stonebranch

Universal Task Documentation

Universal Automation Center support for Monitoring a Database Column

ut-sqlserver-odbc-monitor-windows

Associated Activities:

Date: 14 March 2018

Author: Nils Buer

Revision: 00

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Customer

Revision	Date	Author	Changes
00	20180314	Nils Buer	Initial Document (WIP)

Abstract:

This Universal Task allows to Monitor a value in a Column in a schema of a SQLSERVER database. If the given value is inserted into the database, the Task goes to success. It is also possible to monitor multiple values within a database table row.

Contents

1 Disclaimer

No support and no warranty are provided by Stonebranch GmbH for this document and the related Universal Task. The use of this document and the related Universal Task is on your own risk.

Before using this task in a production system, please perform extensive testing.

Stonebranch GmbH assumes no liability for damage caused by the performance of the Universal Tasks

2 Introduction

The here described task has been created to demonstrate how a value in a database column could be monitored in a SQLSERVER database. If you have an Oracle or MySQL database, the python script in the Universal Template could be adjusted to support other databases.

Some details about the universal task to monitor a value in a database column:

- The Universal Task allows to monitor one or multiple values in a table Row
- With minor adjustments the task also runs on a Universal Agent for Linux/Unix
- The Windows Server needs to have Python 2.7. or higher installed
- The Windows Server needs to have the Microsoft ODBC driver installed
- You can configure all connection Parameters via the Universal Task
- You can select different log-levels e.g. Info and debug
- Restriction: Currently only SQLSERVER with SQLSERVER AUTHENTICATION is support

3 Installation

3.1 Software Requirements

Universal Template name: UT_ODBC_MONITOR_WIN

Related UAC XML Files for template and task: [1]

Software used:

For the set-up you need:

1. Python 2.7.x or higher installed on a server where an Universal Agent is installed.
2. For Python the following modules are required:
    - *sys, for output re-direct processing*
    - *datetime, date and time stamps for messages*
    - *pyodbc to perform the ODBC connection*
    - *logging, for python loglevel support*

    Note: Only the module requests need to be added to python 3.6.3
3. Universal Controller at least 6.4.2.x
4. Universal Agent 6.3.0.3 or higher installed on a Windows Server
5. Microsoft ODBC driver installed on the Windows SQLSERVER
6. A database user with SQLSERVER AUTHENTICATION to perform a select on the table to monitor must be available

3.2 Installation Steps

The following describes the installation steps:

1. **Install Python 2.7.x for Windows on the Universal Controller server or any Windows Server running a Universal Agent.**

   Official Download link: https://www.python.org/downloads/

   Note:

   Install python with the options:

   - add python to windows path
   - Install for all users

2. **Add the required python modules**

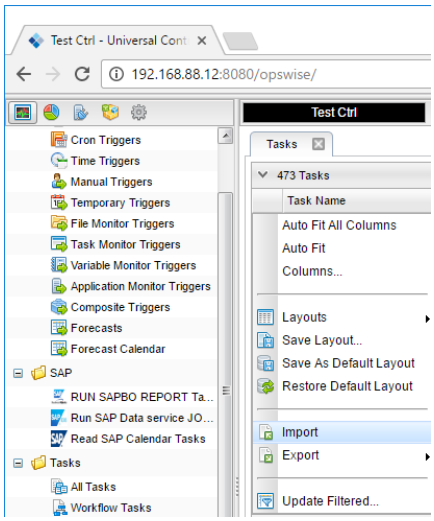
   In a dos command shell run as Administrator:

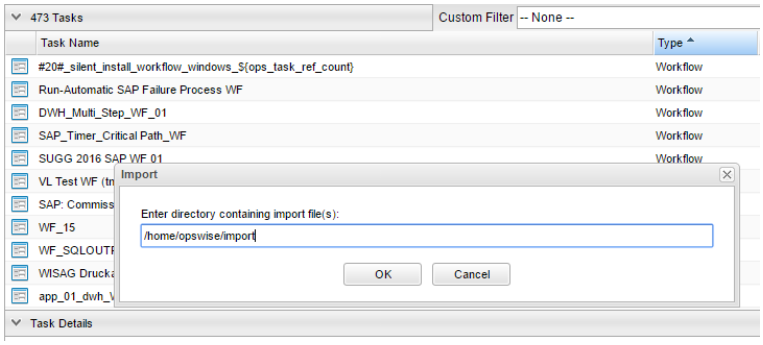
   - *pip install pyodbc*
   - *pip install datetime*
   - *pip install logging*

   Note: The module *pyodbc* contain the commands to perform SQL commands via an ODBC connection a SQLSERVER database.
   It is assumed that the modules sys, datetime are already available. If not install them via pip.

3. **Import the Universal Task including the Universal Template to your Controller**

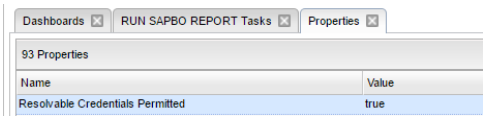
Go to “All Tasks” and load via the Import functionality the Universal Task configuration into the Controller.



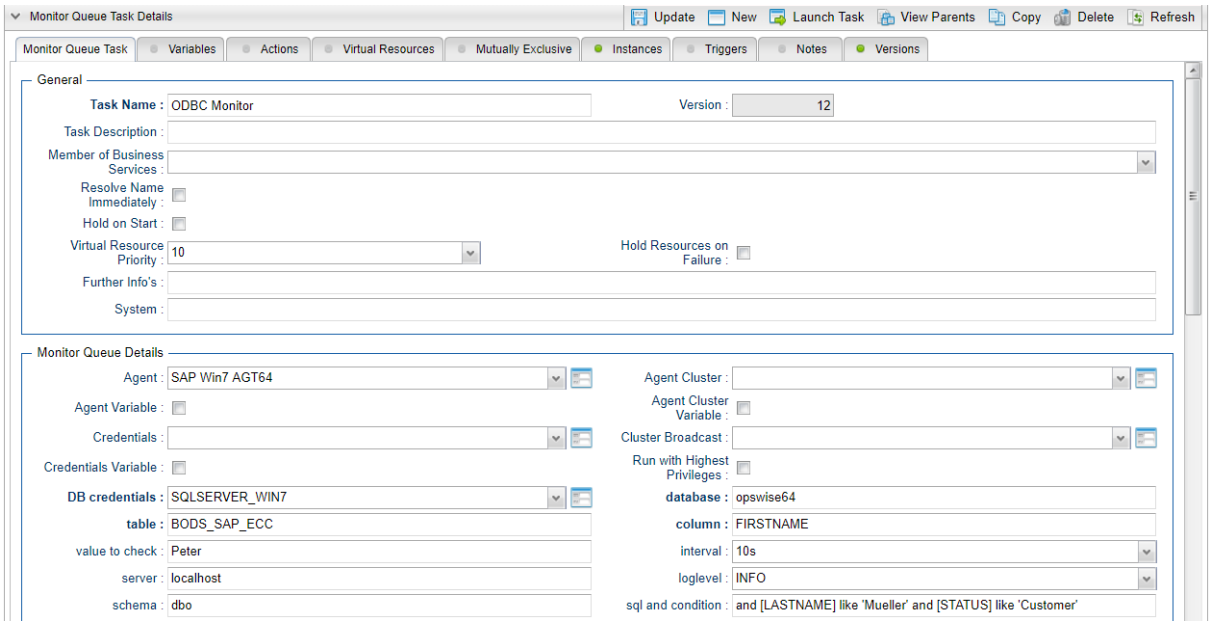


4 Universal Task Configuration

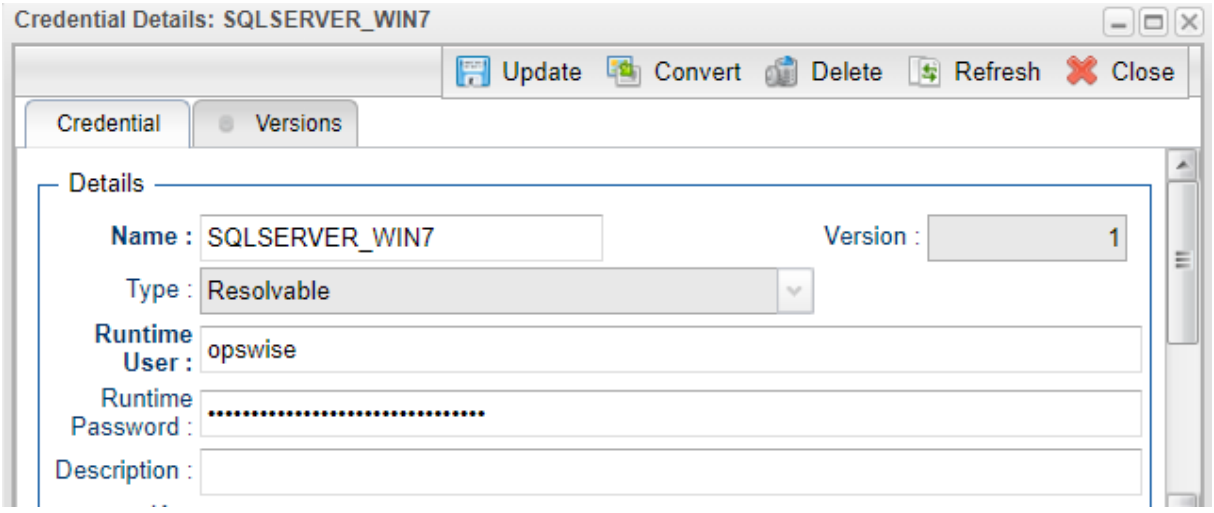
1. Activate: Resolvable Credentials in Universal Automation Center:



2. Fill Out the Universal Task for each Report to be scheduled:



Credential for Universal Task:



Description:

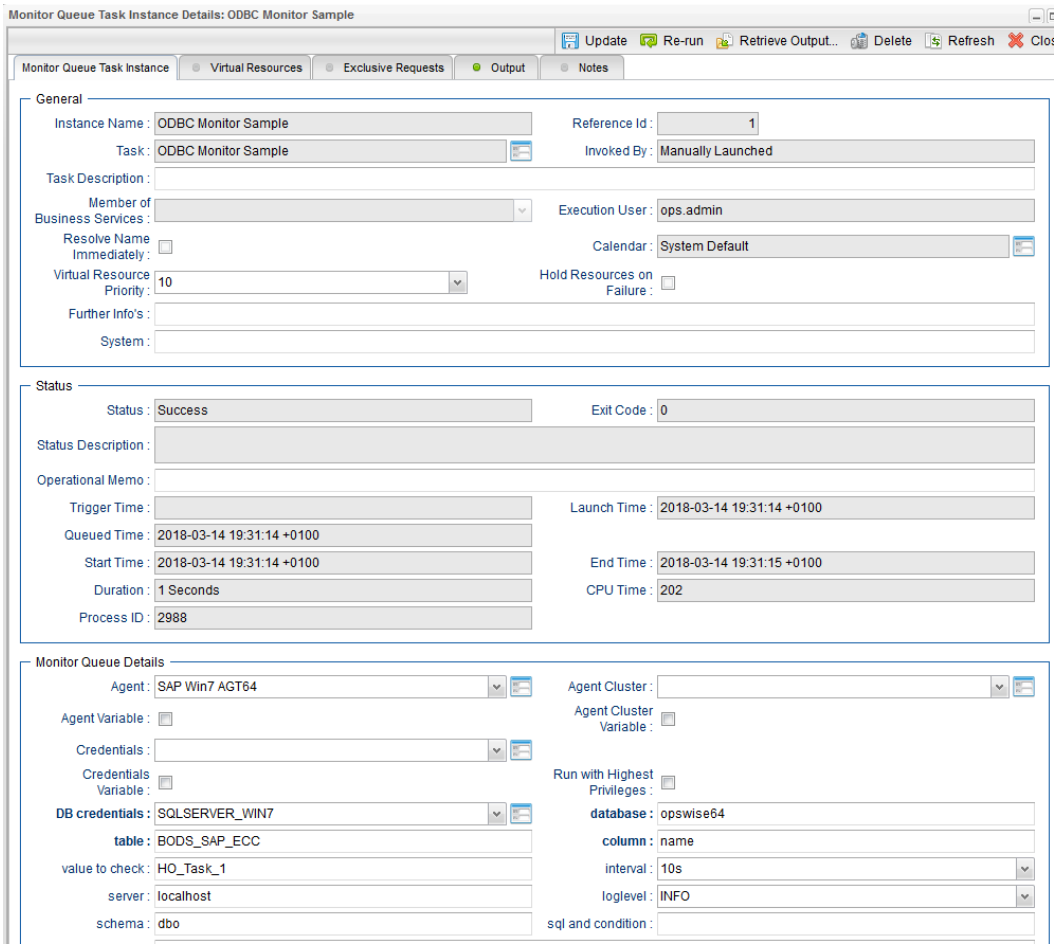
Field	Description
DB credentials	Database Credentials
Database	Database in which the field to monitor is located
Table	Table in which the field to monitor is located
Column	Column in which the field to monitor is located
Value	Value to monitor
Interval	Interval to scan the table for a new value entry
Server	Database Servername e.g. localhost
Loglevel	log level: DEBUG, INFO, WARNING, ERROR, CRITICAL
Schema	Database Schema
Andcondition	If more than one value needs to be monitored add here an and condition in this field e.g.: *and [lastname] like 'meyer' and [firstname] like 'bob'*

5 How to try it out

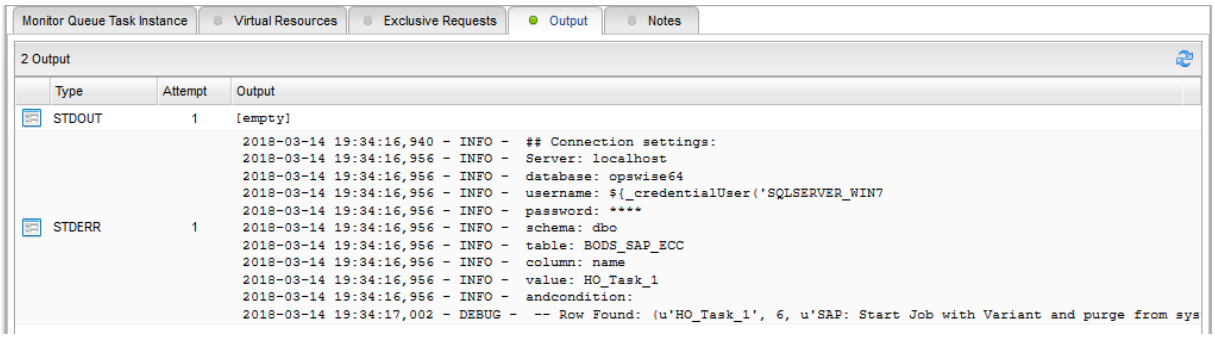
Steps:

a) Configure an Universal Task as described in [4] to monitor a certain value in a database column
b) Launch the Universal Task
    a. In case the Value to Monitor is already in the database, the Universal Task goes to status success. In the output of the task you can see that the value has been found
    b. In case the Value to Monitor is not yet inserted in the Database the Universal Tasks stays in status “running”, until the value is inserted.

Task in status “success” after the value in the database has been identified.


Monitor Queue Task Instance Details: ODBC Monitor Sample
Update
Re-run
Retrieve Output...
Delete
Refresh
Clos
Monitor Queue Task Instance
Virtual Resources
Exclusive Requests
Output
Notes
General
Instance Name : ODBC Monitor Sample
Reference Id : 1
Task : ODBC Monitor Sample
Invoked By : Manually Launched
Task Description :
Member of Business Services :
Execution User : ops.admin
Resolve Name Immediately :
Calendar : System Default
Virtual Resource Priority : 10
Hold Resources on Failure :
Further Info's :
System :
Status
Status : Success
Exit Code : 0
Status Description :
Operational Memo :
Trigger Time :
Launch Time : 2018-03-14 19:31:14 +0100
Queued Time : 2018-03-14 19:31:14 +0100
Start Time : 2018-03-14 19:31:14 +0100
End Time : 2018-03-14 19:31:15 +0100
Duration : 1 Seconds
CPU Time : 202
Process ID : 2988
Monitor Queue Details
Agent : SAP Win7 AGT64
Agent Cluster :
Agent Variable :
Agent Cluster Variable :
Credentials :
Credentials Variable :
Run with Highest Privileges :
DB credentials : SQLSERVER_WIN7
database : opswise64
table : BODS_SAP_ECC
column : name
value to check : HO_Task_1
interval : 10s
server : localhost
loglevel : INFO
schema : dbo
sql and condition :

Output after success:



6 Test Cases

The following basic test cases has been performed:

Case#	Assumed behavior	Result
Start a Job monitoring for a value in a column that exists	The Task goes to success	Correct
Start a Job monitoring for a value in a column that does not exists	The task stays running, until someone inserts the value in the column	Correct
Start Job with wrong DB credentials	[28000] [Microsoft][ODBC SQL Server Driver][SQL Server]Login failed for user 'xyz'.	Correct
Start a Job with wrong database name	[42000] [Microsoft][ODBC SQL Server Driver][SQL Server]Cannot open database "xyz"	Correct
Start a Job with wrong Column name	[42S22] [Microsoft][ODBC SQL Server Driver][SQL Server]Invalid column name 'xyz'	Correct
Start a Job with wrong table name	[42S02] [Microsoft][ODBC SQL Server Driver][SQL Server]Invalid object name 'xyt'	Correct

7 Document References

This document references the following documents:

Ref#	Description
[1] XML extract of Universal Task	UAC XML extract of the Universal Template and Task • ops_unv_tmplt_f23b540a920b40ceb52391af6ce2abb8 • ops_task_universal_1797484ed8ba40da9233ddeb903ac8f4 • ops_credentials_10717d3bff274dd18440abd759a425c7