Integration Pack for MSSQL Tasks 1.1 – User Guide

*By: Charles Joy (charles.joy@microsoft.com)*

Table of Contents

[Introduction 3](#_Toc308087464)

[System Requirements 3](#_Toc308087465)

[Installation 3](#_Toc308087466)

[Removal 3](#_Toc308087467)

[Post-Installation Configurations 4](#_Toc308087468)

[Configuring Orchestrator for MSSQL Connection 4](#_Toc308087469)

[Test MSSQL Connection 5](#_Toc308087470)

[Deploying New Variable Specific Stored Procedures 6](#_Toc308087471)

[Locating the MSSQL Deployment Scripts Installed 6](#_Toc308087472)

[Deploying the MSSQL Scripts 6](#_Toc308087473)

[Activities in Depth 10](#_Toc308087474)

[MSSQL Connection Settings 10](#_Toc308087475)

[Test MSSQL Connection 10](#_Toc308087476)

[Deploy MSSQL Scripts 11](#_Toc308087477)

[Get Variable Info 11](#_Toc308087478)

[Functionality 12](#_Toc308087479)

[Update Variable Value 12](#_Toc308087480)

[Functionality 13](#_Toc308087481)

[Generate XML from MSSQL 14](#_Toc308087482)

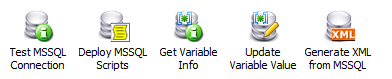
[Functionality 15](#_Toc308087483)

# Introduction

The Integration Pack for MSSQL Tasks provides extended Data Management functions to Orchestrator and MSSQL.

This Integration Pack adds the following Activities to Orchestrator:

* Test MSSQL Connection
* Deploy MSSQL Scripts
* Get Variable Info
* Update Variable Value
* Generate XML from MSSQL



***Figure A:*** Object Images for the Activities in the Integration Pack.

# System Requirements

* System Center 2012 Orchestrator (RC or later)
* System Center 2012 Orchestrator Integration Toolkit (library included)
* Microsoft .NET 2.0 Framework
* Microsoft SQL Server

# Installation

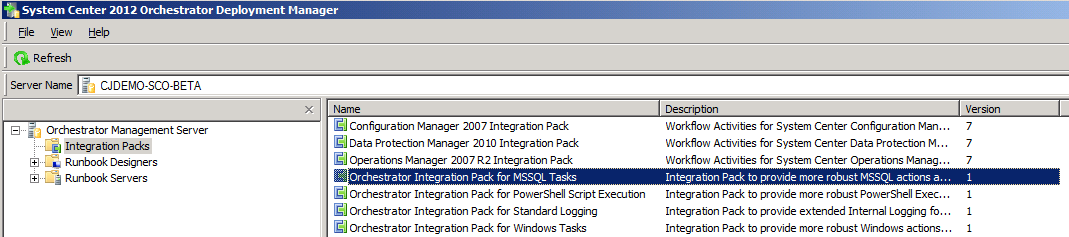
Install the Integration Pack using the Orchestrator Deployment Manager according to the instructions on TechNet (<http://technet.microsoft.com/en-us/library/hh420346.aspx>).

# Removal

Uninstall the Integration Pack using the Orchestrator Deployment Manager according to the instructions on TechNet. The following link is for 6.3 (the steps are fundamentally the same): <http://technet.microsoft.com/en-us/library/gg440809.aspx>

# Post-Installation Configurations

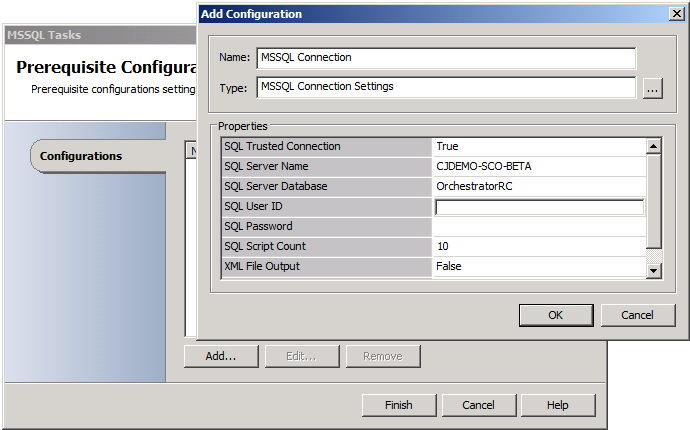
The following illustrates what Post-Installation Configurations are required to execute all the Activities within this Integration Pack.



***Figure B:*** Successful Installation of the Orchestrator Integration Pack for MSSQL Management 1.1.

## Configuring Orchestrator for MSSQL Connection

Configure this Integration Pack the same way as all others, in the Options Menu: MSSQL Management.



***Figure C:*** Configured “MSSQL Connection Settings” within Orchestrator.

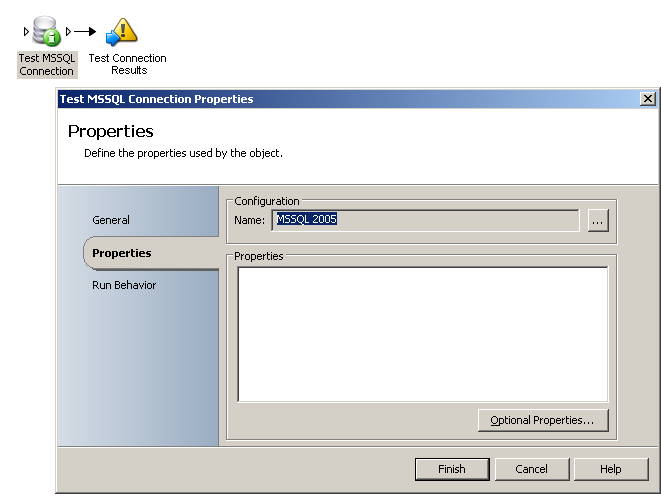
As many connections that are needed can be configured here. Each connection is specific to a single Server and Database.

The Connection can be either Trusted (SSPI) or not (SQL Credentials Required -- as shown in Figure C above). If a Trusted Connection is selected, SQL User ID and SQL Password are not required.

## Test MSSQL Connection

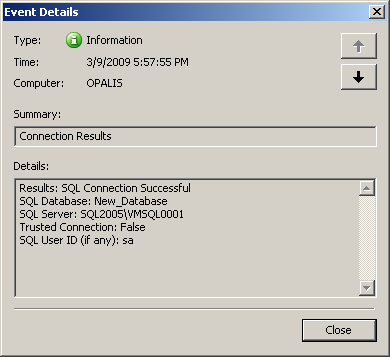
Once a connection has been created, the activity “Test MSSQL Connection” can be used to validate the connection before more Activities are configured and used.

This activity has no Properties to fill in. Simply select the Configuration from Section 3.1 and Click “Finish”.



***Figure D:*** Configured “Test MSSQL Connection” Object.

Execute the Policy (this runbook has a “Send Platform Event” to illustrate some of the Published Data of this activity).



***Figure E:*** Results of the Connection Test.

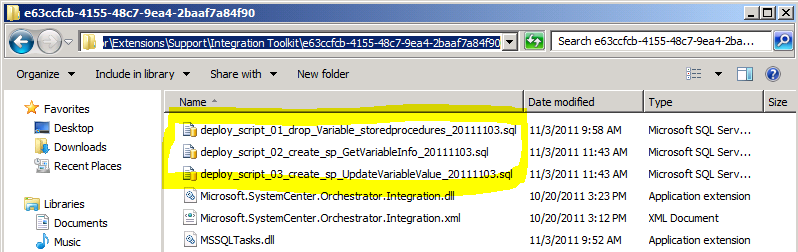
## Deploying New Variable Specific Stored Procedures

Once a valid connection has been configured and validated (Sections 3.1 - 3.2), the next activity in this Post-Installation Configuration Section can be used to Deploy the necessary MSSQL Scripts to the target Orchestrator Database (“Deploy MSSQL Scripts”).

### Locating the MSSQL Deployment Scripts Installed

These MSSQL Scripts contain the DROP and CREATE STATEMENTS required for the successful use of the “Get Variable Info” and “Update Variable Value” Activities.

The MSSQL Scripts were included in the .oip installation file for this Integration Pack and can be found here: *C:\Program Files (x86)\Common Files\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\****e63ccfcb-4155-48c7-9ea4-2baaf7a84f90***



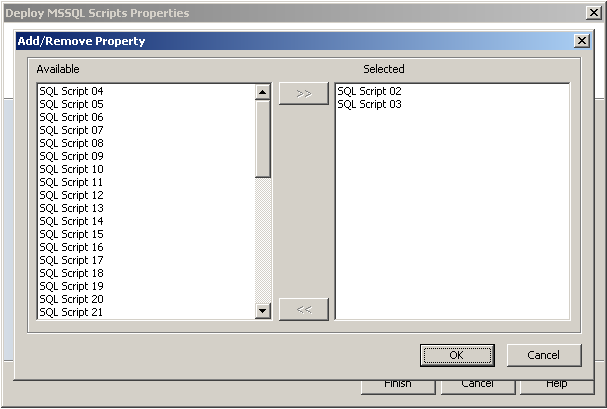
***Figure F:*** Directory of the MSSQL Deployment Scripts for the New Variable Specific Stored Procedures.

***NOTE:*** *The GUID for your install may vary, if it does, simply search for \*.sql in the “Integration Toolkit” directory (one level above the GUID named directory)*

### Deploying the MSSQL Scripts

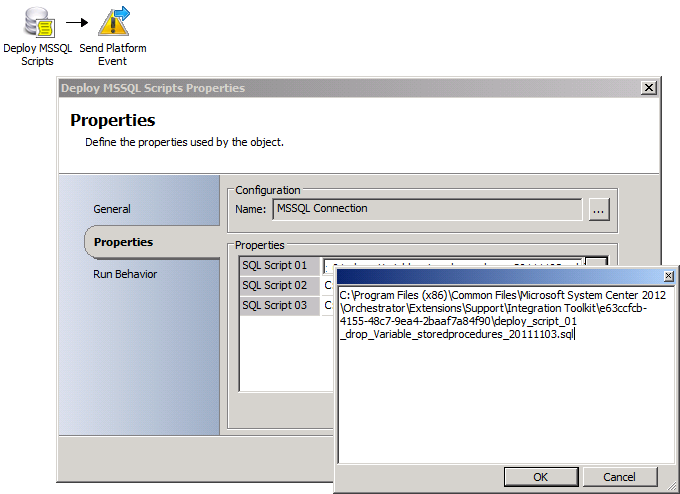
Once the MSSQL deployment scripts have been located, the activity “Deploy MSSQL Scripts” can be used to deploy these scripts to the target MSSQL Server.

This activity has a variable amount of properties to be filled in. By default, only one field is Required: “SQL Script 01”, but a configurable (in the MSSQL Connection Settings Configuration) amount of sequential scripts can be executed by selecting the “Options” button on the Properties Tab. For this section, only “SQL Script 01” - “SQL Script - 03” are necessary.



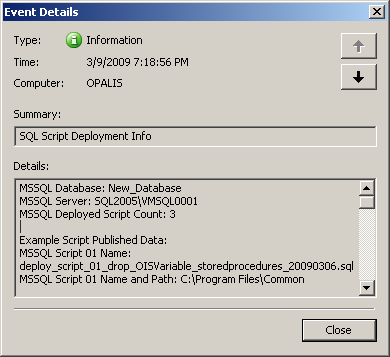
***Figure G:*** Add/Remove Property screen where optional properties can be selected.

Each of the “SQL Script …” Property fields are file selection dialogs. Simply navigate to the desired script and select it. These fields are acted upon sequentially, so be sure to add the files to the fields in the order specified in their name.



***Figure H:*** Configured “Deploy MSSQL Scripts” Object.

Execute the Policy (this runbook has a “Send Platform Event” to illustrate some of the Published Data of this activity).



***Figure I:*** Results of the Script Deployment.

MSSQL Database: OrchestratorRC

MSSQL Server: CJDEMO-SCO-BETA

MSSQL Deployed Script Count: 3

Example Script Published Data:

MSSQL Script 01 Name: deploy\_script\_01\_drop\_OISVariable\_storedprocedures\_20111103.sql

MSSQL Script 01 Name and Path: C:\Program Files (x86)\Common Files\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\e63ccfcb-4155-48c7-9ea4-2baaf7a84f90\deploy\_script\_01\_drop\_OISVariable\_storedprocedures\_20111103.sql

MSSQL Script 01 Path: C:\Program Files (x86)\Common Files\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\e63ccfcb-4155-48c7-9ea4-2baaf7a84f90\

MSSQL Script 01 Text: /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DROP VARIABLE STORED PROCEDURES

The following set of commands will DROP existing versions of the following Stored Procedures:

sp\_GetVariableInfo

sp\_UpdateVariable

INPUT: N/A

USAGE: EXECUTE File Contents (within or MSSQL)

OUTPUT: N/A

HISTORY

Revision Created By Date Comments

--------------------------------------------------------

0.1 Charles Joy 03/06/2008 Created.

0.2 Charles Joy 11/03/2011 Removed References to OIS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if exists (select \* from dbo.sysobjects where id = object\_id(N'[dbo].[sp\_UpdateVariableValue]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)

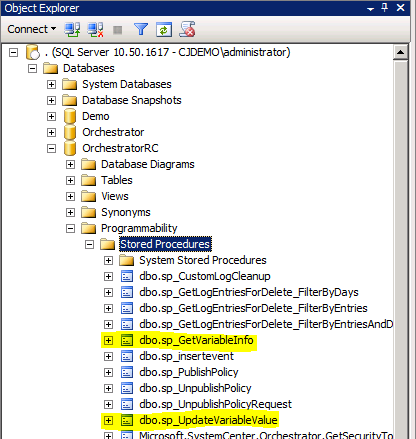
DROP PROCEDURE [dbo].[sp\_UpdateVariableValue]

if exists (select \* from dbo.sysobjects where id = object\_id(N'[dbo].[sp\_GetVariableInfo]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)

DROP PROCEDURE [dbo].[sp\_GetVariableInfo]

***Figure J:*** The entire execution output for the “Send Platform Event” activity from the execution of the “Deploy MSSQL Scripts” Object.

The “physical” results of the execution of all three scripts is the actual DROP (if previous versions exists) then CREATE of two (2) stored procedures in the target MSSQL Database.



***Figure K:*** Stored Procedure Listing on the target MSSQL Database.

# Activities in Depth

The following illustrates an in-depth look at all the Activities and configuration options within this Integration Pack.

## MSSQL Connection Settings

**Input/Output Properties:**

**SQL Trusted Connection** (*Boolean Browser*): True/False for whether a Trusted (SSPI) or Not Trusted (SQL Credentials Required) Connection to MSSQL will be made (Default = True).

**SQL Server Name** (*Text Field*): Field where the name of the MSSQL Server is entered.

**SQL Server Database** (*Text Field*): Field where the name of the MSSQL Server Database for is entered.

**SQL User ID** (*Text Field*): Field where the SQL User ID is entered.

**SQL Password** (*Password Protected Text Field*): Field where the SQL Password associated with the SQL User ID is entered.

**SQL Script Count** (*Text Field*): Number of Scripts to be Displayed as Optional Properties in the Deploy MSSQL Scripts activity (Default = 10).

**XML File Output** (*Boolean Browser*): True/False for whether the XML Output will include XML Files for the Generate XML from MSSQL activity (Default = False).

**XML Output Count** (*Text Field*): Number of XML Output Properties to be Displayed as Optional Properties in the Generate XML from MSSQL activity (Default = 10, XML File Output = True required if these fields are to be seen).

NOTE: See ***Figure C*** above for an image of a configured MSSQL Connection Settings screen.

## Test MSSQL Connection

As seen in Section 3.2, this activity is used to Test the Configured Connection to the target MSSQL Server and Database. (*NOTE: Functionality sub-section here left out en lieu of instruction in Section 3.2.*)

**Input Properties:** **None**

**Published Data:**

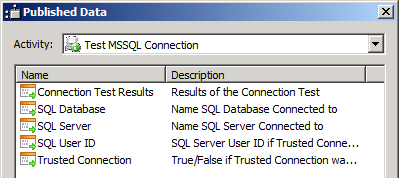
Connection Test Results

Trusted Connection

SQL Database

SQL Server

SQL User ID



***Figure L:*** Published Data for “Test MSSQL Connection”

## Deploy MSSQL Scripts

As seen in Section 3.3.2, this activity is used to Deploy Scripts to the Configured target MSSQL Server and Database. (*NOTE: Functionality sub-section here left out en lieu of instruction in Section 3.2.2.*)

**Input Properties:**

**SQL Script 01[-N]** (*File Browser(s)*): 1 required and N optional fields for the selection of MSSQL Scripts. Each script is deployed in sequence from 01 - N. The ‘0’ Padding to the left of the number will increase directly with the value of N (e.g. If N = 100, Padding = 3, thus the field name will contain 001).

**Published Data:**

SQL Script 01[-N] Name

SQL Script 01[-N] Name and Path

SQL Script 01[-N] Path

SQL Script 01[-N] Results

SQL Script 01[-N] Text

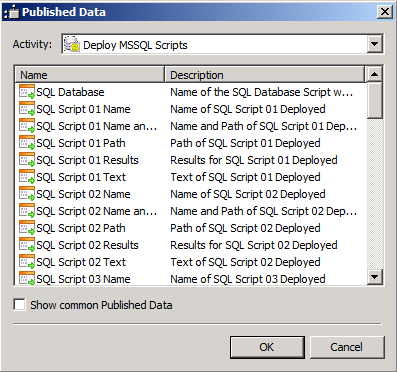
SQL Script Count

SQL Database

SQL Server

SQL User ID

Trusted Connection



***Figure M:*** Published Data for “Deploy MSSQL Scripts”

## Get Variable Info

This activity is used to Get Specific Variable Information from the configured target Orchestrator MSSQL Server and Database.

**Input Properties:**

**Variable Chooser** (*List Browser*): All currently available Variables in the target Orchestrator MSSQL Database. *Data Format:* [Variable Name];[Variable Path];[Variable GUID];[Variable Value]

**Published Data:**

Variable GUID

Variable Info (delimited by semicolon)

Variable Name

Variable Path

Variable Value

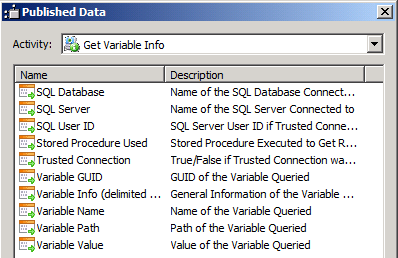
SQL Database

SQL Server

SQL User ID

Stored Procedure Used

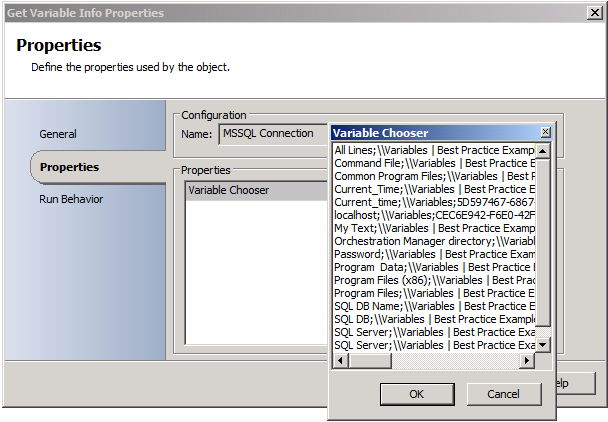
Trusted Connection



***Figure N:*** Published Data for “Get Variable Info”

### Functionality

Simply use the List Browser in the “Variable Chooser” field to select (and view) specific Variable Information.



***Figure O:*** Configured “Get Variable Info” Object.

After this Object’s Execution each of the Delimited Data items in the List Browser for the Variable selected will be available as Published Data.

NOTE: All Data is current as of the execution of the activity; the GUID of the selected item is used to query the Database during Object Execution and returns the most up to date information.

## Update Variable Value

This activity is used to Update Specific Variable Value in the configured target Orchestrator MSSQL Server and Database.

**Input Properties:**

**Use Variable Chooser** (*Boolean Browser*): True/False for whether the Variable Chooser will be used (Default = True).

**Variable Chooser** (*List Browser*): All currently available Variables in the target Orchestrator MSSQL Database. *Data Format:* [Variable Name];[Variable Path];[Variable GUID];[Variable Value]

**New Variable Value** (*Text Field*): Field where the new value of the Variable is entered.

***[Optional]*** **Variable GUID** (*Text Field*): Field where the known Variable GUID is entered (to be used only with **Use Variable Chooser** =False and **Variable Chooser** ="")

**Published Data:**

Variable GUID

Variable Info (delimited by semicolon)

Variable Name

Variable Path

New Variable Value

Old Variable Value

Use Variable Chooser

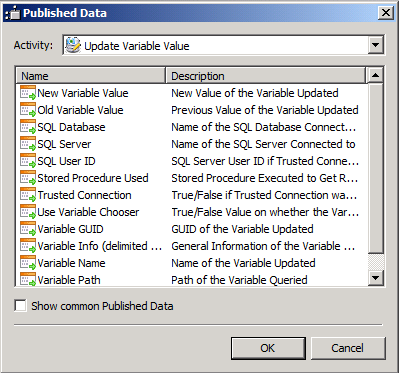
SQL Database

SQL Server

SQL User ID

Stored Procedure Used

Trusted Connection



***Figure P:*** Published Data for “Update Variable Value”

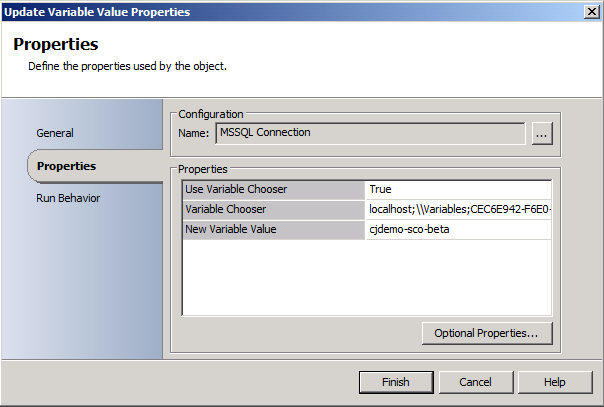
### Functionality

**Use Variable Chooser = True (Default)**

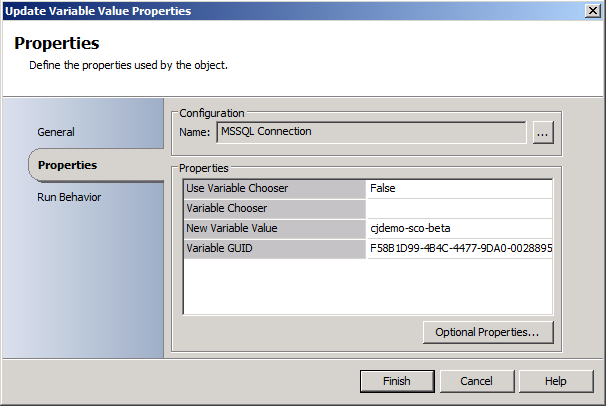
* Simply use the List Browser in the “Variable Chooser” field to select (and view) specific Variable Information.
* Select the Variable to be Updated.
* Enter a value for the new value of the variable in the “New Variable Value” field.

**Use Variable Chooser = False (Optional)**

* Update the “Use Variable Chooser” field to “False”
* Ensure the “Variable Chooser” field is blank.
* Enter a value for the new value of the variable in the “New Variable Value” field.
* Select the Optional field “Variable GUID”
* Enter (or Subscribe to Published Data for) the GUID



***Figure Q:*** Default Configuration for the “Update Variable Value” Object.



***Figure R:*** Optional Configuration for the “Update Variable Value” Object.

After this Object’s Execution the Variable selected will be updated with the New Value and each of the Delimited Data items in the List Browser (if used) for the Variable selected will be available as Published Data. Even if the Optional Configuration is used, all Published Data is still available.

NOTE: All Data is current as of the execution of the activity; the GUID of the selected item is used to query the Database during Object Execution and returns the most up to date information.

## Generate XML from MSSQL

This activity is used to Generate XML Output (XML File and/or Published Data) from MSSQL Query.

**Input Properties:**

**01[-N] XML Output File** (*File Browser(s)*): 1 required and N optional fields for the selection of XML Output files. Each XML output file name is correlated to its respective 01[-N] SQL Query. The ‘0’ Padding to the left of the number will increase directly with the value of N (e.g. If N = 100, Padding = 3, thus the field name will start with 001).

**01[-N] SQL Query** (*Text Field(s)*): 1 required and N optional text fields for the entry of a SQL Query. Each query is executed in sequence from 01 - N. The ‘0’ Padding to the left of the number will increase directly with the value of N (e.g. If N = 100, Padding = 3, thus the field name will start with 001).

**01[-N] SQL Dataset Name** (*Text Field(s)*): 1 required and N optional text fields for the entry of a SQL Dataset Name. Each dataset name is correlated to its respective 01[-N] SQL Query. This SQL Dataset Name will be the primary node in the XML Output. The ‘0’ Padding to the left of the number will increase directly with the value of N (e.g. If N = 100, Padding = 3, thus the field name will start with 001).

**Published Data:**

01[-N] SQL Query

01[-N] SQL Dataset Name

01[-N] XML Output

01[-N] XML Output Name (*optional*)

01[-N] XML Output Path (*optional*)

01[-N] XML Output Name and Path (*optional*)

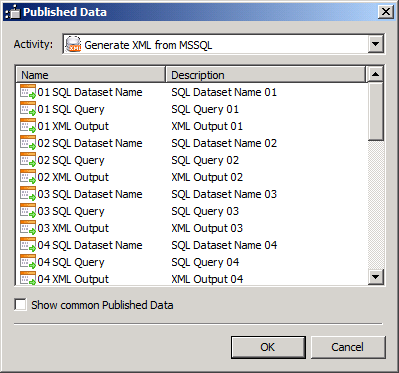
SQL Query and XML Output Count

SQL Database

SQL Server

SQL User ID

Trusted Connection



***Figure S:*** Published Data for “Generate XML from MSSQL”

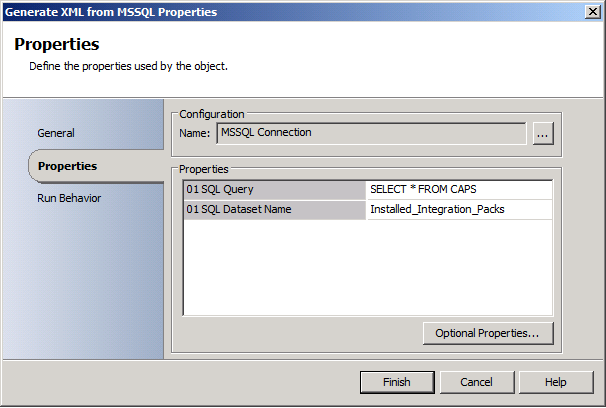
### Functionality

**XML File Output = False (Default)**

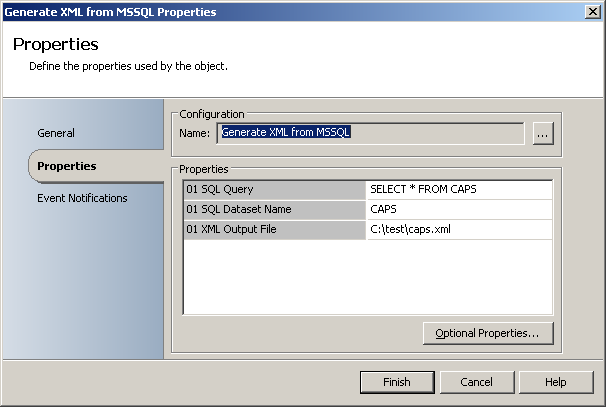
* No XML Output Files will be generated during the execution of the activity.
* XML Output will be limited to Published Data Only
* Simply enter a MSSQL Query, and provide a SQL Dataset Name (for the primary XML Node).

**XML File Output = True (Optional)**

* XML Output Files will be generated during the execution of the activity.
* XML Output will be available as Published Data and the specified XML Output files.
* Simply enter a MSSQL Query, and provide a SQL Dataset Name (for the primary XML Node).



***Figure T:*** Default Configuration for the “Generate XML from MSSQL” Object.



***Figure R:*** Optional Configuration for the “Generate XML from MSSQL” Object.

After this Object’s Execution the XML Output (either just Published Data or Published Data and an XML File) will include the results of the SQL Query entered, formatted as valid XML. This XML Output can then be parsed using the out-of-the-box XML Activities within Orchestrator.