

**Qualys CMDB Integration**  
Developer Documentation  
  
Paul.Senatillaka@fruitionpartners.com

# Table of Contents

Table of Contents 2

Overview 3

Architecture 4

Components 5

Properties 5

Scheduled Job 6

Error Email 6

Import Report 6

REST Web Service 6

Script Includes 7

Script Logs 8

Script Errors 8

Running Jobs 8

Qualys Imported Servers & Applications 8

Modifications to add new mappings 8

Addendum: Sample Qualys Result XML 11

# Overview

Fruition partners developed a Qualys Integration for the ServiceNow CMDB. In this document, we will explain the architecture of the integration and outline the Service Now components that were developed during the implementation in hopes to make maintenance as easy as possible. The integration was built in a modular, well coded and documented manner to emphasis this.   
  
Some features the integration include:

* A properties page for quick configuration and testing
* Custom Report to see the status of newly imported CMDB items.
* Custom logging with optional debug logging and import statistics.
* Configurable Retry Logic – If the integration fails it can wait and retry again.
* Automatic Email and Incident Ticket alerts when the integration fails. Since this is a background process, this provides a feedback mechanism when something goes wrong and needs manual intervention.
* Paginated WS calls to break up the large data set that comes back into many smaller WS calls.
* A “last scan” filter so that each time the integration runs, it filters for only newer Qualys Items with a later scanned date, reducing the load, and allowing for incremental imports.
* Modular Code – The JavaScript that drives the integration was broken out into 3 classes that represent different logical layers of functionality. This promotes code reusability and easier maintenance.
* An application section in the SN navigation bar which links admins to all components of the integration for quick reference.­

The integration uses the Qualys API v2 to do a one way import into ServiceNow of CMDB data. The “Host List Detection” API was used from Qualys and more information about it can be found at the following URLs:

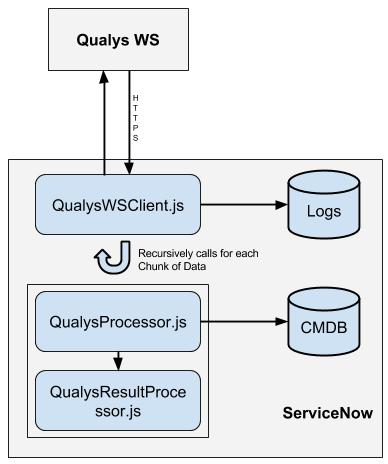
**Quick Reference** - <https://www.qualys.com/docs/qualys-api-v2-quick-reference.pdf>

**User Guide** - <https://www.qualys.com/docs/qualys-api-v2-user-guide.pdf>

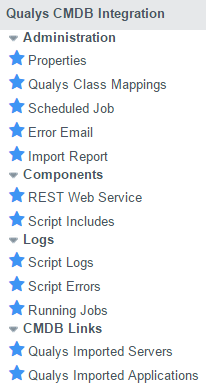
The Integration pulls Asset and related information into SN on a scheduled basis. Currently the integration is setup to map only Servers (including its related data) and Application Cis but is flexible enough to pull any CMDB information as long as the information is returned from Qualys and the mappings are properly setup in the Integration.

# Architecture

The Qualys API is a standard REST Web Service over HTTPS for encryption and uses Basic Auth for Authentication. A scheduled job was setup to import the data on a reoccurring bases. This can be configured through the UI to be run on any schedule desired and is the main trigger point for kicking off the integration. The Scheduled Job calls the Qualys WS Client script include which handles the WS request out to Qualys, in addition to the retry logic and chunking of data . Once the WS Client script has the piece of chunked data it passes it off to the Qualys Processor. The processor takes this data and performs all the parsing and mapping of the data into SN. The data passed from Qualys is XML with Tab Delimited data embedded in a few of the fields in varying column row, row column, etc formats. Because of this, the Qualys Processor makes a call out to the 3rd JavaScript Include, the Result Processor. This processor is a helper utility to parse the differently formatted CSV data embedded in the XML fields and returns a standard array of JSON objects for the Qualys Processor to use in its mapping.



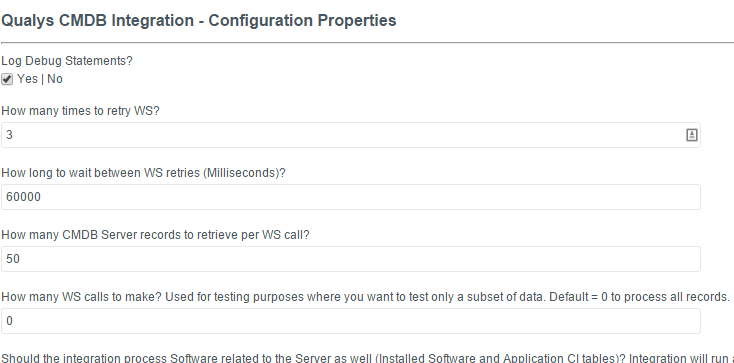
# Components



Pictured above is the Application that was setup in SN for easy access to all of the components for an Admin. In this section we will outline what each component does.

## Properties

This page provides quick access to settings that the scripting in the integration uses. Rather than have to make coding changes to turn something on or off, they are parameterized here for ease of use. The settings all have descriptions on the page. Some notable ones are the ability to change retry logic and to limit the number of WS Calls the integration makes which is good for testing purposes if you only want to import a subset of data.



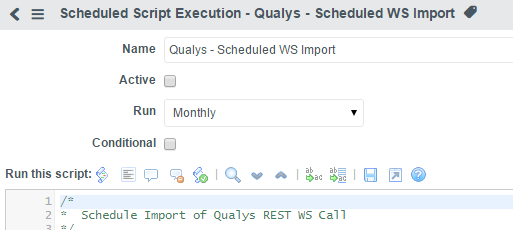
## Class Mappings

This links to a lookup table that provides mapping between hostname codes and what table to map the CI to on import. The hostname code is a 3 character code that is used to match the 4th through 6th character in the Netbios hostname string imported from Qualys. So for example if the lookup class mappings table has an entry for SWT mapping to the Switch CI Table and the scanned host has a Netbios name of BRCSWT14, the record will be imported as a Switch.

If no match is found in the lookup table, the record will be imported to the main CMDB CI table (cmdb\_ci). At a later point, an admin can move this record manually to the appropriate table by setting the appropriate value in the OOB Class field on the CI record. A business rule will automatically move the record.

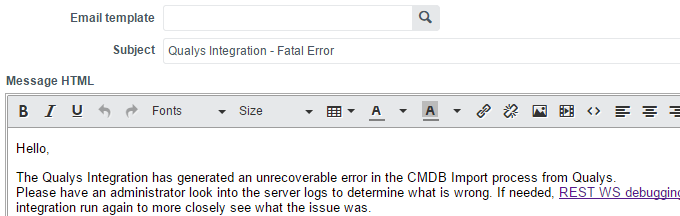
## Scheduled Job

The main entry point that triggers the integration. Use this page to configure run time schedules



## Error Email

The configuration section for the Email Alerts that get sound out when the Integration fails after retry. You can change who the email goes to and the message body here.

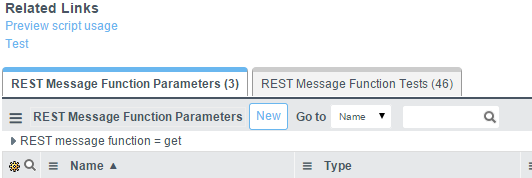


## Import Report

Link to a custom report that shows newly imported CI Servers by the integration.

## REST Web Service

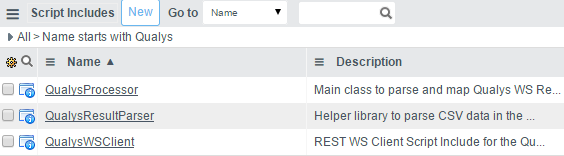
Contains the configuration page of the Web Service Client that the integration uses. Here is where you can configure the URL and Authentication information. The URL parameters that are setup here are dynamically populated later on in the code. Also note, that this OOB Module has a great testing mechanism that allows you to do a quick test that connectivity is working and pull back from raw XML without processing it.

Test Link on page:  


## Script Includes

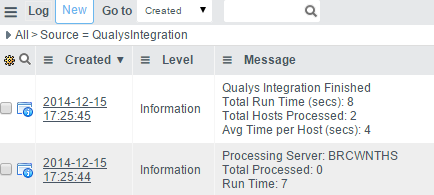
Link to the 3 JavaScript includes that handle the main portion of the integration.

* QualysWSClient - Entry Point for Scripts. Handles WS communication, Retry Logic, Pagination, and logging. Calls other scripts.
* QualysProcessor - Main class to parse and map Qualys WS Response Data. Receives chunked XML from WS Client.
* QualysResultParser - Helper library to parse CSV data in the Result XML field. Receives body of one of the sensor result fields from the XML. Returns a normalized Array of JSON objects. Each Object representing one row.



## Script Logs

Provides a link to the log tables. Logs contain nice stats while the integration is running and even a stat summary when the integration is finished. If Debugging is turned on, more verbose information about each CI being parsed is logged. Normally keep this off until needed.



## Script Errors

Link to Error Log. Things such as uncaught exceptions and failures that are retried are logged here. Good to keep an eye on this every once in a while for any entries, but the Email and Incident alerts should do a good job of warning you first.

## Running Jobs

A link showing all the currently running scheduled jobs on ServiceNow.

Useful to know if an integration is currently running or not and when it was actually kicked off. Shows all running jobs so you have to specifically for the Qualys Integration entry.

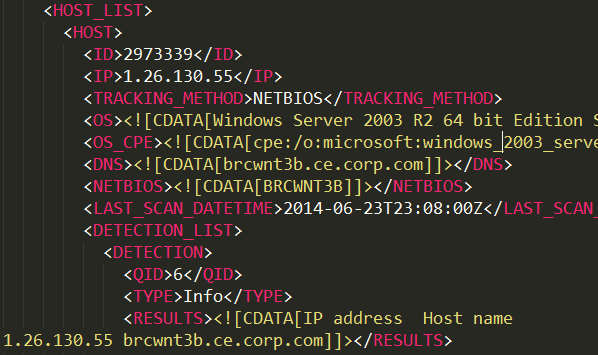
## Qualys Imported Servers & Applications

Direct links to the CMDB tables with a filter only showing Qualys Imported CIs. Handy for navigating through just Qualys CIs.

# Modifications to add new mappings

We designed the integration to be as easy as possible to maintain. Due to functionality requested and the varying formats the CSV data embedded in the XML was returned, we couldn’t use the OOB UI friendly way of setting up Transform maps. This had to be done via JavaScript code which opens up a lot more powerful functionality.

Step 1, in order to pull an additional field of data, you have to update the properties page with the new QID. This gets sent to Qualys as a URL parameter so it knows what servers to pull back and include that sensor data. So for example, pulling back IP addresses would have its own QID which is called a “sensor” in Qualys. In the XML this will return a new entry in the detection list section of the XML. So for example here is a snippet of XML that comes back with one sensor result for IP address:



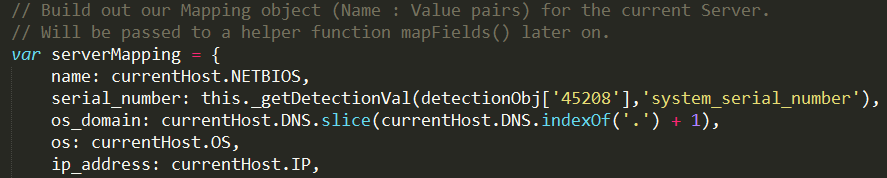
Once you make an update to the Properties page (in effect modifying the URL Parameters), an easy way to test the data coming back would be to go to the REST Web Service link in the admin panel and using the OOB test functionality. More information available at the SN Wiki but essentially it would be as easy as modifying some test parameters, clicking the Test link, and receiving the XML Response back on screen.

Step 2 would be to update the mappings and parsing section of the code. Since all Sensor parsing is done by the helper class QualysResultParser, this is where you should start. The input to the class is a QID number and the content of the <Results> tag for that sensor. Note that the <![CData text you see in the example screen shot above will not be part of the result text that gets passed. It is automatically removed by the WS engine. You would get just the text starting with “IP address ….”.  
In the file there is a Case Select statement that determines what parsing function is mapped to what QID. If there is already a parsing function that can parse the same format of data, then you can reuse it. If not, you can simply write your own function. The function should return the same format of normalized data though. It is an Array of JSON objects. Each object represents one row or record from the CSV. So in the case of the IP sensor it would look like JSON object for 2 IP addresses:



The Json objects are essentially Key:Value pairs. The key can be any name. You just have to use it when adding it to your mappings.

Step 3. Update the mappings section of the QualysProcessor. In the main Process function of this script you can see the existing code build up a “mapping” object (again just Key Value pairs) then pass it to a helper function, MapFields() for a nice clean way to map data in SN. You can just add to this existing pattern. Simply add your new field and value to the object.



So a new entry would look something like this (with placeholder values replaced):



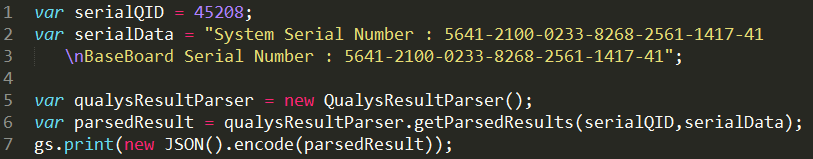
If you are adding new tables as well, you can follow the similar pattern and build a new mapping object and pass it to the mapFields function. In the code the function parameters are detailed with comments. This would include Coelesce fields, Server name, and mapping object.

# Testing

Since each of the script includes are module, an easy way to test when adding new functionality is to test each module separately. You can easy stub some data and call one function or class of the code and print the results to the screen. To do this use the Background Scripts page available in the instance to run server side code. If the high security plugin is turned on you will need the security\_admin role and also need to enable a security session by clicking the lock icon next to your name in the UI. More information available at the SN wiki.

Each of the description fields for the Script Includes contain some sample stub data for unit testing. Feel free to reuse these or write your own test data.

An example test to run would go something like this:



For WS connectivity testing to Qualys, using the OOB testing feature on the REST Web Service page is a great feature too. You can simply stub some data using the UI and click test. Alternatively, you could use command line curl to test connectivity to Qualys although from your machine, not SN.

During testing feel free to add additional debug statements to log more verbose data if needed. This would be something like: this. LogDebug(“Sample Message”);

## Addendum: Sample Qualys Result XML

This is a partial sample of the XML that gets returned from Qualys that the integration parses and maps:

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE HOST\_LIST\_VM\_DETECTION\_OUTPUT SYSTEM "https://qualysapi.qg2.apps.qualys.com/api/2.0/fo/asset/host/vm/detection/host\_list\_vm\_detection\_output.dtd">

<HOST\_LIST\_VM\_DETECTION\_OUTPUT>

<RESPONSE>

<DATETIME>2014-11-13T21:59:59Z</DATETIME>

<HOST\_LIST>

<HOST>

<ID>2973339</ID>

<IP>1.26.130.55</IP>

<TRACKING\_METHOD>NETBIOS</TRACKING\_METHOD>

<OS><![CDATA[Windows Server 2003 R2 64 bit Edition Service Pack 2]]></OS>

<OS\_CPE><![CDATA[cpe:/o:microsoft:windows\_2003\_server::r2:x64:]]></OS\_CPE>

<DNS><![CDATA[brcwnt3b.ce.corp.com]]></DNS>

<NETBIOS><![CDATA[BRCWNT3B]]></NETBIOS>

<LAST\_SCAN\_DATETIME>2014-06-23T23:08:00Z</LAST\_SCAN\_DATETIME>

<DETECTION\_LIST>

<DETECTION>

<QID>6</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[IP address Host name

1.26.130.55 brcwnt3b.ce.corp.com]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>43113</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[HKLM\System\CurrentControlSet\Control\Session Manager\Environment

PROCESSOR\_IDENTIFIER = EM64T Family 6 Model 23 Stepping 10, GenuineIntel]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>45017</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Operating System Technique ID

Windows Server 2003 R2 64 bit Edition Service Pack 2 Windows Registry

Windows 2003 TCP/IP Fingerprint U1751:135

Windows 2003/XP/Vista/2008 MS-RPC Fingerprint

Windows 2003/XP 64 bit Edition NTLMSSP

Windows Server 2003 R2 3790 Service Pack 2/Windows Server 2003 R2 5.2 CIFS via TCP Port 445

cpe:/o:microsoft:windows 2003 server::r2:x64: CPE]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>45039</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Host Name Source

BRCWNT3B.CE.Corp.com NTLM DNS

brcwnt3b.ce.corp.com FQDN

BRCWNT3B NTLM NetBIOS]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>45099</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Interface: HP NC373m Multifunction Gigabit Server Adapter IP Address: 0.0.0.0

Interface: HP NC373m Multifunction Gigabit Server Adapter IP Address: 1.126.110.106

Interface: HP NC373m Multifunction Gigabit Server Adapter IP Address: 0.0.0.0

Interface: HP NC373i Multifunction Gigabit Server Adapter IP Address: 0.0.0.0

Interface: HP NC373i Multifunction Gigabit Server Adapter IP Address: 0.0.0.0

Interface: HP NC373m Multifunction Gigabit Server Adapter IP Address: 1.126.110.106]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>70022</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Description Version TCP Ports UDP Ports HTTP Ports NetBIOS/CIFS Pipes

Compaq Remote Monitor Service 1.1 \pipe\cpqrcmc

DCE Endpoint Mapper 3.0 135 \PIPE\epmapper

DCE Remote Management 1.0 \PIPE\epmapper

DCOM OXID Resolver 0.0 135 \PIPE\epmapper

DCOM Remote Activation 0.0 135 \PIPE\epmapper

DCOM System Activator 0.0 135 \PIPE\epmapper

Microsoft Distributed Transaction Coordinator 1.0 1026

Microsoft Event Log Service 0.0 \PIPE\eventlog

Microsoft Local Security Architecture 0.0 \PIPE\lsarpc

Microsoft Network Logon 1.0 \PIPE\NETLOGON

Microsoft Registry 1.0 \PIPE\winreg

Microsoft Scheduler Control Service 1.0 \PIPE\atsvc

Microsoft Security Account Manager 1.0 1025 \PIPE\samr, \PIPE\lsass

Microsoft Server Service 3.0 \PIPE\srvsvc, \PIPE\wkssvc

Microsoft Service Control Service 2.0 \PIPE\svcctl

Microsoft Spool Subsystem 1.0 1025 \PIPE\lsass

Microsoft Task Scheduler 1.0 \PIPE\atsvc

Microsoft Workstation Service 1.0 \PIPE\wkssvc

Simple Mail Transfer Protocol 2.0 1090 \PIPE\INETINFO

(Unknown Service) 1.0 135

(Unknown Service) 0.0 135

(Unknown Service) 2.0 135

RPC ROUTER SERVICE 1.0 \PIPE\ROUTER

Unimodem LRPC Endpoint 1.0 \pipe\tapsrv]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>70030</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Device Name Comment Type Label Size Description

IPC$ Remote IPC -2147483645

C$ Default share -2147483648 298 MB Disk (mounted)

gdrive 0 G drive 2999 GB Disk (mounted)

ADMIN$ Remote Admin -2147483648 7 GB Disk (mounted)

F$ Default share -2147483648 F drive 1999 GB Disk (mounted)

G$ Default share -2147483648 G drive 2999 GB Disk (mounted)

restore Restore Share 0 60 GB Disk (mounted)

BackupEAS 0 G drive 2999 GB Disk (mounted)

MicroMainDBBkps 0 G drive 2999 GB Disk (mounted)

D$ Default share -2147483648 7 GB Disk (mounted)

E$ Default share -2147483648 60 GB Disk (mounted)]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>90107</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[HKLM\Software\Microsoft\Windows NT\CurrentVersion

ProductName = Microsoft Windows Server 2003 R2

CurrentVersion = 5.2

HKLM\SYSTEM\currentControlSet\Control\ProductOptions

ProductType = ServerNT

ProductSuite = {&quot;Enterprise&quot;, &quot;Terminal Server&quot;}]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>90235</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[Display Name Display Version Install Date Publisher Language

ATI Display Driver 8.24.3-060405a-042344C-HP Not Found Not Found Not Found

Microsoft Internationalized Domain Names Mitigation APIs Not Found 20080303 Microsoft Corporation Not Found

Windows Internet Explorer 7 20061107.210147 20080303 Microsoft Corporation Not Found

GDR 4060 for SQL Server Integration Services 2005 (64-bit) ENU (KB2494113) 9.3.4060 20110708 Microsoft Corporation Not Found

GDR 4060 for SQL Server Notification Services 2005 (64-bit) ENU (KB2494113) 9.3.4060 20110708 Microsoft Corporation Not Found

GDR 4060 for SQL Server Database Services 2005 (64-bit) ENU (KB2494113) 9.3.4060 20110708 Microsoft Corporation Not Found

GDR 4060 for SQL Server Tools and Workstation Components 2005 (64-bit) ENU (KB2494113) 9.3.4060 20110708 Microsoft Corporation Not Found

Security Update for Windows Server 2003 (KB923561) 1 20100224 Microsoft Corporation Not Found]]></RESULTS>

</DETECTION>

<DETECTION>

<QID>105504</QID>

<TYPE>Info</TYPE>

<RESULTS><![CDATA[HKLM\SYSTEM\CurrentControlSet\Services\smstsmgr is missing]]></RESULTS>

</DETECTION>

</DETECTION\_LIST>

</HOST>

</HOST\_LIST>

<WARNING>

<CODE>1980</CODE>

<TEXT>1000 record limit exceeded. Use URL to get next batch of results.</TEXT>

<URL><![CDATA[https://qualysapi.qg2.apps.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&show\_igs=1&qids=90235,90107,45017,45017,6,45039,70030,43113,105054,45099,105059,70022,105060,105504,105001&id\_min=3700609]]></URL>

</WARNING>

</RESPONSE>

</HOST\_LIST\_VM\_DETECTION\_OUTPUT>