Manual de explotación SynchronizationManager

Control de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Descripción** | **Versión** |
| 20/10/2015 | Borja Gómez | Versión inicial | 0.01 |

Revisiones/Validaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Autor** | **Versión** | **Responsabilidad** | **Rol** | **Fecha** |
|  |  |  |  |  |

Índice de contenidos

[1 Introducción 4](#_Toc431368161)

[2 Resumen de la solución 5](#_Toc431368162)

[2.1 Descripción del proceso 5](#_Toc431368163)

[2.1.1 Recepción de un mensaje 5](#_Toc431368164)

[2.1.2 Arranque del servicio 5](#_Toc431368165)

[2.2 Configuración 5](#_Toc431368166)

[2.2.1 HostConfiguration.json 5](#_Toc431368167)

[2.2.2 KafkaConfiguration.json 6](#_Toc431368168)

[2.3 Logs 7](#_Toc431368169)

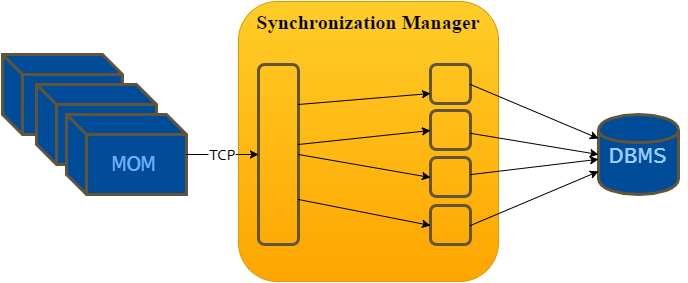
[3 Requerimientos de despliegue 8](#_Toc431368170)

[3.1 Seguridad 8](#_Toc431368171)

# Introducción

* **SynchronizationManager:** Servicio que escucha mensajes enviados al Message oriented middleware, de ahora en adelante MOM, para sincronizar el estado de las distintas bases de datos.

# Resumen de la solución



## Descripción del proceso

### Recepción de un mensaje

1. Al arrancar el sistema lee la configuración.
2. Se levanta en un hilo de ejecución independiente el subscriptor de mensajes que se encarga de escuchar mensajes de la cola de mensajería.
3. Una vez levantado el subscriptor se subscribe a los topics configurados.
4. Se levantan cada uno de los messageHandler configurados y se deja, cada uno en un hilo independiente, a la escucha de un bus interno (memoria).
5. Cuando el subscriptor recibe un mensaje: lo deserializa comprueba su validez y lo publica en el bus interno.
6. Una vez publicada en el bus interno los messageHandler los messageHandler específicos se ejecutan
7. Una vez ejecutados los messageHandler el mensaje se marca como leído.

## Configuración

Todos los ficheros de configuración son ficheros de texto que siguen la notación [JSON](http://www.json.org/). Si alguno de los ficheros no fuese un documento JSON válido el servicio daría un error de configuración en el arranque del servicio.

Para que los cambios de configuración surjan efecto tras modificar la configuración se ha de reiniciar el servicio.

### HostConfiguration.json

Este fichero permite configurar:

1. **KafkaTopicsToListen**: Los topics que el servicio va a escuchar escuchar.
2. **kafkaConfiguration**: Configuración de cliente de Kafka tiene los siguientes valores:
   1. **Addres**: Uris de Kafka
   2. **ConsumerGroup**: Agrupación de consumidor
3. **PortalPacienteConnectionString**. Cadena de conexión a la base de datos portal paciente.
4. **VisionLocalConnectionString**. Cadena de conexión a la base de datos VisionLocal.
5. **MessagesHandlers**. Cada uno de los messageHandlers que se han de ejecutar. Por cada MessageHandler hay que indicar:
6. **MessageHandlerName**: Nombre del messageHandler.
7. **SystemToSynchronize**: Lista de bases de datos que se quiere sincronizar. Los valores válidos son los definidos en el enumerado Davalor.SynchronizationManager.Domain.ESynchroSystem: VisionLocal, PortalPaciente

## Logs

El servicio utiliza [ETW](https://msdn.microsoft.com/en-us/library/windows/desktop/bb968803(v=vs.85).aspx) para la escritura de trazas. Si se quiere monitorizar la actividad del servicio se puede utilizar [Logman](https://technet.microsoft.com/en-us/library/cc753820(WS.10).aspx), [Windows performance monitor](https://technet.microsoft.com/en-us/library/cc749249.aspx) o Davalor.EtwConsumer. Toda la actividad del sistema se traza usando el eventSource “SynchronizationManagerEventSourcing” y se trazan los siguientes eventos:

* Service\_started. Se emite cada vez que se inicia el servicio.
* Service\_stoped. Se emite cada vez que se para el servicio.
* Message\_received. Se emite cada vez que se recibe un mensaje.
* Invalid\_Message\_received. Se emite cada vez que se recibe un mensaje y se detecta que este es inválido.
* Message\_procesed. Se emite cada vez que un handler procesa un mensaje correctamente.
* Message\_processing\_Exception. Se emite cada vez que un handler detecta un error al intentar procesar un mensaje.

## Sistema Plugable

El sistema de sincronización está concebido como un sistema plugable es decir permite añadir funcionalidad, sin necesidad de redesplegar el servicio o modificar el código preexistente, añadiendo nuevas dll’s que sean compatibles con el sistema de plugins utilizado ([MEF](https://msdn.microsoft.com/es-es/library/dd460648(v=vs.110).aspx)) y reiniciando el servicio. Para profundizar en el sistema de plugins leer Manual de desarrollo SynchronizationManager

# Requerimientos de despliegue

## Seguridad

No hay requerimientos especiales.