

WiseTime Setup

WiseTime integration consists of multiple components

- WiseTime
- WiseTime Console
- WiseTime Patrawin Connector
- Scheduled Patrawin task
- Service Broker
- Patrawin

WiseTime consultants handles setup of WiseTime, the Console and the WiseTime Patrawin Connector.

CPA global sets up the scheduled Patrawin task.

WiseTime

The monitoring software that keeps track of what the users spends their time on and uploads it to the console.

WiseTime Console

The web interface where the users get an overview of their time spent and decides which time records to post to Patrawin

WiseTime Patrawin Connector

Installed on the client's server and is called by the WiseTime Console when a user posts time to Patrawin.

The connector calls the stored procedure `pw_PostTime` which stores pending time records in `PENDING_TIME_335`.

The procedure also performs some basic validation. These error code are returned back to the console giving the user immediate feedback if the posting was successful or not.

- `CASE_OR_CLIENT_ID_NOT_FOUND`
- `CLIENT_BLOCKED`
- `USER_NOT_FOUND_OR_INACTIVE`
- `ACTIVITY_CODE_NOT_FOUND_OR_INACTIVE`

Service Broker

The scheduled task mentioned below requires the [Service Broker](#) to be enabled on the Patrawin database.

This can be done by executing the following sql on the database. No other connections are allowed at the same time.

```
ALTER DATABASE dbname SET ENABLE_BROKER
```

Scheduled Patrawin Task

A scheduled task that moves the pending time records into Patrawin needs to be set up.

This command starts Patrawin as a pending time monitoring process. The process will run until terminated manually and will pick up any record inserted into `PENDING_TIME_335` and insert it as a record in `TimeCase` ready for invoicing. The user account used to run this job must have the *db_owner* role in the database, and the service broker must be enabled on the database.

```
Patrawin.exe /function:PendingTimeMonitor /server:dbsrv01 /database:pwdb
```

This task should run indefinitely and monitors `PENDING_TIME_335` for new records and creates records in `ATGARDER_120`.

It can be a good idea to recycle the process every 24 hours. The setup below means that the scheduled task is started at some point in time, and then attempted to be restarted every 10 minutes for 24 hours.

A new instance should NOT be created if one is already running.

Edit Trigger Close **Create Task** Close

Begin the task: On a schedule

Settings

☐ One time ☒ Daily ☐ Weekly ☐ Monthly

Start: 2019-12-12 07:02:00 ☐ Synchronize across time zones

Repeat every 1 days

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☒ Repeat task every: 10 minutes for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

☐ Stop task if it runs longer than: 3 days

☐ Expires: 2020-12-12 14:12:28 ☐ Synchronizing across time zones

☒ Enabled

General **Triggers** **Actions** **Conditions** **Settings**

Specify additional settings that affect the behavior of the task.

☒ Allow task to be run on demand

☒ Run task as soon as possible after a scheduled start is missed

☐ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☒ Stop the task if it runs longer than: 3 days

☒ If the running task does not end when requested, force it to stop

☐ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

OK **Cancel** **OK** **Cancel**

Another option would be to start it once and then restart it every 10 minutes indefinitely if not already running.

Edit Trigger Close **Create Task** Close

Begin the task: On a schedule

Settings

☒ One time ☐ Daily ☐ Weekly ☐ Monthly

Start: 2019-12-12 07:02:00 ☐ Synchronize across time zones

Repeat every 1 days

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☒ Repeat task every: 10 minutes for a duration of: indefinitely

☐ Stop all running tasks at end of repetition duration

☐ Stop task if it runs longer than: 3 days

☐ Expires: 2020-12-12 14:12:28 ☐ Synchronizing across time zones

☒ Enabled

General **Triggers** **Actions** **Conditions** **Settings**

Specify additional settings that affect the behavior of the task.

☒ Allow task to be run on demand

☒ Run task as soon as possible after a scheduled start is missed

☐ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☒ Stop the task if it runs longer than: 3 days

☒ If the running task does not end when requested, force it to stop

☐ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

OK **Cancel** **OK** **Cancel**

Patrawin

Records inserted in to Patrawin can be opened in TimeCase and if necessary amended before they are finally invoiced.