

Requirements for SapXForm

Contents

- [SAP System](#)
- [HVR License for SapXForm](#)
- [Requirements for SAP Transform Engine](#)
 - [SAP Transform Engine on Windows](#)
 - [SAP Transform Engine on Linux](#)
 - [Verifying Mono Installation](#)
- [Channel Configuration](#)

HVR's SapXForm feature allows capturing changes from SAP "cluster" and "pool" tables and replicate into target database as "unpacked" data. For example, HVR can capture the SAP cluster table (**RFBLG**) from an Oracle based SAP system and unpack the contents (**BSEG**, **BSEC**) of the cluster table into a Redshift database; the HVR pipeline does the 'unpacking' dynamically. An essential component of the SapXForm feature is the SapXForm engine executable.

For using SapXForm with HVR 4.7.x, see [Configuring SapXForm with HVR 4.7](#).

SAP System

SAP database typically contains tables that fall into one of the following three categories:

- Transparent tables - ordinary database tables which can be replicated in a usual way.
- Pooled and Cluster tables - are special in that the data for several Pooled or Cluster tables are grouped and physically stored together in a single database table. HVR uses "SAP Transform Engine" to extract and replicate individual tables from SAP table pools and clusters.
- SAP Catalogs - contain metadata and do not usually need to be replicated. HVR and SAP Transform Engine themselves, however, need data from SAP Catalogs for the purpose of Pooled and Cluster tables processing.
HVR SapXForm supports capturing changes from SAP system with either of the databases [Oracle](#), [DB2i](#), [HANA](#) and [SQL Server](#).

To enable replication from SAP database using SapXForm, ensure that the SAP Dictionary tables (**DD02L**, **DD03L**, **DD16S**, **DDNTF**, **DDNTT**) exist in the source database. HVR uses the information available in SAP dictionary for unpacking data from SAP pool and cluster tables.

HVR unpacks all pool tables available in pack tables **ATAB** or **KAPOL**.

HVR also unpacks tables identified as cluster in the SAP dictionary. Examples of such tables are **BSEG** and **BSEC** which are packed inside **RFBLG**.

There are tables that SAP does not identify in its dictionary as "cluster tables" even though the tables contain clustered data. These are not supported. Examples include **PCL1**, **PCL2**, **MDTC**, and **STXL**.

HVR License for SapXForm

The regular HVR license file does not enable the SapXForm feature, so an additional license is needed. Contact HVR Support to obtain the proper SapXForm license. Place the license file (e.g. **hvr.lic** and **hvrspa.lic**) in **hvr_home/lib**.

HVR supports "accumulating" license files; this mean a hub machine could have several license files. For example, one license file (**hvr.lic**) enables all features (except SapXForm) of HVR to be used perpetually and another license file (**hvrspa.lic**) enables SapXForm feature.

If a valid license to use SapXForm is not available in the **hvr_home/lib** directory then the option to query the **SAP Dictionaries** in [Table Explore](#) will not be displayed in the HVR GUI.

Requirements for SAP Transform Engine

SAP Transform Engine is special executable which is shipped as part of the HVR distribution, and is licensed separately. The SAP Transform Engine can be run on Windows and Linux platforms; it can be configured to execute either on the HVR hub machine or the integrate machines.

SAP Transform Engine on Windows

In Windows, the SAP Transform Engine requires .NET Framework version 4.5 or higher to be installed in the system.

To verify the version of .NET Framework installed in the system, see [Microsoft Documentation](#).

SAP Transform Engine on Linux

In Linux, the SAP Transform Engine requires [Mono](#) version 4.4 or higher to be installed in the system. Mono is an open source implementation of Microsoft's .NET Framework.

- For newer Linux versions (EPEL 6, EPEL 7, Debian 7, Debian 8), Mono can be downloaded and installed from [Mono Downloads](#).
- For older Linux version (EPEL 5) or to use as an alternative method for EPEL 6 and EPEL 7, the installation for Mono is not available in the [Mono Downloads](#). However, the Mono installation files for EPEL 5 can be obtained from a community managed repository - <https://copr.fedorainfracloud.org/coprs/tpokorra/mono>. This repository contains Mono-community built packages which install into **/opt** for CentOS (5,6,7). These are automatically compiled from original sources.
 1. Click **Packages** tab.
 2. Click **mono-opt** package ("mono installed in opt").
 3. Click on the latest successful **Mono 4.6.x** build (for example **4.6.2-1**).
 4. In the **Results** pane, click **epel-5-x86_64**.
 5. Click **mono-opt-4.6.x.x86_64.rpm** to download the RPM file.

Verifying Mono Installation

The Mono installation directory contains a script file **env.sh**. This script is intended to be included before running Mono, and is used to configure host specific environment. To verify if the installed Mono is working, perform the following steps (assuming Mono is installed in **/opt/mono**) :

1. Start a new clean shell.
2. Execute the following command to include the mono environment setup file into current shell:

```
source /opt/mono/env.sh
```

3. Execute the following command to view Mono:

```
mono --version
```

4. If any error such as library error is displayed, it indicates incorrect installation/configuration of Mono. Verify the parameters defined for configuring the host specific environment in **/opt/mono/env.sh** and rectify them, if required.
5. Repeat this verification procedure in a new clean shell.

Channel Configuration

SapXForm requires [Integrate /Burst](#) to be defined in the channel.