del tiempo. Aquí está la configuración de producción que muestra dos elementos: servicio TCP HL7 y operación Passthrough de FTP.



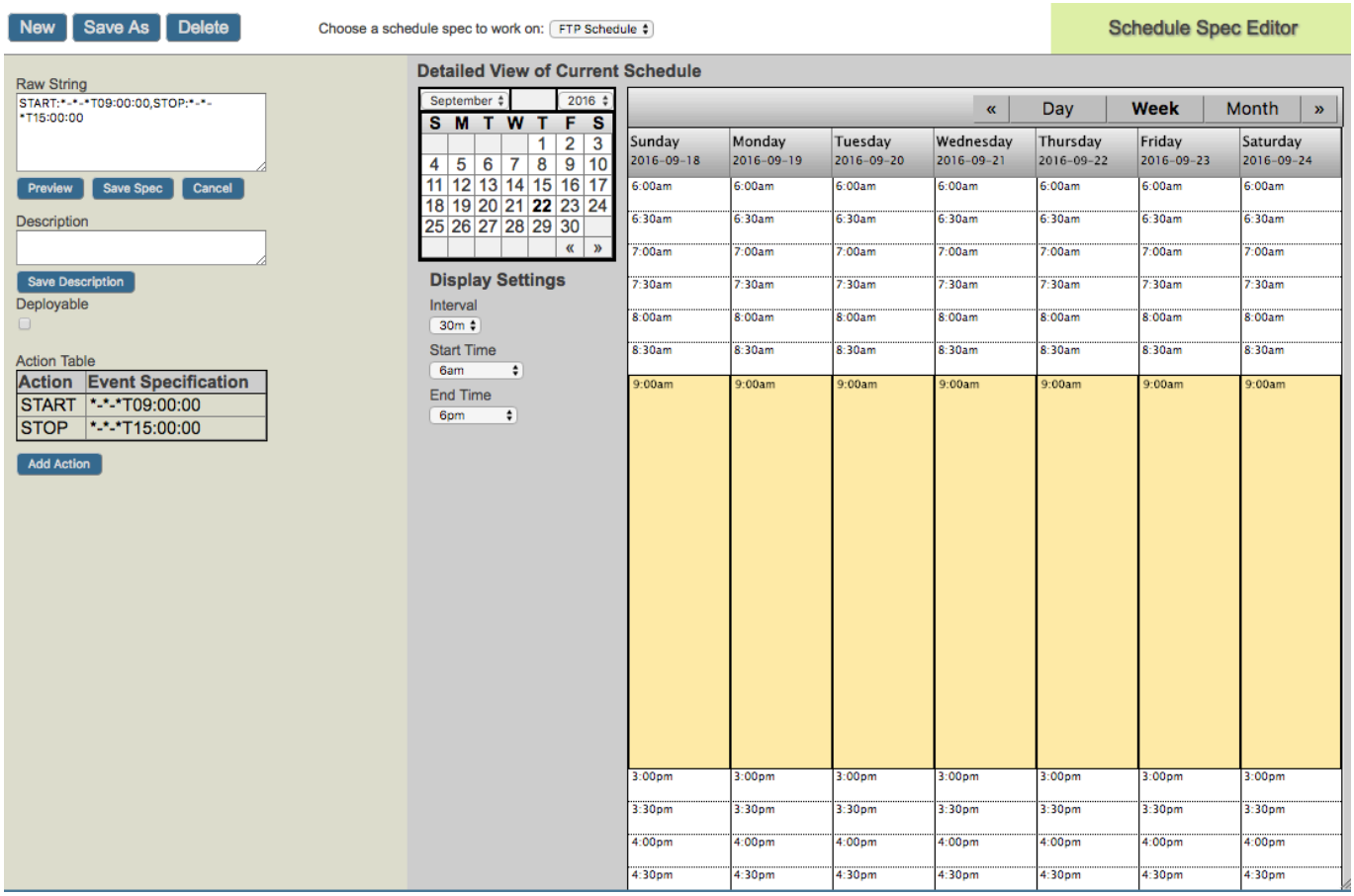
Abre el programa (schedule) haciendo clic en la pequeña lupa a la derecha de la configuración del programa:



Puede ver que el servicio HL7 TCP está programado para comenzar a las 09:00 y terminar a las 17:00:

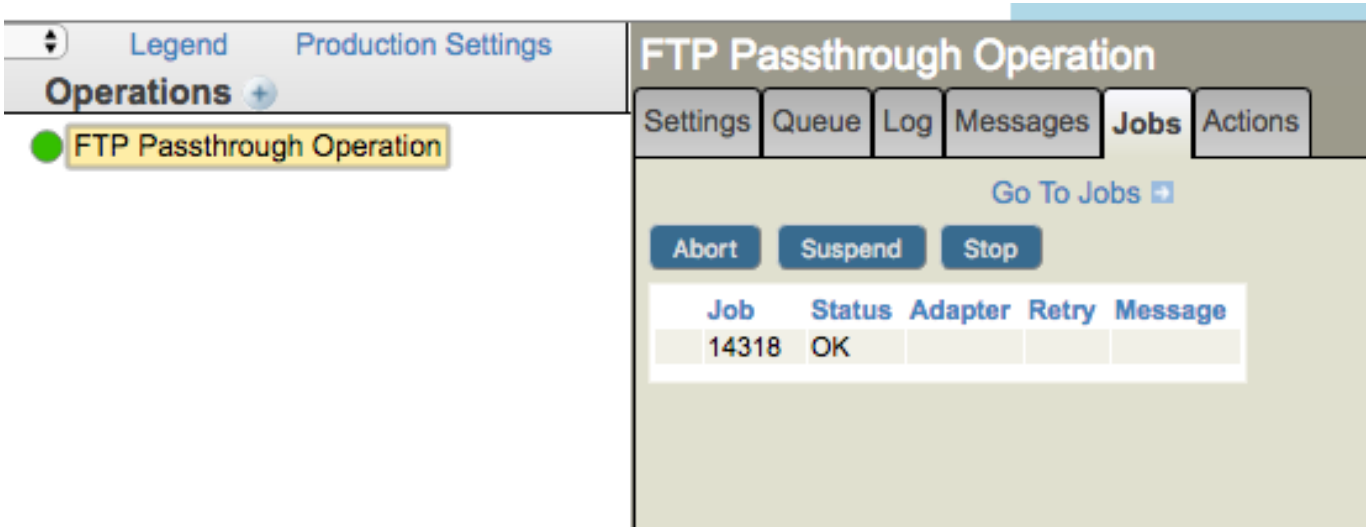


Asimismo, si nos fijamos en el cronograma de la operación FTP Passthrough, veremos que también está programado para comenzar a las 09:00, pero se detiene a las 15:00:



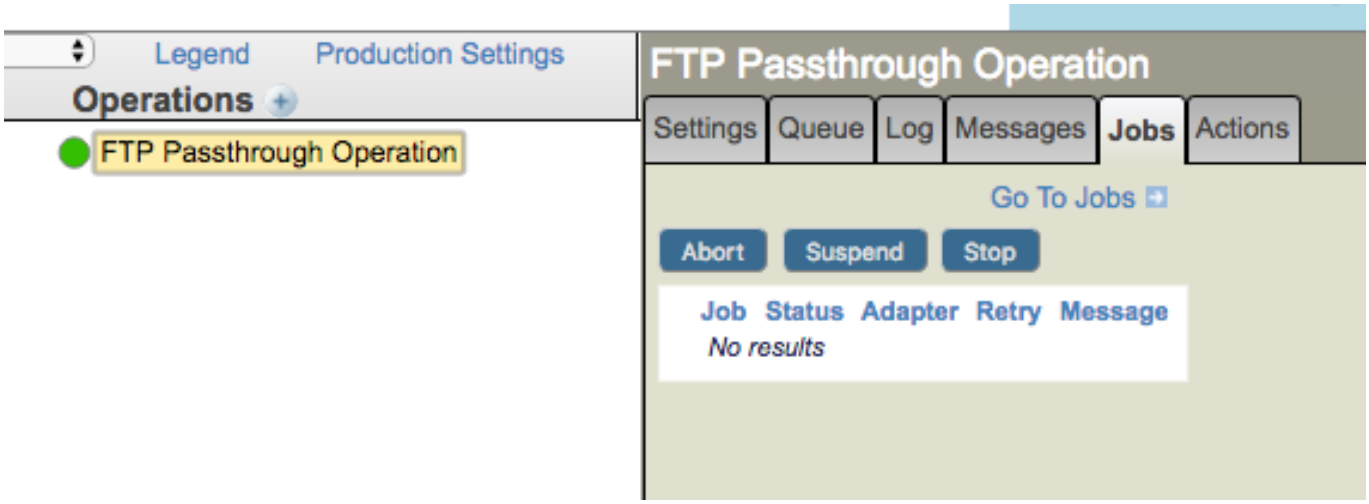
El planificador de Ensemble que inicia y detiene un componente no es lo mismo que habilitar o deshabilitar el componente. El programador de Ensemble solo inicia y detiene uno o más procesos del sistema operativo (según el tamaño del grupo) para ese componente; no habilita ni deshabilita el componente. Un componente debe estar habilitado para que se ejecute de acuerdo con su programación. Un componente deshabilitado nunca se ejecutará, independientemente de su programación.

Para ver si se inicia un componente programado, seleccione el componente en la página de configuración de producción y seleccione la pestaña "Procesos". Si se inicia el proceso, verá una tabla que muestra los ID de proceso de su sistema operativo en el panel "Procesos" a la derecha. Aquí, el tamaño del grupo para la operación FTP es 1, por lo que solo hay un ID de proceso en la lista (14318 en el ejemplo siguiente). Si el tamaño de la agrupación fuera mayor, vería que se enumeran muchos procesos:

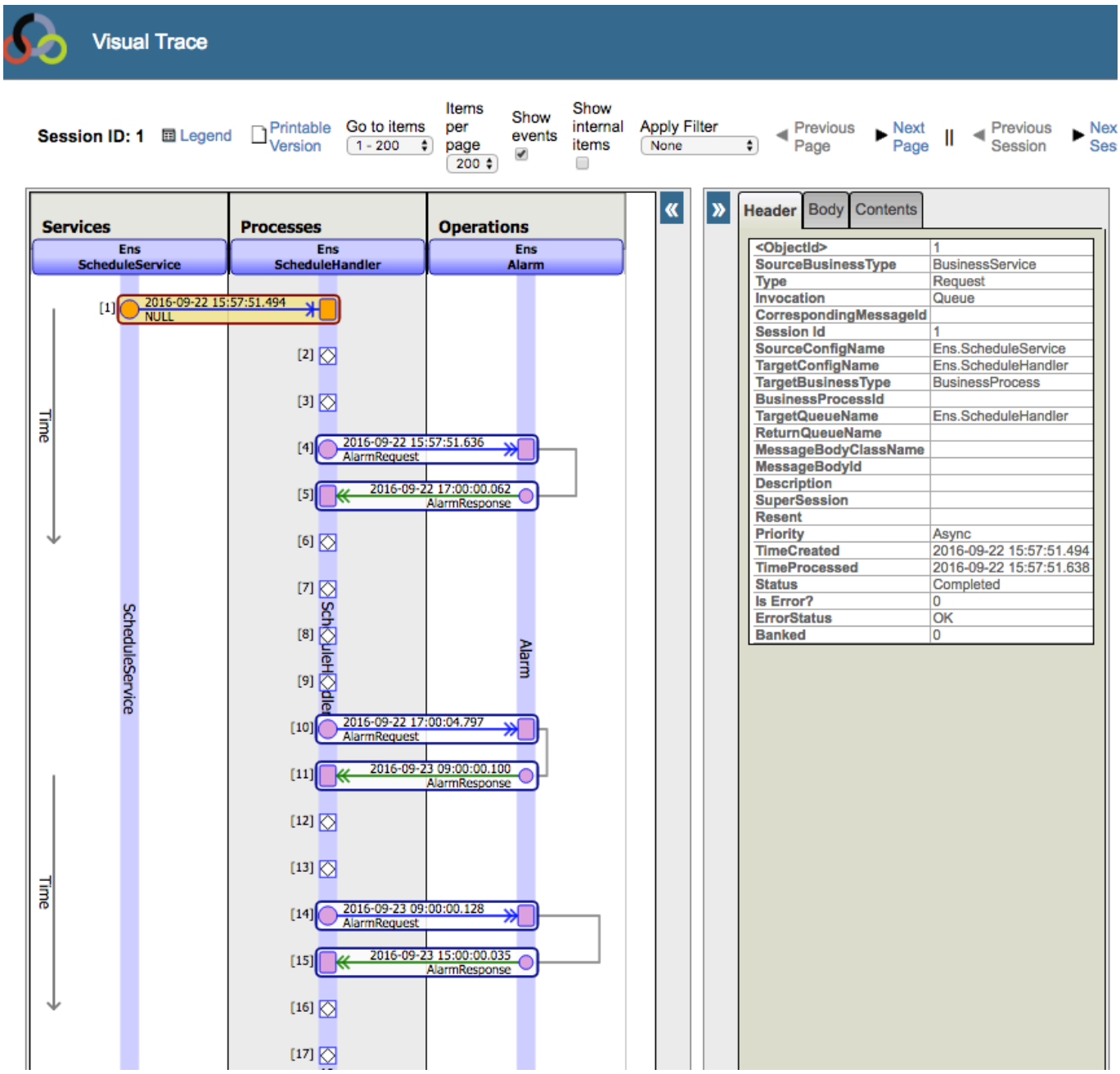


Tenga en cuenta que el color verde del componente muestra si está habilitado o inhabilitado, pero no indica si se inició. Si el programador de Ensemble detuvo la operación de paso a través de FTP (por ejemplo, justo después

de las 15:00), Aún verá la operación mostrando un círculo verde, pero no habrá procesos listados en la pestaña "Procesos":



Para ver cómo funciona el programador de Ensemble "bajo el capó", mire el Visor de mensajes y verá una sesión que comienza con Ens.ScheduleService. Envía un mensaje a Ens.ScheduleHandler y, a partir de ese momento, mensajes entre Ens.ScheduleHandler y Ens.Alarm y viceversa. Este ir y venir continúa mientras la producción está en marcha.



Este intercambio de mensajes es el mecanismo detrás del Planificador de Ensenble.

- Cuando comienza la sesión, Ens.ScheduleService envía un mensaje a Ens.ScheduleHandler.
- Ens.ScheduleHandler llama a la clase ##(Ens.Director).UpdateProduction().
- UpdateProduction analiza el estado de la producción y el cronograma.
    - Si el cronograma dice que se supone que un componente debe estar detenido (no ejecutándose) en ese momento, mata los procesos que ese componente está ejecutando.
    - Si el programa dice que se supone que un componente debe iniciarse (en ejecución), inicia el número apropiado de procesos (según lo especificado por Pool Size).
- Una vez que se completa UpdateProduction(), todos los componentes deben tener la cantidad correcta de procesos en ejecución (de acuerdo con la programación). Ens.ScheduleHandler ha completado los procesos de inicio y detención

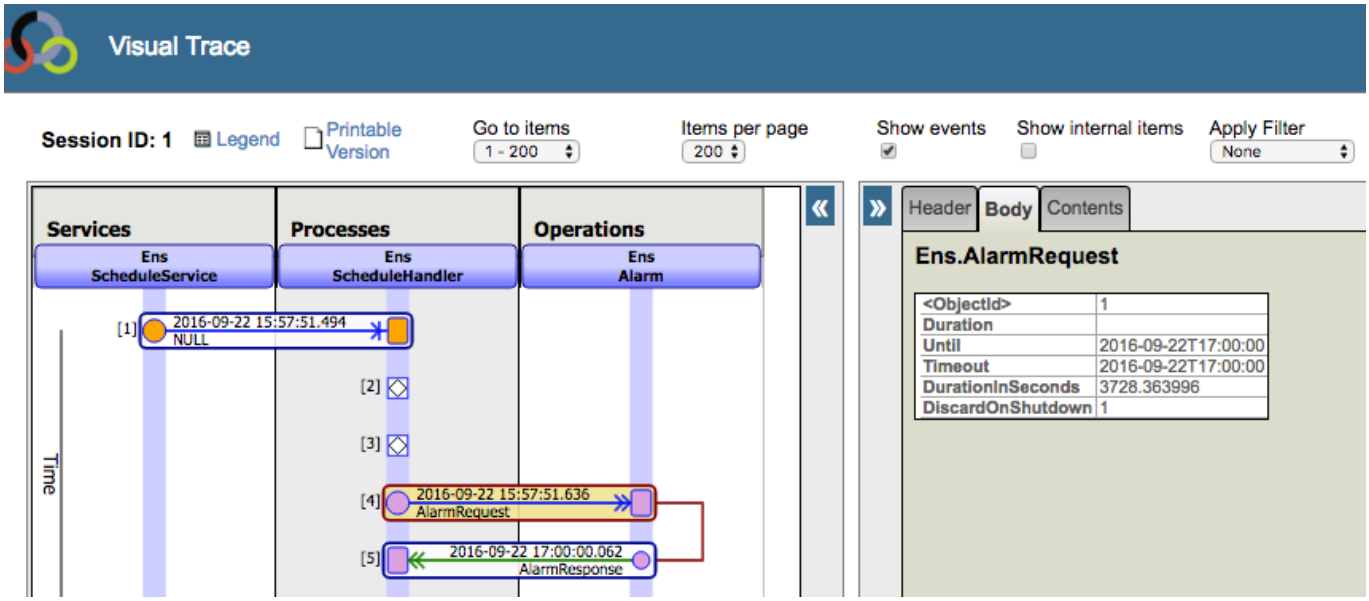
Después de la llamada a UpdateProduction(), Ens.ScheduleHandler determina la próxima vez que debe ejecutarse. La planificación:

1. Recopila la lista de horas de START y STOP programadas y
2. Encuentra el próximo tiempo de transición programado (ya sea un INICIO o una PARADA).
3. Envía un mensaje a Ens.Alarm
4. Espera una respuesta

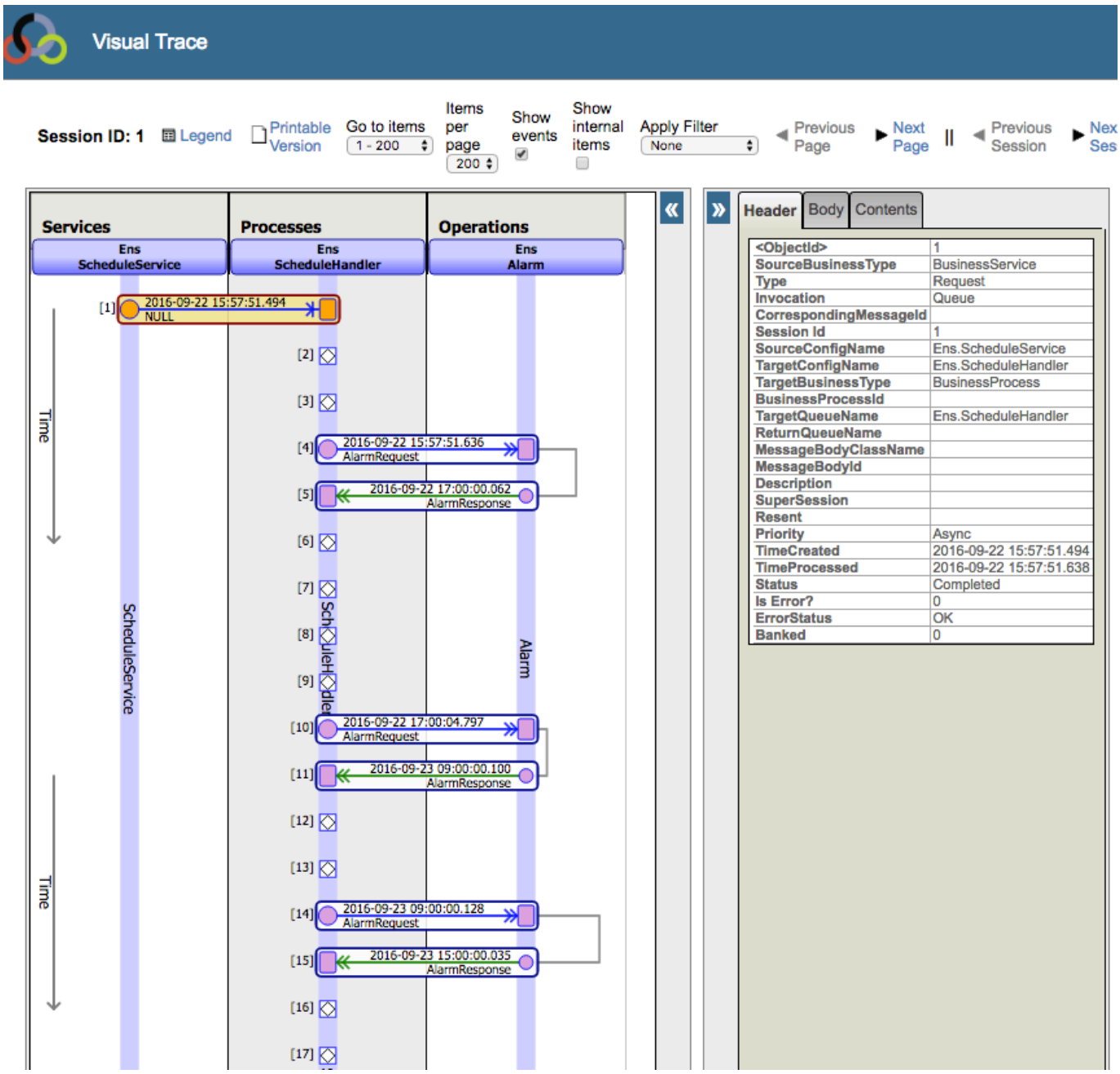
En el ejemplo anterior, Ens.ScheduleHandler recopiló estos tiempos:

09:00: START del servicio HL7 TCP
09:00: START de la operación Passthrough de FTP
15:00: STOP de la operación de Passthrough de FTP
17:00: STOP del servicio HL7 TCP

Si observa con atención el visor de mensajes de abajo, puede ver la marca de tiempo que muestra cuándo se estaba ejecutando Ens.ScheduleHandler: 15:57:51. En ese momento, la próxima hora programada es a las 17:00, por lo que envía un mensaje a Ens.Alarm que básicamente dice "Despiértame a las 17:00":

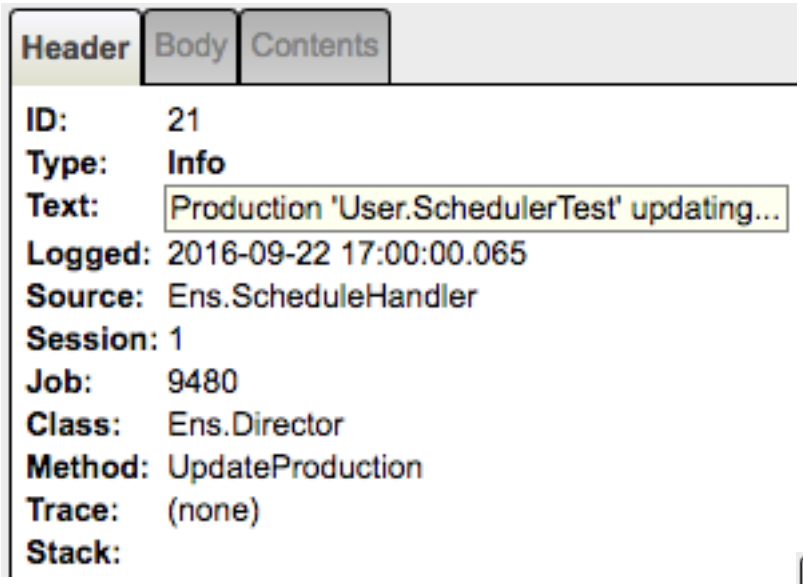


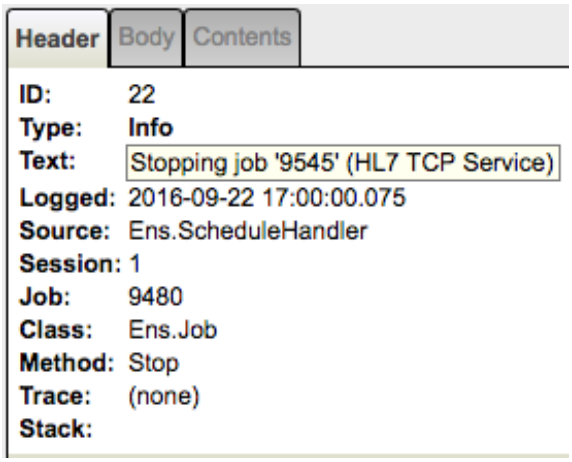
Ens.Alarm espera hasta las 17:00 y luego envía su respuesta. Encima de la " Respuesta de alarma " puede ver la marca de tiempo: 17:00:00, que es la hora a la que respondió la alarma Ens.Alarm. Este mensaje de respuesta hace que Ens.ScheduleHandler llame a UpdateProduction() y el proceso comienza de nuevo. Inicia y detiene los procesos correspondientes. En este caso, como son las 17:00, el servicio HL7 TCP está programado para detenerse.

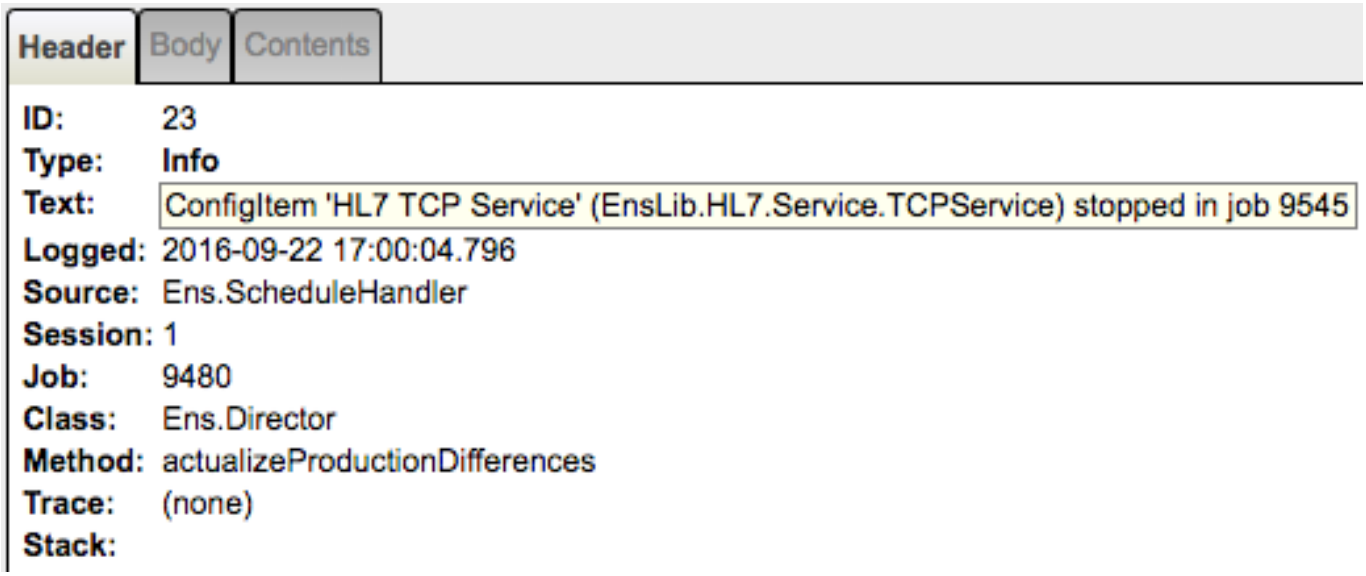


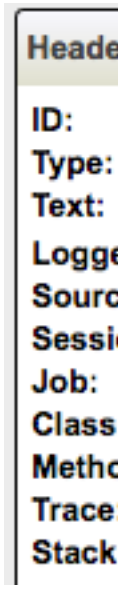
Si hace clic en los elementos de seguimiento marcados con diamantes blancos etiquetados como [6], [7], [8] y [9], verá en el panel derecho:

[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:

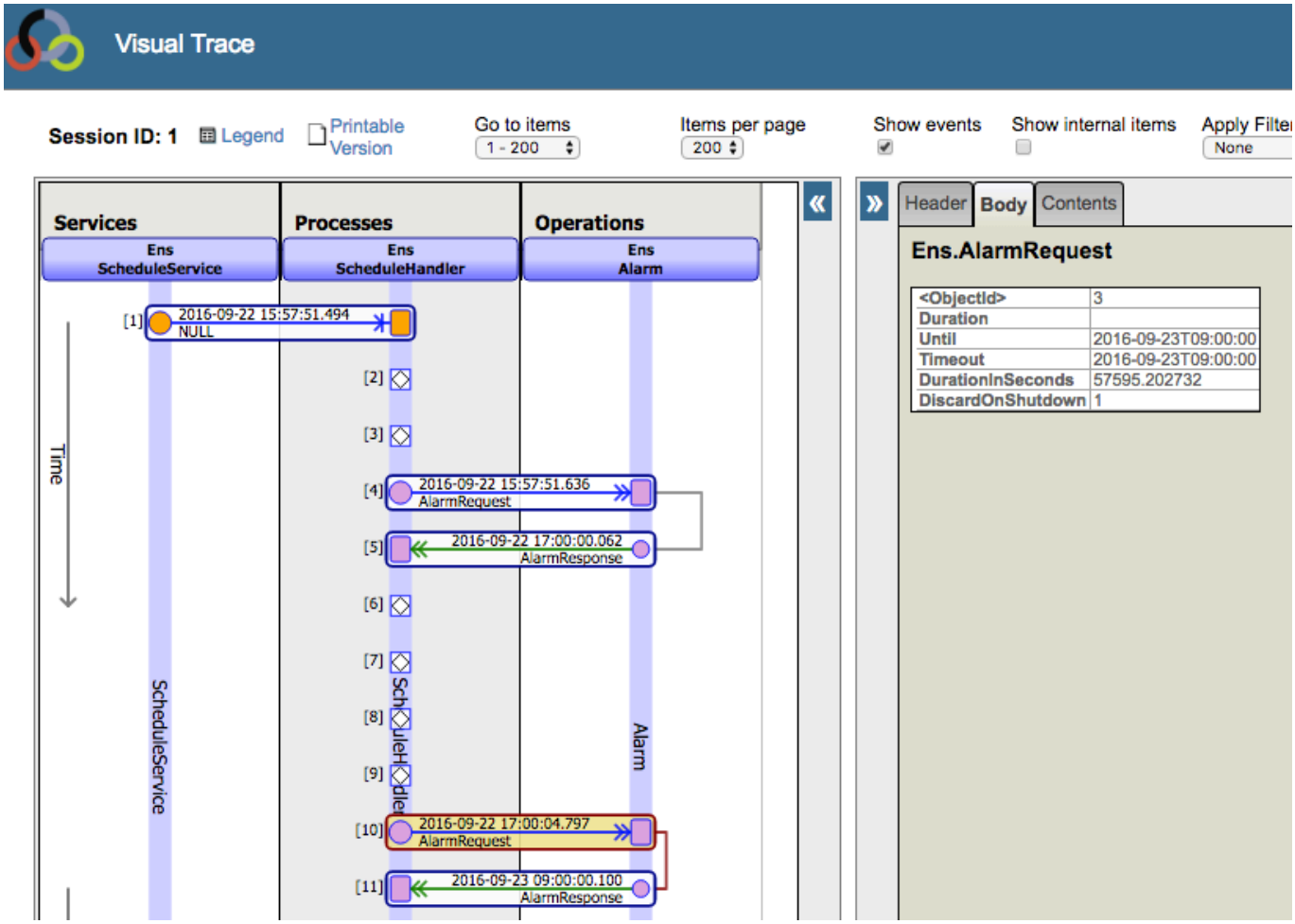

Header
Body
Contents
ID: 23
Type: Info
Text: ConfigItem 'HL7 TCP Service' (EnsLib.HL7.Service.TCPService) stopped in job 9545
Logged: 2016-09-22 17:00:04.796
Source: Ens.ScheduleHandler
Session: 1
Job: 9480
Class: Ens.Director
Method: actualizeProductionDifferences
Trace: (none)
Stack:



Si hubiera más procesos que detener, vería un par de trazos de diamantes blancos similares a [7] y [8] por cada proceso que detiene el programador.

A continuación, Ens.ScheduleHandler observa las horas de START y STOP programadas y determina que la próxima vez que debe actuar es a las 09:00, por lo que envía un mensaje a Ens.Alarm solicitando que despierte a Ens.ScheduleHandler a las 09:00.

La secuencia de pasos a continuación, donde se inician los procesos en lugar de detenerlos, es en gran medida la misma que la anterior, con la excepción de algunas pequeñas trazas que faltan, como se describe a continuación.

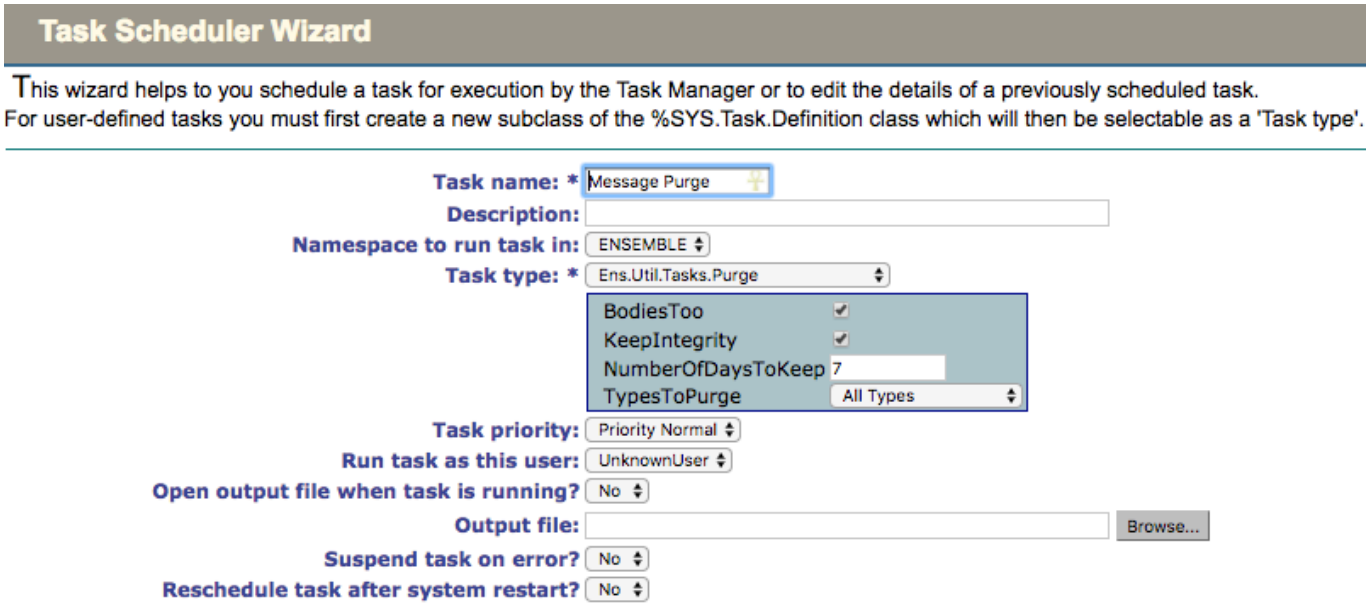


Ens.Alarm espera hasta las 09:00 y luego envía una respuesta. Ens.ScheduleHandler ve que son más de las 09:00 y que necesita iniciar tanto el servicio HL7 TCP como la operación de Passthrough de FTP. El visor de mensajes no muestra elementos de seguimiento que muestren que los componentes se están iniciando porque los propios componentes realizan el registro en lugar de Ens.ScheduleHandler, pero puede verlo en el registro de eventos:

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...

Los mensajes continúan pasando de un lado a otro entre Ens.ScheduleHandler y Ens.Alarm hasta que se detiene la producción (o hasta que alguien elimina los horarios).

Dado que el planificador utiliza mensajes Ensemble, debe configurar la tarea de depuración para que no depure el mensaje del planificador. La forma más segura de evitar la purga accidental de los mensajes del programador es seleccionar siempre la opción Mantener integridad en la tarea Ens.Util.Tasks.Purge (ver captura de pantalla a continuación). Keep Integrity especifica que la tarea solo debe purgar los mensajes que están "completos". Dado que la sesión del programador nunca se completa, no se debe purgar, pero establecer "Keep Integrity" en falso anulará esta protección. Para mas información sobre configuración de " Keep Integrity " , ver https://community.intersystems.com/post/why-keepintegrity-important-when-purging-healthshareensemble-data



Si tiene alguna pregunta o problema con el programador, llame al WRC al 617-621-0700 o envíenos un correo electrónico a support@intersystems.com

#Administración del sistema #Ensemble

URL de fuente:https://es.community.intersystems.com/post/como-funciona-la-planificaci%C3%B3n-de-ensemble