

How to upgrade from V3 to V4 using V3V4 Adapter

By OESIS team, 2015

Purpose

This document is to help OESIS V3 customers upgrade to OESIS V4 using the Interface Adapter. The upgrade process is intended to be a smooth drop-in solution if the customer is following the recommended practices for OESIS V3 integration. Otherwise, depending on the implementation details, some limitations would be introduced to the integration after upgrading.

Deployment package

OESIS V4 is completely rebuilt from the ground up as a successor of OESIS V3. It is designed to be an OESIS V3 replacement with new modern technology. Therefore, everything in the V4 deployment is V4 specific. No part of OESIS V3 exists in the V4 deployment, neither the native V4 package nor the V3V4 Adapter package.

Native OESIS V4 package contains 2 components:

Engine Package, OESIS_V4_4_X_X_Xr.zip

Data Package, OESIS_V4_offline_resource_4_X_X_Xr.zip

There are 5 different flavors of V3V4 Adapter available for customer to choose:

OESIS_V3V4_Adapter_vc8_4_X_X_X.zip

OESIS_V3V4_Adapter_vc9_4_X_X_X.zip

OESIS_V3V4_Adapter_vc10_4_X_X_X.zip




















OESIS_V3V4_Adapter_vc11_4_X_X_X.zip

OESIS_V3V4_Adapter_vc12_4_X_X_X.zip

Note: since the Adapter approach is designed to be a temperately solution to help existing OESIS V3 customer to move to OESIS V4 within a short period of time, please do not expect any new CRT versions of the Adapter package to be supported in future releases.

In order to setup a deployment package with similar features to that of OESIS V3, the customer would need to download the OESIS V4 Engine Package and one of the V3V4 Adapter packages. Here is an example of deployment steps of VC9 package:

1. Download OESIS_V4_4_X_X_Xr.zip and OESIS_V3V4_Adapter_vc9_4_X_X_X.zip from our portal. <https://portal.opswat.com/en/product-categories/oesis-framework>.
2. Extract both of the packages, and create a temp folder to host the deployment files.
3. Copy libwaapi.dll, libwaheap.dll, libwautils.dll from the following location to the temp folder.
OESIS_V4_4_X_X_Xr\bin\detection\- 4. Copy libwalocal.dll, wa_3rd_party_host_32.exe, wa_3rd_party_host_64.exe (optional, only needed if it is running on an x64 system) from the following location to the temp folder.
OESIS_V4_4_X_X_Xr\bin\manageability\- 5. Copy all files under OESIS_V3V4_Adapter_vc9_4_X_X_X\bin\- 6. Copy the pre-downloaded customer specific license.cfg file into the temp folder.
- 7. After the above 6 steps, the temp folder should contain 19 files as below:

 bridgetest.exe	10/20/2015 11:16 ...	Application	471 KB
 CoreUtils.dll	10/20/2015 11:16 ...	Application extens...	33 KB
 libbridge.dll	10/20/2015 11:16 ...	Application extens...	366 KB
 libwaapi.dll	10/21/2015 9:51 A...	Application extens...	518 KB
 libwaheap.dll	10/21/2015 9:51 A...	Application extens...	75 KB
 libwalocal.dll	10/21/2015 9:51 A...	Application extens...	752 KB
 libwaresource.dll	10/20/2015 11:18 ...	Application extens...	1,431 KB
 libwautils.dll	10/21/2015 9:51 A...	Application extens...	865 KB
 license.cfg	2/11/2015 10:19 A...	CFG File	7 KB
 mfc90u.dll	9/5/2015 1:44 AM	Application extens...	1,136 KB
 Microsoft.VC90.CRT.manifest	9/5/2015 1:44 AM	MANIFEST File	1 KB
 Microsoft.VC90.MFC.manifest	9/5/2015 1:44 AM	MANIFEST File	1 KB
 msvcm90.dll	9/5/2015 1:44 AM	Application extens...	220 KB
 msvcp90.dll	9/5/2015 1:44 AM	Application extens...	556 KB
 msucr90.dll	9/5/2015 1:44 AM	Application extens...	641 KB
 OESISCore.dll	10/20/2015 11:16 ...	Application extens...	14 KB
 OesisDiagnose_bridge.exe	10/20/2015 11:16 ...	Application	352 KB
 wa_3rd_party_host_32.exe	10/21/2015 9:52 A...	Application	578 KB
 wa_3rd_party_host_64.exe	10/21/2015 9:52 A...	Application	660 KB

8. Launch the bridgetest.exe for evaluating or testing OESIS V3V4 Adapter feature. The bridgetest.exe and OesisDiagnose_bridge.exe are not required files.

V3V4 Adapter licensing

Similar to OESIS V4, the V3V4 Adapter license contains 2 components, a license.cfg file and a passkey. The customer can download the specific license file and passkey from <https://portal.opswat.com> after logging in.

Product	License	Support
+ Metascan Online		Request Support
+ Gears		Request Support
- OESIS	Trial Account	Request Support
Increase Limits		
License File	Download license file	
Pass-key	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
Expiration dates for modules	- Name: Expiration:	
Online mode	Yes	

The license.cfg and passkey are pair to pair match, therefore, customer would need both of them to initial OESIS V3V4 Adapter. The license.cfg need to sit under the same folder as libwaapi.dll. And the passkey would require to be passed in via the OESIS_SetLicense call. Below is a sample code snip for the V3V4 Adapter set license call which is actually the same as V3:

```
//initialize license
typeCharStr lic = L"OESIS_V4_Passkey";

OESIS_SetLicense(lic.c_str());
OESIS_Init();
```

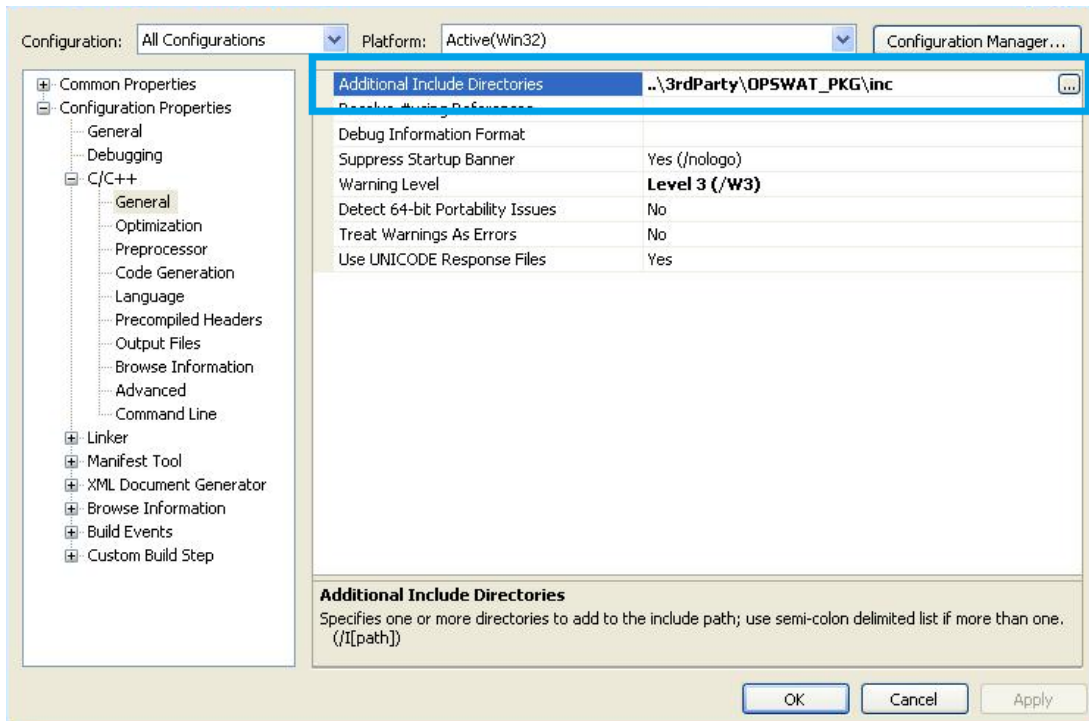
Same as OESIS V3, a successful OESIS_Init call must be made before most OESIS function calls will work. And every call to OESIS_Init must have a call to OESIS_Deinit. Initialization of the internals are held with reference counts that are incremented and decremented by these 2 functions. Once the reference count has hit 0, the internals will be deinitialized.

Configuring the integration project

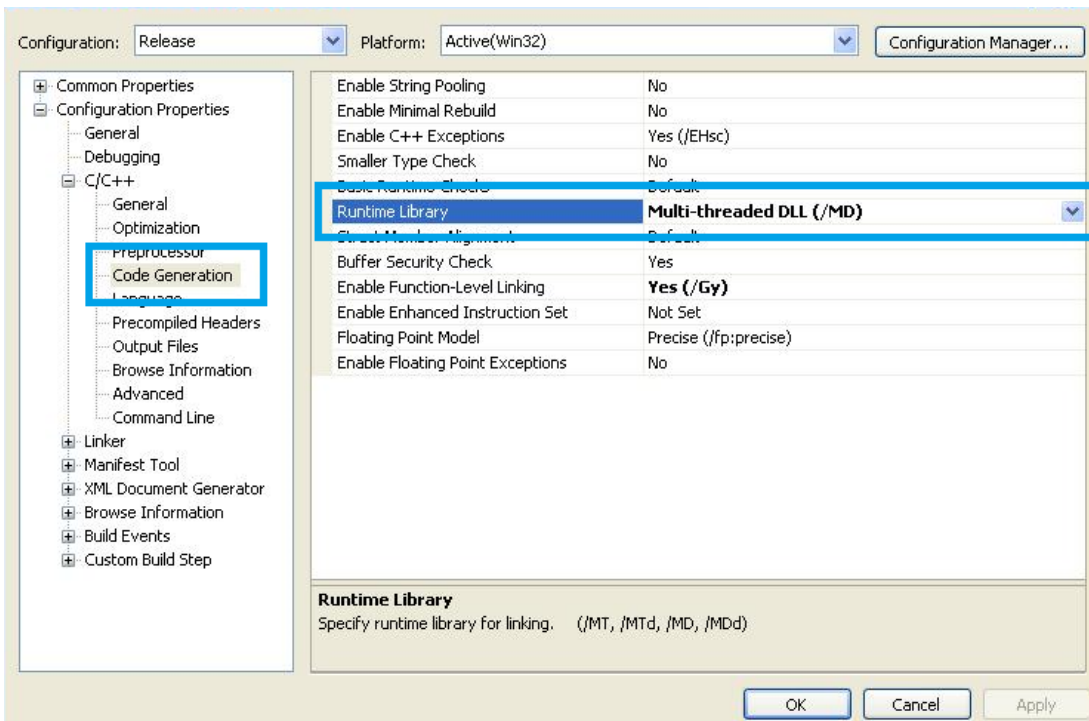
OESIS V4 does not have support for full static libs and neither does the V3V4 Adapter. If the customer was static linking with OESIS V3 (using the static unified builds), then in order to upgrade to V3V4 Adapter, customer would need to switch to use dynamic linking instead. Here is a step by step guide for reference:

1. The first step of configuring the project is to add a reference to our include directory.

Note: this is an optional step, if you prefer to use a full path in the #include statements.



2. After that, customer would need to setup the Code Generation setting correctly.
 - 1) Open the Project Properties
 - 2) Navigate to Configuration Properties -> C/C++ -> Code Generation.
 - 3) In the Debug Configuration, set the Runtime Library to "Multi-threaded Debug DLL (/MDd)"
 - 4) In the Release configuration, set the Runtime Library to "Multi-threaded DLL (/MD)"



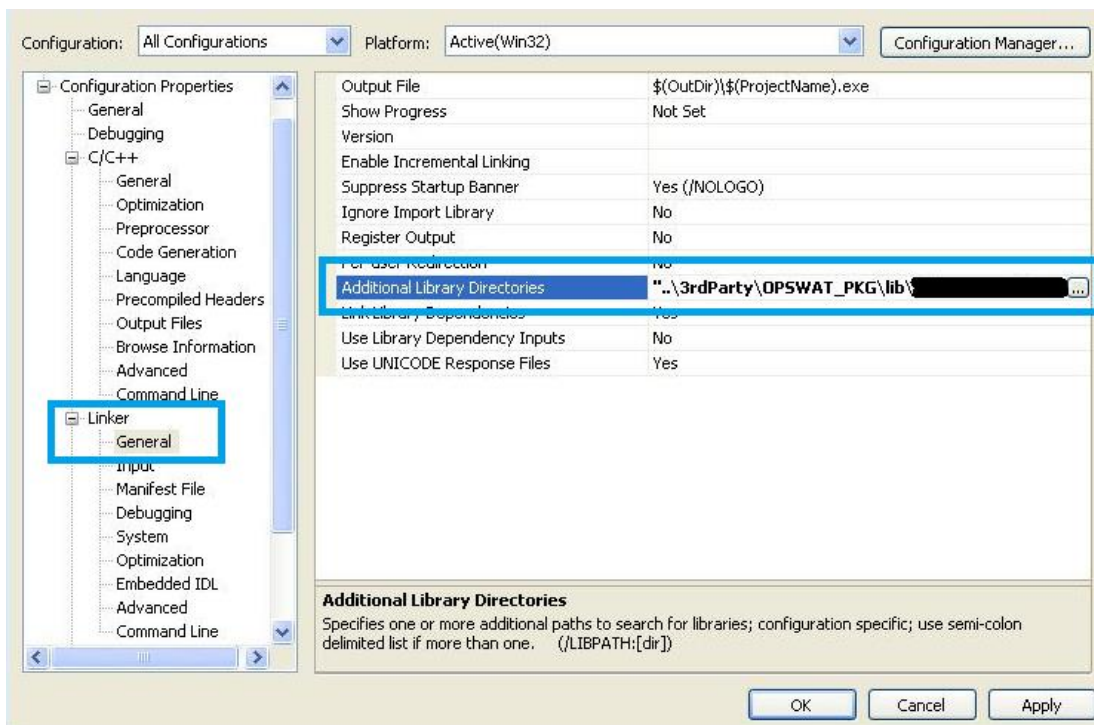
3. Then customer would need to decide whether to do implicit linking, or use explicit runtime linking (via LoadLibrary/GetProcAddress).

a. Implicit (dynamic) linking

This gives customer simpler code, but will cost the ability to update OESIS binaries without shutting down the customer application. To do implicit linking, customer would need to add OESISCore.lib (or OESISCore64.lib) to the project. NOTE: libbridge.lib can also be used in place of OESISCore(64) and CoreUtils as it supplies the same exports.

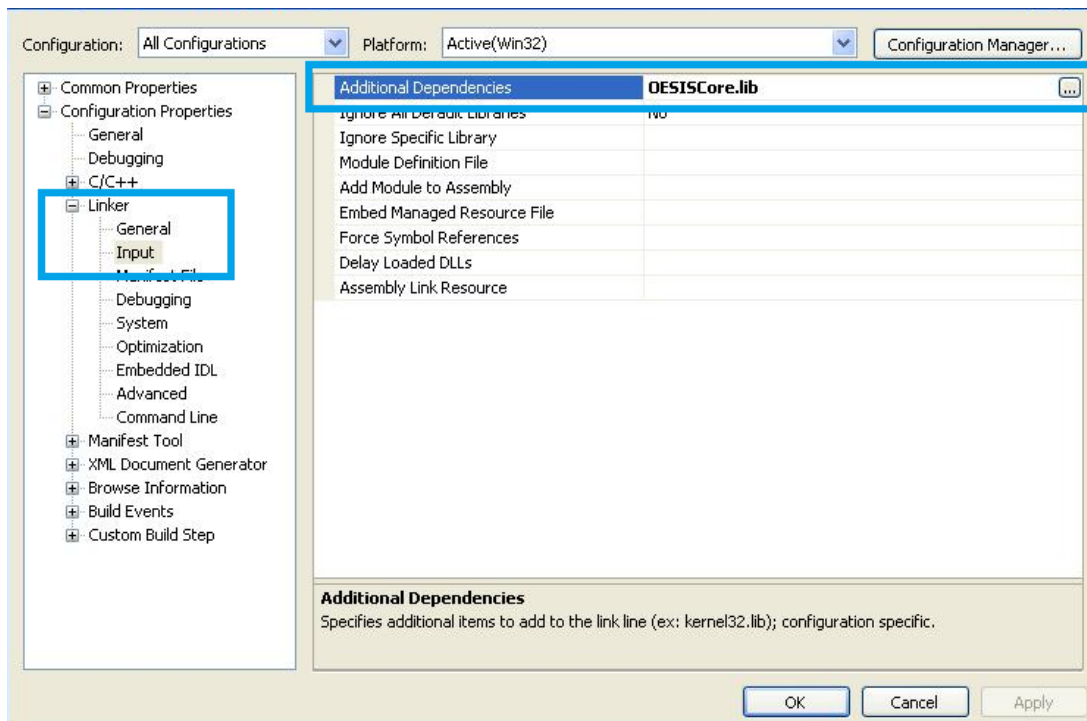
1) Open the Project Properties window again. Navigate to Configuration Properties -> Linker -> General.

2) In the Additional Library Directories setting, add the path to where OESIS libs can be found.



3) Navigate to Configuration Properties -> Linker -> Input

4) Under Additional Dependencies, add OESISCore.lib (or OESISCore64.lib)



Note that implicit linking could also be done through #pragma statements, but the concepts are the same.

b. Explicit (runtime) linking

This will give customer the greatest flexibility in deployment options and features, however it can complicate the integration code and calls made to the framework. If using Runtime Linking, there are no additional settings which need to be made to the linker.

Interacting with Microsoft runtime

The native OESIS V4 is statically built in with Microsoft runtime, therefore, the native Engine Package of OESIS is completely VC independent. The runtime dependency is introduced for the V3V4 Adapter only and the Adapter dynamically links using the CRT (/MD option).

As mentioned earlier in this document, OESIS now provides VC8, VC9, VC10, VC11 and VC12 version of the Adapter packages. Depending on which CRT the customer application is using, the customer should pick the corresponding V3V4 Adapter package.

It is currently a requirement that the customer application dynamically links against the same Microsoft runtime as V3V4 Adapter. This will simplify the customer deployment, as well as avoid potential issues with memory heap allocation.

Note: If the customer is statically linking the runtime then there may be possibilities to make this work, but it adds additional complication and potentially more upgrades to the application code. Since OESIS

and the customer application may not be linked against the same instance of the runtime there may be needs for using `OESIS_FreeTypeProperty` to release the memory, where there was no need in the past. The V3 example code for `OESIS_InvokeMethod` show examples of using `OESIS_FreeTypeProperty` with dynamic allocation, but this may also be necessary with pass-by-reference variables as memory ownership may transfer to a separate runtime heap with static linking.

Configuring the OESIS options

Unfortunately, the `OESIS_Config` function from OESIS V3 is mostly unsupported due to the options no longer being applicable to OESIS V4. Any customer who is using `PROP_CORE_RT_IMPL_DIR` option to load all the binaries from a location other than the process location will now need to make a WINAPI call to `SetDllDirectory()` to specify the library location before initializing the OESIS API. See [msdn](#) for reference.

In V3V4 Adapter, all the adapter and V4 specific configuration is done through one or more `.dat` files. These files could be found under the `OESIS_V3V4_Adapter_vcX_4_X_X_X\config` folder. Each of them acts as an enable/disable flag for certain features of the V3V4 Adapter. Simply including them in the package with the binaries will enable the options.

v3greedy_detection.dat

Enables the greedy detection logic when using V3 data mapping. If this file is present and the `v3map` lock file is present then detection results will include all possible detection results for mapping from V4 to V3.

When V4 detects a product (signature), it will map it back to a V3 product ID. In some cases there could be more than one option and instead of using the regular deterministic mapping, the Adapter will return all possible options as detected products. For some products this may produce results that look like duplicate detection, but ensures that no possible mappings are missed.

v3map.dat

This enables V3 data compatibility mapping. It is required to use the `v3greedy_detection`. If this file is present, V3 data mapping will be used internally by the V3V4 Adapter. All the product information, including product id, product name and vendor name, would be mapped to the original V3 data before returning. This should be used if V3 product data is being passed to the Adapter from legacy data sources that cannot be updated to use V4 ids and names.

v3mockup.dat

This is used for testing the API with provided testing data. If this file is present, `mockup_mode` configuration in native OESIS V4 will be used with the V3V4 Adapter. It allows for calls to be made and return fixed data for those calls to test certain outputs. All data returned is defined by

a mockup.json file. If interested in this functionality there is a SDK Guide in the native OESIS V4 documentation which explains in more detail.

v3service.dat

Enables service mode in native OESIS V4. If this file is present, service_mode configuration will be used with the V3V4 Adapter. This should be used if the customer integration runs in a service context under the SYSTEM account on Windows.

Interface compatibility and data compatibility

Both OESIS V4 and V3V4 Adapter are built using data driven design with the target to answer the same device compliance validation questions to the industry as OESIS V3. It is a brand new technology with the foundation to be extended and scaled in the future. With all these additional features, neither OESIS V4 nor V3V4 Adapter is going to be a fully complete, backward compatible solution to the legacy V3. However, V3V4 Adapter does provide, to some extent, a drop in replacement option to customers who are using limited OESIS functions for popular security products and follow the best integration practices from V3. For the rest of our customers who have specific use cases, part of the integration program may require modification even with the V3V4 Adapter.

Interface compatibility

The V3V4 Adapter maintains around 85% of the APIs from OESIS V3. Since the V3V4 Adapter does not have static libraries and the OESIS V3 libraries no longer exist, the static interface functions, such as OESIS_InitStaticAntivirus, are no longer relevant and so no longer exist.

Some security interfaces and OESIS_Invoke calls are not supported and some only support certain keys for their input and output arguments. Please refer to the [OESISV4_upgradeGuide](#) for the API differences in detail.

The Callbacks and Timeouts interfaces are not supported in the current version of V3V4 Adapter. However, they are on the roadmap to be supported (in some capacity) in a future release.

Data compatibility

The V3V4 Adapter internally leverages OESIS V4 for actual product implementations and supports 3000+ applications and 5000+ methods. All this data is coming from the native OESIS V4 package and therefore this data is V4 data, which includes the product id, product name, vendor name and etc. In order to have V3V4 Adapter return these information in OESIS V3 data, the customer could enable the data compatibility using the v3map.dat as mentioned above.

OESIS V3 is built at 2006 and it contains a huge backpack of data for the legacy 3rd party products, such as implementations for Kaspersky Internet Security 11.x which is from 2009. For products of this age, OESIS V4 and V3V4 Adapter will not provide support since there is no reason to increase the package size for information that is no longer relevant. The general guideline of the V3V4 Adapter is to support major vendor products which are released after 1/1/2013. For any products that are released on or before 2012, it would require on-demand requests to support them in OESIS V4. Customers would need to contact the OESIS team directly to request support for certain products that are not included under the general policy.

Parsing OESIS support chart

V3V4 Adapter supports a sub-set of the legacy V3 categories only. Because of the growing nature of the device security industry, some of the categories are no longer part of the market standard requirements. In V3, OESIS supports 19 different security categories, but the V3V4 Adapter only provides support for 14 of them. Two new categories, Cloud Storage Drive and Unclassified are introduced to better fit in recent use cases. Here is a comparison table for detail:

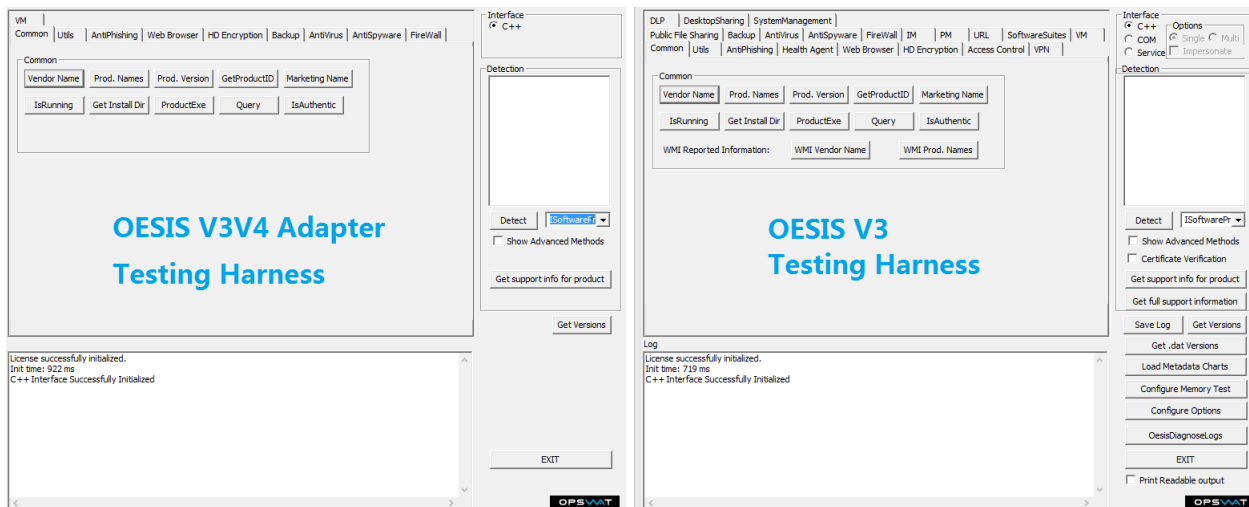
Category name	Supported in V3	Supported in V3V4 Adapter
APSDK	Yes	Yes
ASSDK	Yes	Yes
AVSDK	Yes	Yes
BCSDK	Yes	Yes
DEVSDK	Yes	No
DLPSDK	Yes	Yes
DSSDK	Yes	No
FWSDK	Yes	Yes
HASDK	Yes	Yes (planned)
HDSDK	Yes	Yes
IMSDK	Yes	Yes
OBIS	Yes	Yes (supported as BRSDK)
P2PSDK	Yes	Yes
PMSDK	Yes	Yes
SMSDK	Yes	No
Software Suite	Yes	No
URLSDK	Yes	No
VMSDK	Yes	Yes
VPNSDK	Yes	Yes
CLDSDK	No	Yes
UNCSDK	No	Yes

Similar to OESIS V3, the Adapter provides support charts in XML format with xsl style files. The XML schema is 99% similar to the legacy V3 ones, depending on how the customer parsing code is built, there is a very good chance that the support charts could be a drop in replacement for the V3 support charts.

Verifying OESIS APIs during integration

In OESIS V3, the customer would have 2 major debugging tools to verify the OESIS API functions, V3Testing Harness.exe and OesisDiagnose_V3.exe. Replacements for both of them could be found within the V3V4 Adapter package still.

The bridgetest.exe is designed to perform individual API call validation for customers. Similar to V3Testing Harness.exe, it makes API calls to OESIS on demand, function by function. They provide a similar looking UI as well.



OesisDiagnose_bridge.exe contains similar functionality to OesisDiagnose_V3.exe. It collects some system information as well as user permissions, and then calls each of the OESIS “get” functions for each of the detected applications. It generates two output files, one in txt (.log) format, and the other in XML (.xml) format.

Modifying automation programs which download OESIS packages

Some customers are downloading OESIS from <https://portal.opswat.com> manually. This step is definitely still supported and it is always going to work as a fallback. For the customers who use an automation program to download the latest OESIS package periodically, a modification to those programs will be required to maintain a similar mechanism.

In OESIS V3, the customer program would go to <https://admin.opswat.com/latest.xml> to pull down the xml file, then parse the content to look for download links of the latest OESIS packages. In both OESIS V4 and V3V4 Adapter use case, instead of the origin url, the customer program would need to be switched to https://software.opswat.com/OESIS_V4/OesisPackageLinks.xml to get similar information.

```
<OesisPackageLinks SchemaVersion="1.0" Date="2015-01-06">
  <Platform Name="Windows">
    <Releases Name="OESIS Local V4">
      <LatestRelease Version="4.2.275.0" Date="2015-10-20">
        <Package Name="Engine Package" Version="4.2.275.0" Link="https://portal-downloads.s3.amazonaws.com/OESIS_V4_4_2_275_0r.zip"
          MD5="D0C1B0A22048033E100FE052BD255569" sha256="96AC94F79673D9C97CA6D4F5882F3376C60A5DCC04B16EE91E0112B2D86839AE"/>
        <Package Name="Data Package" Version="4.2.275.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V4_offline_resource_4_2_275_0r.zip" MD5="FCC7934ED9C81669F47EAB1303543F80"
          sha256="828DBAEF29BB3DCAB41D42B436C7050CA8FAA52D222F1A07B090B31A89C82AC9"/>
        <Package Name="OESIS V3 to V4 Interface Adapter VC8" Version="4.2.645.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V3V4_Adapter_vc8_4_2_645_0.zip" MD5="08E7C3F0A52C8B57E073208F801D796F"
          sha256="5F8EF1B3E773F0F5A4E914C776915AE3C147803D15D991EFBE7891912E3C3825"/>
        <Package Name="OESIS V3 to V4 Interface Adapter VC9" Version="4.2.645.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V3V4_Adapter_vc9_4_2_645_0.zip" MD5="550BFC5005F673024DE823CA4716219C"
          sha256="3AAEDCC1D148387DBA994D81860EA47A2D63D28AC86725FEA092E47F32A5F464"/>
        <Package Name="OESIS V3 to V4 Interface Adapter VC10" Version="4.2.645.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V3V4_Adapter_vc10_4_2_645_0.zip" MD5="32729B7C35BBF832369FE219071EE95A"
          sha256="5E4869DA4BDFBA495CDC1DFBFF8E4B1993EA4FD3B58997C4646FA178CDE8CB73"/>
        <Package Name="OESIS V3 to V4 Interface Adapter VC11" Version="4.2.645.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V3V4_Adapter_vc11_4_2_645_0.zip" MD5="C8F28258B214BFE1044FAAF2B261EF85"
          sha256="584375CF71F17C9BD4A24781F89F790B0958AF9823B1A360C8C61221020E78EA"/>
        <Package Name="OESIS V3 to V4 Interface Adapter VC12" Version="4.2.645.0" Link="
          https://portal-downloads.s3.amazonaws.com/OESIS_V3V4_Adapter_vc12_4_2_645_0.zip" MD5="C36C294665A957A17F1DC113D43CE55F"
          sha256="8CF2DF8DE4C6136638B15632231A413DC2447E239956939184798A3747B97337"/>
      </LatestRelease>
    </Releases>
  </Platform>
</OesisPackageLinks>
```

Within the OesPackageLinks.xml file, there are direct links to download the OESIS V4 packages as well as the V3V4 Adapter packages. Each of the package entries contain MD5 and sha256 hash values, which customers could potentially use to verify whether the package download succeeded or not.

Summary

OESIS V4 and V3V4 Adapter are new technologies which were built in 2014, to replace an aging V3 which is already 8+ years old. Some of the legacy functions and product support are dropped by design to better achieve the performance and flexibility of OESIS SDK. When upgrading from OESIS V3 to V4, native OESIS V4 upgrade would require a completely new integration which is estimated to take around 2 weeks of customer engineering time including testing. While V3V4 Adapter is designed to help legacy V3 customers, as a temporary solution, to move away from V3 within a short period of time. At the time of this document creation, there is no solid EOL date defined for V3V4 Adapter, however native OESIS V4 is the suggested integration practice which gives out the best API performance and the richest feature support. V3V4 Adapter should not be treated as a long term solution for any new customer integration action.