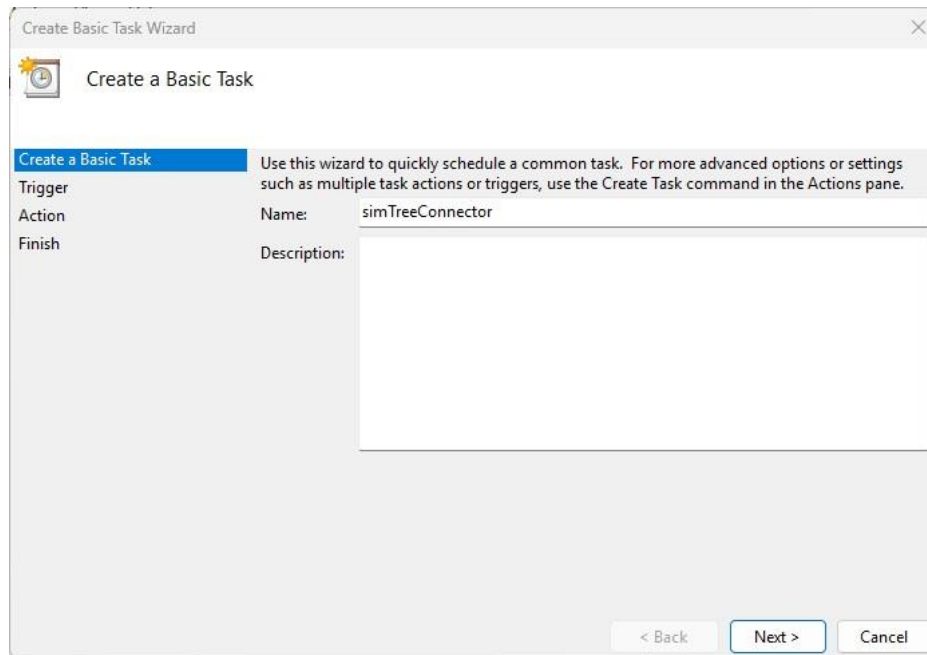




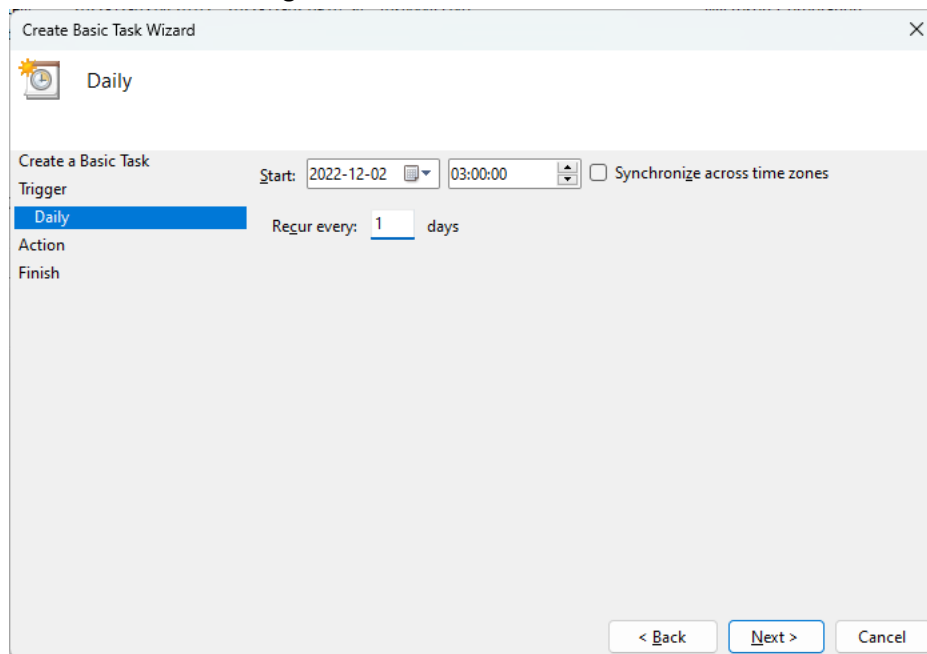
## CONSOLE PROGRAM

### Requirements: .NET Framework 4.8

1. Create a user by navigating to [simTree/settings](https://simTree/settings). Select 'Users' and 'Add API user'. Use the generated e-mail and password or enter your own values. The e-mail does not have to be a real e-mail address, but it must be unique. Press 'Save' after clicking 'OK'.
2. Edit the values in file `simTreeConnector.exe.config`. These values are needed for the connection to your simTree account, the connection to your data source, and to define which data is to be collected. The definitions for the key value pairs can be found in the file `documentation.md`. (The schema is a unique identifier for each customer. If you do not have your schema, reach out to simTree staff to receive it)
3. Run the program `./simTreeConnector.exe` in the console to confirm that it works. If you have not yet imported data, the program will import the last 6 weeks of data. If data has been imported previously, it will import all data with a higher id than included in the last import.
4. To schedule an automatic import of data:
  - Open the Windows program 'Task Scheduler'
  - In the menu choose Action > Create Basic Task...

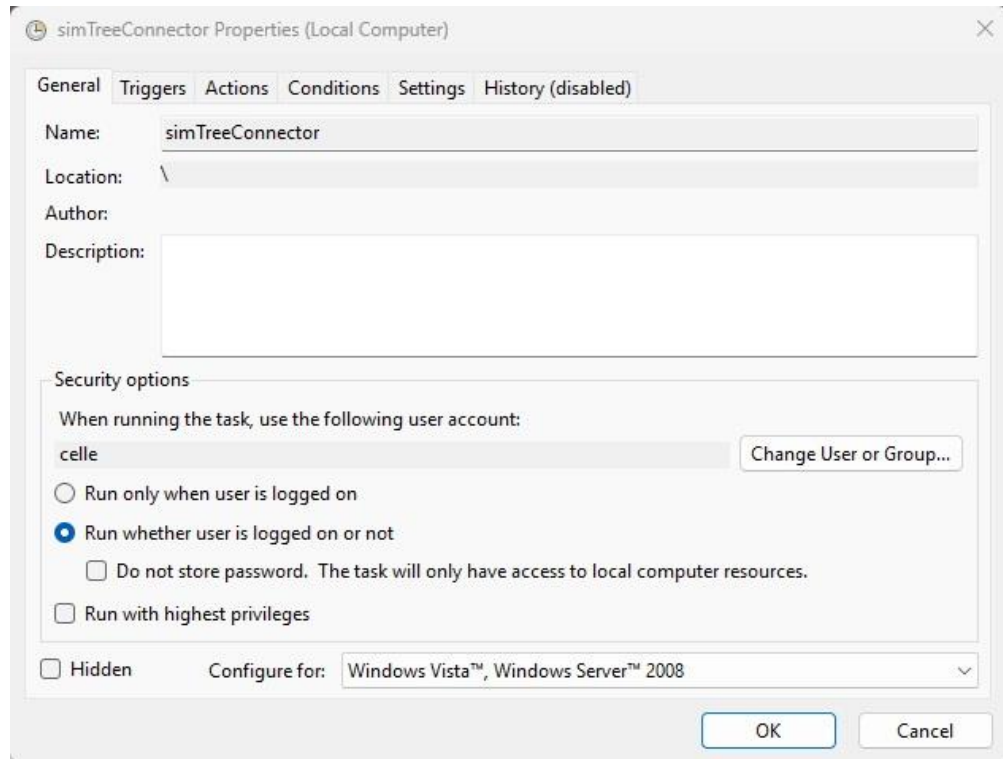


- Give the task a name e.g. simTreeConnector
- Choose Daily as Trigger and choose a time in the night as start time so the data will be available in the morning.



- On the next screen select 'Start a program'
- On the next screen click Browse and select simTreeConnector.exe
- On the next screen check 'Open the Properties dialog for this task when I click Finish' and click Finish

- Select 'Run whether user is logged on or not' if your user is not always logged in.



## SQL QUERY INFORMATION

The SQL query used to retrieve calls from your database differs depending on whether you are using Infinity or Genesis. Review the following SQL query to understand how information is retrieved from each respective system. Please keep in mind when using the console program that you are responsible for the calls made to your database. The SQL queries used in the console program do not include deletion, and thus the risk to your data is minimal. deepNumbers takes no responsibility for any data loss resulting from the misuse of the console program.

### INFINITY SQL QUERY FOR CALLS

```
SELECT TOP 500 ID,
           DATEADD(MINUTE,[Timestamp], '1899-12-31') AS timestamp,
           CAST(CASE WHEN CallKind in (0, 1, 3, 16) THEN [WrapTime] ELSE
[Optime] END AS int) AS 'wrap',
           CAST([Talk] AS int) AS talk,
           CAST([Account] AS bigint) as tag1,
           CAST([Billing] AS bigint) as tag2,
           [CDTNum] as tag3,
           [Abandoned]+[SysAbandon]{vmcbAbandon}+[GreetHangup] as 'Abandons'
FROM mCallEnd
WHERE (CallKind in (0, 16, 2, 3, 4) or (CallKind = 1 and talk > 0))
AND talk BETWEEN -2147483648 AND 2147483647 AND [WrapTime] BETWEEN
-2147483648 AND 2147483647 AND [Optime] BETWEEN -2147483648 AND 2147483647
AND [Account] NOT IN (0, 800) AND [Billing] NOT IN (0, 800) AND ID
> @minId ORDER BY ID"
```

**CallID:** ID

**Timestamp:** Timestamp is used for the time of the call

**Wrap time:** If the CallKind is either 0, 1, 3 or 16, WrapTime is used for the wrap time. Otherwise, Optime is used.

**Talk time:** Talk is used to for the talk time

**Tags:** Account is used as tag1, Billing is used as tag2, CDTNum is used as tag3

**Abandoned calls:** Abandoned, SysAbandon, and GreetHangup are used for Abandoned calls. If your system has the VmcbAbandon column in the table mCallEnd, this is also included in Abandoned calls.

## GENESIS SQL QUERY FOR CALLS

```
$$@"SELECT TOP 500 CAST(CE.callId AS int) as CallID,
      CS . Stamp as timestamp,
      CAST(1000 * CE.selDisc AS int) AS 'wrap',
      CAST(1000 * CE.selTalk AS int) AS 'talk',
      CAST(C.ClientNumber AS bigint) AS tag1,
      CAST(C.BillingCode as bigint) AS tag2,
      Cast(C.Skill AS int) AS tag3,
      case when ce . agtId Is Not NULL and ce . kind in ( 1 , 2 ) and ce
. selTalk = 0 and ce . CompCode = 3 Then 1 --Abandon
      When ce . CompCode = 2 Then 1 --System Abandon
      --When ce.agtId Is Null and ce.kind in (1,2) and ce.CompCode=0 Then
1 --Greeting Hangup
      Else 0
      End as Abandons
FROM dbo.statCallEnd CE
JOIN dbo.statCallStart CS ON CS . callId = CE . callId
JOIN dbo.cltClients C ON C . cltId = CE . cltId
WHERE CE . Kind in ( 1 , 2 , 3 , 4 , 20 , 21 , 20 )
AND CE.callId > @minId ORDER BY CE.callId"
```

**CallID:** The value callId from the table CE is used for CallID

**Timestamp:** Stamp from the table CS is used for timestamp

**Wrap times:** The value selDisc in table CE is used to calculate the wrap time in seconds

**Talk times:** The value selTalk in table CE is used to calculate the talk time in seconds

**Tags:** The values ClientNumber, BillingCode, and Skill in table C are used for tag1, tag2, and tag3 respectively

**Abandoned calls:** The abandons consist of three categories of calls from the table ce:

1. Calls with an agent, a kind of either 1 or 2, and a talk time of 0, and a CompCode of 3
  2. Calls with a CompCode of 2
  3. Calls with no agent, a kind of either 1 or 2, and a CompCode of 0
- These are all included as abandoned calls

## GENESIS SQL QUERY FOR AGENT EVENTS

```
"SELECT [Name] as employeeSwitchId
      , [stamp] as StartTime
      , CAST([type] - 1 AS int) AS kind
      , [value] as reasonCode
FROM dbo.statAgentTracker
WHERE [type] IN (1, 2, 3, 4) AND Stamp >= @minDt AND Stamp < @maxDt
AND stationType in (0, 10, 29)"
```

The agent events use the table `dbo.statAgentTracker`. The events with a type of 1, 2, 3, or 4, and a start time within a certain time frame are exported. Only events with `stationType` 0, 10, or 29 are exported.

**employeeSwitchId:** The value `Name`

**StartTime:** The value `stamp`

**Kind:** the value `type` subtracted by 1

**ReasonCode:** The value `value`