# VMware Storage Policy SDK (SPBM) Release Notes

Release date: 17 APR 2018 | ESXi build 8169922, SDK build 7970399
For vSphere 6.7 GA. Last document update: 19 SEP 2018
Check frequently for additions and updates to these release notes.

## Overview

VMware vSphere includes a Storage Policy Server that provides support for virtual machine storage profiles in the vSphere environment. Administrator can define storage policies, or use existing ones, to help automate storage provisioning for individual virtual machines. The Storage Policy SDK provides an API that your client application can use to create and manipulate storage policies that define storage requirements for virtual machine files.

This approach to storage provisioning is called Storage Policy Based Management (SPBM). SPBM is most useful when used in combination with Virtual Volumes (VVols).

## Distribution Kit

The VMware Storage Policy SDK is distributed as part of the vSphere Management SDK, which is is a collection of vSphere SDKs based on web services. When you extract the contents of the distribution kit, the VMware Storage Policy SDK appears in the `SDK/spbm` sub-directory:

```
VMware-vSphere-SDK-6.7.0-buildNumber.zip
    SDK
        eam
        sms-sdk
        spbm
        ssoclient
        vsphere-ws
```

## What's New in This Release

VMware Storage Policy SDK 6.7 introduces the following new features:

- Storage policy support for first class disk (FCD).
  First class disk is virtual disk (VMDK) not associated with a virtual machine. In this release, SPBM can direct FCD to a datastore with specific characteristics. FCD is identified by a `ServerObjectRef` and has its FCD ID set, so it can be distinguished if and when associated with a virtual machine. FCD is capable of most normal disk operations, including replication.

- Persistent memory support (PMem).
  ESXi 6.7 supports non-volatile memory (NVM) for byte-addressable memory. During virtual machine provisioning, administrators set a storage policy to select Pmem as the storage type. Pmem can be consumed two ways: as memory attached to a virtual machine, or as a virtual disk policy existing within the storage subsystem. Programs use `VirtualMachineProfileSpec` instead of a datastore in the backing object to specify PMem for virtual disk storage, If a storage policy identifies VMs in a cluster with NVM attached, vMotion always migrates NVM-backed virtual disks along with their VMs.

- SPBM support added to the vSphere Automation SDK.
  Storage policy calls have been added to the vSphere Automation API, also called vAPI. Look for PBM related function in the various language bindings.

- Better VASA provider for vSAN.
  For better performance and reliability, the SOAP based VASA provider for vSAN was replaced with a vSphere API (VMODL) based VASA provider for vSAN.

- Phone home integration for SPBM.
  Telemetry information related to storage policies was added as part of the Customer Experience Improvement Program (CEIP) in this release.

- Enhancements to support VMware Cloud.
  Additional storage profiles were created for VMware Cloud use, and privilege checking was tightened for the storage policy APIs.

VMware Storage Policy SDK 6.5 introduced the following new features:

- Lines of Services definition for SPBM policies.
  vSphere 6.5 introduces the Lines of Services concept as a new way to define SPBM policies. In this scheme, VASA providers explicitly advertise the different types of services they can provide, such as Persistence (storage), Inspection, Replication, Encryption, Cacheing, and so on. This allows administrators to create sophisticated policies that combine different service types to meet diverse application requirements.

- Support for Replication line-of-service policy for provisioning and monitoring.
  vSphere 6.5 supports provisioning of virtual machine storage based on replication line-of-service policy as supported by Virtual Volume (VVol) enabled storage arrays.

## Documentation

For information about how to use the VMware Storage Policy API, see the *VMware Storage Policy API Programming Guide* and the *VMware Storage Policy API Reference*. You can find these documents on the VMware {Code} website in the vSphere Management SDK section.

# Resolved Issues

The 6.7 Storage Policy SDK and vSphere 6.7 resolved the following issue:

- **Compliance check runs out of memory if passed lots of virtual machines.**
  When checking compliance with `fetchCompliance`, SPBM hits a 120 second time-out, and the Policy Summary page of the vSphere Web Client appears to hang. This is caused by an out-of-memory error when serving too many VM compliance requests, especially with IVD (improved virtual disk, formerly FCD) for example more than 10,000. The `PbmFetchComplianceResult` routine throws "java.lang.OutOfMemoryError: Java heap space" into the SPS runtime log. The workaround is to use the vSphere Client (HTML-5) where the problem was fixed by extending the API with a pagination feature to accept smaller batches in succession. See KB 56930.

# Known Issues

The 6.5 Storage Policy SDK was documented with the following issues, which still exist:

- **Compatibility issues with I/O scripts running against vCenter Server 6.5.**
  If you run programs or scripts that create or modify storage policies containing I/O filter rules that use the `IOFILTERS` namespace against vCenter Server 6.5, those programs or scripts fail. I/O filter capabilities are now exposed under the I/O filter's name. Scripts that directly call Storage Policy-Based Management APIs should be updated to identify the I/O filter capabilities correctly.

  **Workaround:** Modify the scripts to invoke the `PbmFetchCapabilitySchema` API under `PbmProfileManager` in order to get the schemas (`PbmCapabilityObjectSchema`). In the returned schemas, if `PbmCapabilityObjectSchema.lineOfService` is an instance of `PbmVaioDataServiceInfo`, then that schema represents the I/O filter capabilities exposed by an I/O filter. There will be one schema exposed per I/O filter. The namespace of this schema will be the same as the I/O filter name.
  Note: While authoring the storage policy, the I/O filter capabilities in the policy should be as per the I/O filter schema exposed by Storage Policy-Based Management.

- **PbmFetchCapabilityMetadata does not validate resourceType.**
  `PbmFetchCapabilityMetadata` does not validate the `resourceType` parameter and returns the result for `STORAGE - resourceType` instead.

- **Internal server error if tagging description field is empty.**
  The Tag Category validation currently ignored during PBM `queryOvf`. Therefore, if the OVF descriptor has an invalid or non existing Tag Category, no errors are added to `OvfParamsResult` during `queryOvf` when `TagBasedPlacementGroupSection` has an invalid `TagCategory`.

  When a Storage Policy API user uses OVF `tryInstantiate` API to validate the OVF, no error will be thrown when the Tag Category specified in the OVF is invalid as long as the tag specified in the OVF is valid and is available to the VDC clusters. However when the user actually tries the import such an OVF the actual import will fail with `NotFound` error.

  Note: When you actually exports to an OVF, both the tag and tag category ID are exported. In most cases, when tryInstantiate is called, either both the values will be valid or both invalid and so an error should be thrown if invalid.

  This is related to an issue with the vSphere Automation SDK that happens because the Tag Category Description field is optional in the vSphere Web Services API, but it is mandatory in the vSphere Automation SDK (this is the new SDK for vCenter services). Because the field is mandatory for vCenter services, if the description field is empty, the vSphere Automation SDK can also return an internal server error that states, "the 'description' field of the Category model is empty".

  **Workaround:** When creating tags, do not leave the description field empty. If you do not have any description, you may add a whitespace character (i.e. space).

- **PbmFetchVendorInfo method parameter resourceType is required.**
  The VMware Storage Policy API Reference shows the `resourceType` parameter as optional. Although it is an optional parameter, currently you must specify this parameter.

- **PbmCapabilityProfile property lastUpdatedBy is required.**
  There is currently a mismatch between the VASA API and Storage Policy API, where the lastUpdateBy field is optional in the VASA API and required in the Storage Policy API. Therefore, it should be treated as mandatory in both APIs.

- **Virtual machine is created despite an invalid storage policy.**
  Using the vSphere API, you can create a virtual machine with a storage policy that is not in compliance with the associated storage capability. This occurs even if the subprofile constraint property `forceProvision` is set to false. If you use the vSphere Web Client to create the virtual machine, the create operation fails because the policy is not compliant.

- **Do not use the VirtualMachine.Relocate method to change the storage profile of a VM.**
  The `VirtualMachine.Relocate` method allows you to specify a new storage profile for a disk or virtual machine without specifying either a new host or datastore. The vCenter Server does not recognize this input as valid and does not change the storage profile. Use the `VirtualMachine.Reconfigure` function to assign a new storage profile to the virtual disk and/or virtual machine.

- **When cloning a virtual machine, use a VirtualMachineRelocateSpec to specify a storage profile.**
  The `VirtualMachineCloneSpec` object provides two ways to specify a storage profile.

  1. The `location` property is a `VirtualMachineRelocateSpec` object. Use the relocate spec property `profile` to specify a storage profile.

2. The `config` property is a `VirtualMachineConfigSpec` object. You can use the config spec property `vmProfile` to specify a storage profile, but this works only if the source virtual machine is not associated with a storage profile.