SAP Virus Scan Interface

Note

This PDF document contains the selected topic and its subtopics (max. 150) in the selected structure. Subtopics from other structures are not included.

Table of content

1 SAP Virus Scan Interface

Description

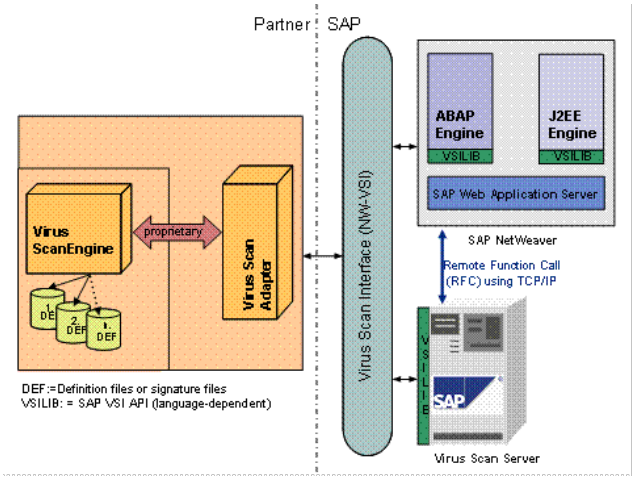
Virus scanning should be performed every time potentially polluted data is imported through input channels into the SAP system. Possible input channels are:

- File upload from front end PCs or file system on the application server
- File upload using the Internet
- Document exchange by RFC, XML, XI

What Do I Get from the SAP NetWeaver Platform?

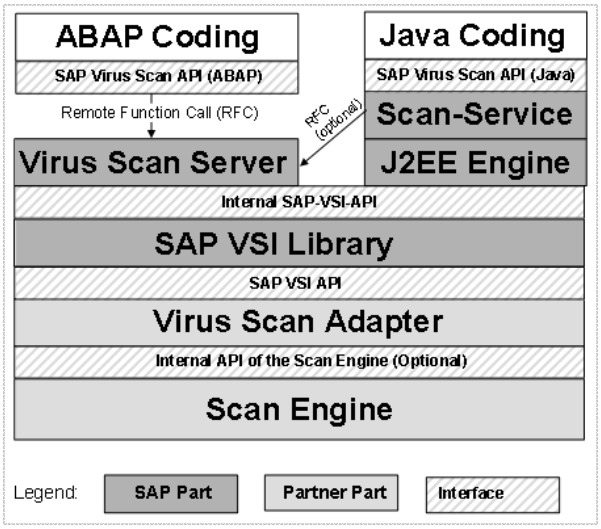
The architecture of the Virus Scan Interface (VSI) allows you to combine different products, systems, and platforms to scan your applications for viruses. On the SAP side, different VSILIB layers are used to include the ABAP and Java worlds, and to deal with platform dependencies in the integration of the virus scan interface. For more information, see Architecture of the Virus Scan Interface.

Elements of the Virus Scan Interface



The graphic below clarifies the layer structure of the SAP Virus Scan Interface (SAP VSI API) and shows which parts are delivered by SAP, and which by the relevant partners.

Software Layers of the Virus Scan Interface



The SAP Virus Scan Interface (SAP VSI API):

- Is accessed by partner products directly with the scan engine or indirectly using a separate virus scan adapter.
- Contains the functions required to configure and to initialize the scan engine.
- Provides the parameters and data for every virus scan.
- Processes the check result.

ABAP or Java application programs start virus scans with dedicated classes and methods of the SAP virus scan interface, which, in turn, call a virus scan server or the AS Java directly using RFC.

What Do I Need to Do?

To be able to use SAP's Virus Scan Server, you must maintain data in the Implementation Guide.

1. **Scanner Groups**
   Scanner groups bundle together one to many Virus Scan Servers or Business Add-In implementations (BADI VSCAN_INSTANCE) of scan engines. At this level, you can maintain a set of configuration parameters that contain initialization parameters for the Virus Scan Servers. Be aware that Customizing settings are cross-client since the controlled entities (Virus Scan Servers or BADIs) are cross-client. For detailed information, see Defining Scanner Groups.
2. **Virus Scan Servers**
   The Virus Scan Server is an RFC server, that holds the connection to the scan engine using the certified Virus Scan Adapter. It runs either as part of the application server as a separately started executable or as a standalone program and communicates with the application servers using RFC. A Virus Scan Server entry always belongs to a single scanner group. The detail screen of the Virus Scan Server maintenance shows a summary of the scan engine type and the supported features. The Server may be manually started, stopped and re-initialized from Customizing. To set up Customizing, see Defining Virus Scan Servers.
   You can integrate the Virus Scan Interface into your own developments using the methods from class CL_VSI. For detailed information, see Integrating the Virus Scan Interface into Customer Developments. See also the following SAP notes to clarify questions concerning the Virus Scan Interface: SAP Note 817623, SAP Note 786179.
3. **Virus Scan Profiles**
   You can create separate Virus Scan Profiles for each application in Customizing, mapping Scanner Groups to Virus Scan Servers or customer-defined Business Add-Ins.
   Profiles need to be in the separate namespace delivered by SAP
   `/<Name of package>/<Freely assignable name>`
   Profiles can be:
   - Selected as the Default Profile
   - Activated or deactivated
   - Assigned to reference profiles

   For more information, see Defining Virus Scan Profiles.

If Customizing is set up for the active Virus Scan Interface, you may integrate the VSI into your application using the class CL_VSI. To perform a virus scan for a byte sequence check in an ABAP program, you need to perform the following steps:

1. Use the static method GET_INSTANCE to generate an instance of the virus scan interface, which is based on the specified virus scan profile. Every application that implements VSI should use its own virus scan profiles so that the virus scan functions can be activated and deactivated for each application.

   Example

```
* Retrieve scanner instance for my application
 DATA:
 lo_scanner TYPE REF TO cl_vsi.

 CALL METHOD cl_vsi=>get_instance
 EXPORTING
 if_profile = '/MYPACKAGE/MYFUNCTION'
 IMPORTING
 eo_instance = lo_scanner
 EXCEPTIONS
 profile_not_active = 1
 OTHERS = 2.

 CASE sy-subrc.
 WHEN 0.
 "OK
 WHEN 1.
* The system administrator has disabled virus scanning for
* my application '/MYPACKAGE/MYFUNCTION'.
* What must happen now depends on whether virus scanning
* is optional or mandatory for your application.
* In the first case, you can ignore it, in the second
* case
* you must react with an error. This exception has a
* SY-Message that leads the user to the right place in
* customizing.

 WHEN OTHERS.
* This situation is always an error (misconfiguration)
* of the Virus Scan Interface. It must be reported.
* Use the SY-Message that always accompanies the
* exception.

 ENDCASE.
```

2. Now you can perform virus scanning of data that are present in XSTRING objects.

   Example

```
* Retrieve scanner instance

 CALL METHOD lo_scanner->scan_bytes
 EXPORTING
```

```
if_data = lf_data
if_do_clean = 'ABAP_TRUE'
IMPORTING
ef_scanrc = lf_scanrc
EXCEPTIONS
OTHERS = 1.
```

The result of the scan is one of these three situations:

- **The return code LF_SCANRC has the value CL_VSI=>CON_SCANRC_OK.** This indicates that the scan task was successfully performed and no infection was found.
- **The return code LF_SCANRC has another value.** Some of the most prominent error codes are CON_SCANRC_… attributes in class CL_VSI. You can use the method CL_VSI=>GET_SCANRC_TEXT to retrieve a short explanation for an error code. In general, this situation must be treated as failed scan.
- **An exception is thrown:** This indicates a configuration error or severe problem during virus scanning. It must always be reported, and the virus scan is to be considered as failed.

If the parameter **IF_DO_CLEAN** is set with the value ABAP_TRUE, a cleanup should be performed. In the cleanup is successful, the parameter **EF_DATA** returns EF_SCANRC = CON_SCANRC_CLEAN_OK.

If the parameter **IF_DO_CLEAN** has the value ABAP_FALSE, it should only be checked.

In addition to the above method SCAN_BYTES, there are two other methods for virus scanning available in the class CL_VSI:

- Method SCAN_FILE for scanning a local file on the application server.
- Method SCAN_ITAB for scanning an internal table with row type X or C.

For more information about DDIC objects, tables, views, search helps, messages, function modules, class library reports and transactions of the Virus Scan Interface, see the online documentation in the ABAP Workbench.

For information on integrating the Virus Scan Interface into customer developments, see Integrating the Virus Scan Interface into Customer Developments.

You can see a commented source code for a VSI application for scanning files that are uploaded from a workstation in Commented Example Program. The report RSVSCANTEST contained in the system performs this task in an appropriate way and can also be used as a demonstration object.

Be Aware of the Following Problems

If an input channel does not implement the Virus Scan Interface itself, then a successive application may implement its own VSI.

However, you should avoid a document being scanned multiple times during its way into the system if the component delivering the data has already performed a virus scan.

Security Audit Log Triggered by VSI

Class CL_VSI automatically creates entries in the Security Audit Log for found infections and scan errors, together with the following information:

- About the profile
- The profile step, which allows the detection of the scanner-group
- The kind of virus found (if available with internal virus ID of the scan engine)
- The user name and timestamp

The logged messages are located in message class VSCAN using the System Log messages BU8 and BU9 (created in SE92). The severities are set to “High” and “Medium” respectively, for the Audit class it is set to “Miscellaneous”.

Further Information

- SAP Tutorials:
  - Configuration of the RFC destination
  - Configuration of the Virus Scan Server
  - Virus Scan Trace

  These tutorials are available on the SAP Service Marketplace at **service.sap.com/rkt**. See the Index: Find Learning Maps section and select Security for NetWeaver 04. The tutorials are available for both Technology Consultants and Development Consultants.
- SAP Note 786179: Data security products: Application in the antivirus area
- SAP Note 817623: Integrating a virus scan in SAP applications
- SAP NetWeaver Virus Scan Interface (NW-VSI) Specification

  www.sdn.sap.com/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/library/icc/NW-VSI%20Interface%20Documentation.pdf
  This document is available on the SAP Developer Network at **www.sdn.sap.com/irj/sdn/services** under SAP Integration and Certification Center (ICC) → Integration Scenarios (alphabetical) → NW-VSI.