

IMPORTANT: Use this code under your own responsibility. It's not supported and there is no guarantee that cannot impact your system in case of unknown bugs or if used incorrectly.

License Monitor

Clases:

websys.MonitorLicense.cls
websys.MonitorLicenseUnits.cls
websys.MonitorProcessLUs

Versión: Caché 2014.1+

Instalación: Importar y Compilar en el NameSpace escogido

Configuración y Ejecución

1) Monitorización continua

Llamaremos al método Save() con una frecuencia determinada (hasta el minuto), durante varios días o continuamente.

a. Crear tarea "MonitorLicencias"

Ejecutaremos el método Save() para guardar las métricas de uso de licencias. En el ejemplo, se ejecuta cada minuto mientras la tarea está activa y la tarea en sí se ejecuta diariamente, de 8:00 a 21:00 horas.

Nombre de tarea: * MonitorLicencias
Descripción: Monitorización Uso de Licencias Diario
Namespace para ejecutar tarea: TOOLS
Tipo de tarea: * RunLegacyTask
ExecuteCode: do ##class(webSys.MonitorLicense).Save()
Prioridad de tarea: Prioridad Normal
Ejecutar tarea como este usuario: SuperUser
Abrir archivo de salida cuando la tarea está en ejecución? Sí
Archivo de salida: C:\InterSystems\Ens2017v1\mgr\TASK-ControlLicencias.log Examinar...
Suspender tarea en caso de error? No
Reprogramar tarea tras reinicio del sistema? Sí
¿Con qué frecuencia desea que Gestor de tareas ejecute esta tarea? Diario
Cada 1 día(s)
Fecha Inicio (mm/dd/aaaa): 04/17/2017 Fecha final: 04/19/2017
Ejecutar una vez a esta hora: 08:00:00
Ejecutar cada 1 minutos
1ª hora de ejecución: 08:00:00
Última hora de ejecución: 21:00:00

Ajuste umbral de detalle (opcional). Por defecto, para cada ejecución de la tarea (método Save()) se almacenará un registro con los datos generales de uso. En caso de que las licencias en uso sean más del 50% de las totales, se almacenará igualmente una relación de los usuarios y procesos activos en ese momento. Si se quiere aumentar ese umbral, se deberá indicar como parámetro en la llamada al Save().

Ejemplo: Queremos recoger información de usuarios y procesos sólo cuando el consumo en el momento de la monitorización sea superior al 80%. Para ello configuraremos la tarea con la llamada:

do ##class(webSys.MonitorLicense).Save(,0.8)

Importante.- Para una instalación de unos 300 usuarios concurrentes y unos 80 procesos abiertos en media, esta tarea puede generar unos 6 MB/hora de datos aproximadamente (o 66 MB/día para la franja de 11 horas propuesta en el ejemplo). Si no se dispone de mucho espacio, puede exportarse periódicamente la información recogida y realizar un borrado completo de las estadísticas almacenadas.

2) Monitorización por franja horaria

Cuando queremos monitorizar más en detalle (con más frecuencia) el uso de licencias en una franja horaria necesitamos llamar a `RunByMinute(pPollingTime)`

a. Crear tarea "MonitorDetalleLicencias"

Indicaremos como parámetro del método `RunByMinute` el número de segundos entre muestra.

The screenshot shows a web-based task configuration interface. The task name is 'MonitorDetalleLicencias' and its description is 'Monitor de Uso de Licencias por Minuto'. The namespace is set to 'TOOLS' and the task type is 'RunLegacyTask'. The execution code is `do ##class(webSys.MonitorLicense).RunByMinute(3)`. The priority is 'Normal', the user is 'SuperUser', and the output file is 'C:\InterSystems\Ens2017v1\mgr\TASK-ControlLicPorMin.log'. The task is configured to run daily at 12:00:00, every 1 minute, with a start date of 04/17/2017 and an end date of 04/20/2017. The execution frequency is set to 'Diario' (Daily) and the polling time is 1 minute.

Importante.- Para una instalación de unos 300 usuarios concurrentes y unos 80 procesos abiertos en media, esta tarea puede generar 75 MB/hora de datos aproximadamente. Para evitar problemas de disponibilidad de espacio, sólo debe ejecutarse en aquellos casos en que se quiera tener información detallada en periodos muy limitados de tiempo.

Exportación de Información

Para exportar los datos recogidos entre 2 fechas determinadas podremos hacerlo desde el terminal ejecutando, por ejemplo:

```
do ##class(webSys.MonitorLicense).Export("2017-10-01","2017-10-15","c:\temp\datos")
```

Si solo indicamos el directorio donde queremos los archivos, podemos dejar las fechas en blanco y nos exportará los datos recogidos en la última semana:

```
do ##class(webSys.MonitorLicense).Export(,"c:\temp\datos")
```

Borrado de Estadísticas

Para borrar todas las estadísticas almacenadas:

```
do ##class(webSys.MonitorLicense).Clear()
```