Site Recovery Manager
API Developer's Guide

Site Recovery Manager 8.4
You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/
Contents

1 About Site Recovery Manager API Developer's Guide 11

2 APIs for VMware Site Recovery Manager 12
   API Releases 12
   Site Recovery Manager Appliance Management API 14
      List of API Operations 15
      Managed Object Hierarchy 18
   Site Recovery Manager API 21
      List of API Operations 21
      Managed Object Hierarchy 30
      Logging into Sites with SAML Tokens 36
   WSDL Programming Environments 38
   Accessing Site Recovery Manager APIs 39

3 SDK Installation and Setup 41
   Contents of the SDK Package 41
   SDK Directory Structure 42
   Download and Setup 43

4 SDK Samples for Site Recovery Manager Appliance Management API 45
   About C# .NET Samples 45
      Build Sample Code with Visual Studio 2017 45
      Run Sample Code from Visual Studio 2017 45
   About Java JAX-WS Samples 46
      Build JAX-WS Sample Code 46
      Run JAX-WS Sample Code 47
      Clean up JAX-WS Sample Code 47
   About Java Axis Samples 48
      Build JAVA AXIS Sample Code 48
      Run Java Axis Sample Code 48
      Clean up JAVA AXIS Sample Code 49

5 SDK Samples for Site Recovery Manager API 50
   About the C# .NET Samples 50
      Build Sample Code with Visual Studio 2008 50
      Build Sample Code with Visual C# 2008 Express 51
      Run Sample Code from Visual Studio 51
      Run C# Sample Code 52
6 Logical Usage Order - Site Recovery Manager Appliance Management API 56

Appliance Manager 56
  GetAllTimeZones 56
  GetCurrentDateTime 57
  GetCurrentTimeZone 57
  GetDiskInfo 57
  GetInfo 58
  GetNetworkInfo 58
  GetTimeSyncConfig 61
  Restart 61
  SetCurrentTimeZone 61
  SetNetworkInfo 62
  SetTimeSync 62
  Stop 63

Configuration Manager 63
  GetRunningTask 63
  CheckRegistration 64
  ClearSrmConfiguration 64
  ConfigureSrm 66
  ConfigureSyslogForwarding 67
  ConfigureSyslogServers 68
  EnableSyslogLogging 68
  GetSyslogServers 69
  IsReconfigureRequired 69
  ListVcServices 69
  ReadCurrentConfig 70
  SendSyslogTestMessage 71
  ValidateConnection 71
  SetHbrSrvNic 72
  GetHbrSrvNic 72

Configuration Task 73
  GetTaskInfo 73
CancelSrmConfiguration  74
Database Manager  74
  ReadStatus  74
  ChangePassword  75
Diagnostic Manager  76
  GetRunningTask  76
  GenerateSystemLogBundle  76
  RetrieveSystemLogBundle  77
  DeleteSystemLogBundle  78
Service Instance  78
  RetrieveContent  79
  LoginDrConfig  80
  LogoutDrConfig  81
  ChangeUserPassword  81
Service Manager  81
  IsSrmServerRunning  82
  DrConfigStartService  82
  DrConfigStopService  83
  DrConfigServiceStatus  83
  DrConfigRestartService  84
  DrConfigAllServicesStatus  84
SRA Manager  84
  GetRunningTask  85
  GetSraImages  85
  DeleteImage  86
  DeleteImageContainers  87
  GetImageInfo  87
  CopySraConfiguration  88
  ResetToFactorySettings  89
SSL Certificate Manager  89
  ProbeSsl  90
  DrConfigGenerateCSR  90
  DrConfigSetCertificate  91
  DrConfigSetKeyCertificate  92
  AddCaCertificates  93
  RemoveCaCertificates  94
  RetrieveCaCertificates  94
  ClearCaCertificates  94
  InstallSelfSignedCertificate  95
  InstallCertificate  95
  GetCertificateInfo  96
Update Manager 96
  GetRunningTask 96
  UpdateRepository 97
  DrConfigCheckForUpdates 97
  InstallUpdate 98
  GetRepositories 99

7 Logical Usage Order - Site Recovery Manager API 100

Service Instance 101
  GetSiteName 101
  GetPairedSite 102
  RetrieveContent 103
  GetLocalSiteInfo 103
  Solution User Information 104
  SAML Token Authentication 105
  Credential Based Authentication 108
  GetLicenseInfo 111
  ProbeSsl 112
  PairSrm 113
  BreakPairing 114
  ReconfigureConnection 115
SrmExtApiTask 116
  IsSrmExtApiTaskComplete 116
  GetSrmExtApiTaskInfo 116

SRM Folder 116
  GetName 116
  GetParentFolder 117
  GetChildType 117
  CreateFolder 117
  MoveFolder 118
  DestroyFolder 119
  RenameFolder 120

Inventory Mappings 120
  AddFolderMapping 120
  RemoveFolderMapping 121
  AddNetworkMapping 121
  RemoveNetworkMapping 122
  AddResourcePoolMapping 122
  RemoveResourcePoolMapping 123
  AddTestNetworkMapping 123
  RemoveTestNetworkMapping 124
GetFolderMappings 124
GetNetworkMappings 125
GetResourcePoolMappings 125
GetTestNetworkMappings 126
AutoProtect Manager 126
SetAutoProtectUser 126
GetAutoProtectUser 127
SetDefaultAutoProtectUser 127
IsActive 127
Protection 128
ListProtectionGroups 128
ListInventoryMappings 129
ListReplicatedDatastores 130
GetProtectionGroupRootFolder 130
ListUnassignedReplicatedDatastores 131
ProtectionListReplicatedDatastores 131
ListUnassignedReplicatedVms 131
ProtectionListProtectedVms 132
CreateAbrProtectionGroup 132
CreateHbrProtectionGroup 133
CreateHbrProtectionGroup2 135
CreateVvolProtectionGroup 135
RemoveProtectionGroup 136
Protection Group Folder 137
ListChildProtectionGroupFolders 137
ListChildProtectionGroups 137
GetProtectionGroup 137
Create Protection Group Task 138
IsCreateProtectionGroupComplete 138
GetCreateProtectionGroupResult 138
GetNewProtectionGroup 139
Protection Group 139
GetInfo 139
ProtectionGroupGetParentFolder 140
GetPeer 140
ListProtectedVms 141
ListProtectedDatastores 142
ListAssociatedVms 143
GetProtectionState 144
ProtectionGroupListRecoveryPlans 144
ProtectionGroupQueryVmProtection 145
ProtectVms 146
UnprotectVms 149
AssociateVms 150
UnassociateVms 151
CheckConfigured 152
ProtectionGroupGetOperationalLocation 152
AddDatastores 152
RemoveDatastores 153
ReconfigureVvolProtectionGroup 153
GetVvolGroupDetails 154
MoveGroup 155
GetPlaceholderVmInfo 156
RecreatePlaceholder 157
GetRecoveryLocationSettings 158
ReconfigureRecoveryLocationSettings 158
Protection Task 159
GetProtectionStatus 159
GetTasks 160
IsComplete 161
GetResult 161
Recovery 162
ListPlans 162
GetHistory 162
GetRecoveryPlanRootFolder 163
CreateRecoveryPlan 163
DeleteRecoveryPlan 165
MovePlan 165
Recovery Plan Folder 166
ListChildRecoveryPlanFolders 166
ListChildRecoveryPlans 167
GetRecoveryPlan 167
Recovery Plan 168
RecoveryPlanGetInfo 168
RecoveryPlanGetPeer 170
Start 170
Cancel 172
ListPrompts 172
AnswerPrompt 173
RecoveryPlanGetParentFolder 174
GetRecoverySettings 174
SetRecoverySettings 177
AddProtectionGroup 178
AddTestNetworkMappingToRecoveryPlan 178
RemoveTestNetworkMappingFromRecoveryPlan 179
RemoveProtectionGroupFromRecoveryPlan 180
RecoveryPlanGetLocation 180
RecoveryPlanHasRunningTask 181
Recovery History 181
GetRecoveryResult 181
GetResultCount 183
GetResultLength 184
RetrieveStatus 184
IP Subnet Mapper 185
GetIpSubnetMappings 185
AddIpMapping 187
RemoveIpMappings 188
Storage Adapter 189
FetchInfo 189
GetAdapterConnectionSpec 189
Storage 191
DiscoverDevices 191
QueryArrayManagers 192
CreateArrayManager 192
QueryStorageAdapters 193
RemoveArrayManager 194
ReloadAdapters 194
ArrayManager 194
ReadInfo 195
QueryReplicatedArrayPairs 195
GetArrayInfo 195
GetAdapter 196
AddArrayPair 196
RemoveArrayPair 197
DiscoverArrays 198
Reconfigure 198
GetArrayDiscoveryStatus 199
ReplicatedArrayPair 200
QueryReplicatedRdms 200
GetDevices 201
GetDeviceGroups 202
GetReplicatedDatastores 203
GetDeviceDiscoveryStatus 203
8 Deprecated APIs  210

9 Site Recovery Manager Faults  211
   Faults in Site Recovery Manager Appliance Management API  211
   Faults in Site Recovery Manager API  214

10 SSL Certificates and SNMP Traps  223
   SSL Certificates  223
      Get vCenter Server Certificate  223
      Export Cached Certificates to a Local Directory  224
      About the Virtual Machine Keystore  225
   SNMP Traps  226
      MIB Names for SNMP Traps  226
      Configuring SNMP Receivers in vCenter Server  227
      SNMP Traps and Object IDs  227
About Site Recovery Manager API Developer’s Guide

The Site Recovery Manager API Developer’s Guide provides information about programming applications with the Web services interfaces to VMware Site Recovery Manager.

This manual provides information about interfaces in the Site Recovery Manager 5.0, 5.8, 6.0, 6.1, 6.5, 8.1, 8.2, and 8.3 releases, for developers who are interested in automating configuration of Site Recovery Manager Virtual Appliance or Site Recovery Manager tasks.

Intended Audience

This book is intended for developers who want to set up their environment to program applications with the Site Recovery Manager API. These developers are typically programmers using the Java or C# language and libraries to perform configuration of Site Recovery Manager Virtual Appliance or perform replication, recovery, and reprotect of virtual machines in VMware vSphere.

Site Recovery Manager developers should have some familiarity with the Web Services Description Language (WSDL) and the Simple Object Access Protocol (SOAP) for transmitting XML across the network. However, the important interfaces are completely visible in Java or C# code.

VMware Developer Publications

To view the current version of this book and other VMware API and SDK public documentation, go to http://www.vmware.com/support/developer.

Visit https://www.vmware.com/support/pubs/srm_pubs.html for information about this version of Site Recovery Manager.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.
APIs for VMware Site Recovery Manager

This manual describes Web services programming interfaces to programmatically configure the Site Recovery Manager Virtual Appliance, and create protection groups and recovery plans in VMware Site Recovery Manager.

The Site Recovery Manager Virtual Appliance Management API and the Site Recovery Manager API provides an interface similar to vSphere API, an object-oriented Web service that provides access to vSphere and virtual machine management on vCenter Server and ESXi hosts.

You can program the vSphere API in Java, C#, or any language that supports the Web services definition language (WSDL).

The Site Recovery Manager Virtual Appliance Management API provides a way for configuring Site Recovery Manager Server or OS-specific settings. The Site Recovery Manager API allows the third-party systems to create protection groups and initiate test, recovery, and reprotect operations and collect the results.

This chapter includes the following topics:

- API Releases
- Site Recovery Manager Appliance Management API
- Site Recovery Manager API
- WSDL Programming Environments
- Accessing Site Recovery Manager APIs

API Releases

The Site Recovery Manager 5.0 release extended the API with new methods to list and modify protection groups, and revised methods to list, modify, and run recovery plans.

The Site Recovery Manager 5.8 release introduced 30 methods and 4 new managed objects, adding several requested features to the API:

- Ability to add folder, network, and resource pool mappings
- Support for planned migrations
Navigation capabilities for protection group folders and recovery plan folders

Ability to create protection groups, and to modify selected fields in a virtual machine’s recovery settings

The Site Recovery Manager 6.0 release introduced new methods to support authentication by tokens:

- New concept of the Site Recovery Manager solution user for vCenter Single Sign-On authentication
- Functions to get the Site Recovery Manager solution user name for the local and remote sites
- Functions to log in to local or remote sites using security assertion markup language (SAML) token
- Ability to use lookup service URL and vCenter Server instance ID
- Although DisasterRecoveryApi is deprecated, it gains forward compatibility with LoginByToken

In release 6.1, VMware Site Recovery Manager introduced Storage Profile Protection Groups (SPPG). However, the Site Recovery Manager API does not support SPPG related objects.

Site Recovery Manager API 6.5 introduced methods that expand the operations on the inventory mappings, recovery plans, and protection groups.

The Site Recovery Manager 8.1 release introduces new methods that provide ability to:

- Add replicated datastores (that are newly provisioned) to an existing protection group.
- Remove datastores from the protection group.
- Configure the IP address and corresponding DNS, WINS of the virtual machine, after the migration is complete.

The Site Recovery Manager 8.2 release introduces new methods that provide ability to:

- Read information specific to an ArrayManager instance and list all the array pairs in the array manager.
- Get information about all the replicated RDMs in a replicated array pair.
- Get a list all the available array managers.

The Site Recovery Manager 8.3 release introduces the following:

- Site Recovery Manager Virtual Appliance Management APIs for configuring Site Recovery Manager Virtual Appliance or OS-specific settings.
- APIs for vVOL management.
- APIs for automatic protection.
- APIs for managing IP Subnet Mapping between protection and recovery site networks, defining the rules used for translating VM’s IP settings between protection and recovery sites, and customizing the IP information for a specific network adapter.
The Site Recovery Manager 8.4 release introduces the following:

- Site Recovery Manager APIs for pairing, manual per VM protection, array manager management, and folders operations.
- Site Recovery Manager Virtual Appliance Management APIs can be used for configuring VRMS vSphere replication management server (VRMS-HMS) and VRS vSphere replication server (VRS-HBR).

**Terminology**

This document uses the following terms.

**SOAP**

Client applications invoke operations by sending SOAP formatted messages. When passing data objects between client and server, programs build or parse XML messages representing data structures described by the WSDL. Standardized by W3C as Simple Object Access Protocol (SOAP) 1.1.

**Web service operations**

Client interfaces that perform server-side management and monitoring tasks. Standardized as Web Services Interoperability Organization (WS-I) Basic Profile 1.0.

**WSDL**

The Web services API is defined in a WSDL file, which is used by client-side Web services to create proxy code (stubs) that client applications use to interact with the server. Standardized as Web Services Description Language (WSDL) 1.1.

**XML**

A text representation scheme similar to HTML but with more stringent, regularized syntax. Standardized by W3C as Extensible Markup Language (XML) 1.0.

The Site Recovery Manager Appliance Management API and the Site Recovery Manager API are similar to and derived from the vSphere API. For information about the vSphere API, see the [vSphere Web Services SDK Programming Guide](https://www.vmware.com/support/pubs/vsphere_web_services_sdk.html) and the [vSphere API Reference](https://pubs.vmware.com/vsphere/65/topic/com.vmware.vsphere.client.doc_6.5/vsphere_api_reference/index.html) at the VMware website.

**Site Recovery Manager Appliance Management API**

The Site Recovery Manager Appliance Management API provides language-neutral interfaces to the Site Recovery Manager Virtual Appliance management framework for configuring Site Recovery Manager Server or OS specific settings.

The API is implemented as industry-standard Web service, running on Site Recovery Manager Virtual Appliance. The Site Recovery Manager Appliance Management API complies with the Web Services Interoperability Organization (WS-I) Basic Profile 1.0, which includes XML Schema 1.0, SOAP 1.1, WSDL 1.1. For more information about the WS-I Basic Profile 1.0, see:
List of API Operations

The following tables provide a list of Site Recovery Manager Appliance Management API methods arranged in alphabetical order:

**Table 2-1. Appliance Manager**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAllTimeZones</td>
<td>Gets all available time zones. It returns list representing all available time zones.</td>
</tr>
<tr>
<td>GetCurrentDateTime</td>
<td>Gets current date and time of the appliance. It returns a vmodl.DateTime object with the appliance date and time.</td>
</tr>
<tr>
<td>GetCurrentTimeZone</td>
<td>Gets current time zone of the appliance. It returns a string representing the current time zone.</td>
</tr>
<tr>
<td>GetDiskInfo</td>
<td>Retrieves appliance disks information. It returns a DiskInfo object which contains disk information about the appliance.</td>
</tr>
<tr>
<td>GetInfo</td>
<td>Retrieves appliance information. It returns an ApplianceInfo object which contains information about the appliance.</td>
</tr>
<tr>
<td>GetNetworkInfo</td>
<td>Retrieves appliance network information. It returns a NetworkInfo object which contains network information about the appliance.</td>
</tr>
<tr>
<td>GetTimeSyncConfig</td>
<td>Gets appliance time sync mode. It returns a TimeSyncInfo object representing the timeSyncMode.</td>
</tr>
<tr>
<td>Restart</td>
<td>Restarts the appliance.</td>
</tr>
<tr>
<td>SetCurrentTimeZone</td>
<td>Sets appliance time zone.</td>
</tr>
<tr>
<td>SetNetworkInfo</td>
<td>Sets appliance network information.</td>
</tr>
<tr>
<td>SetTimeSync</td>
<td>Sets appliance time sync information.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops the appliance.</td>
</tr>
</tbody>
</table>

**Table 2-2. Configuration Manager**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckRegistration</td>
<td>Checks whether the given extension key is already registered in SSO, lookup service, and as vCenter extension. Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>ClearSrmConfiguration</td>
<td>Clears the SRM server configuration with the vSphere infrastructure. Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>ConfigureSrm</td>
<td>Configures the SRM server and connects it to the vSphere infrastructure. Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>ConfigureSyslogForwarding</td>
<td>Sets syslog log forwarding. When enable is set to true, this method will add a rule to rsyslog configuration for given apps and restart rsyslog service. Applicable to SRM, VRMS, and VRS.</td>
</tr>
</tbody>
</table>
### Table 2-2. Configuration Manager (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigureSyslogServers</td>
<td>Sets the syslog log forwarding. Applicable to SRM, VRMS, and VRS.</td>
</tr>
<tr>
<td>EnableSyslogLogging</td>
<td>Enables or disables logging to syslog. Applicable to SRM.</td>
</tr>
<tr>
<td>GetHbrSrvNic</td>
<td>Gets the HBR filter and management IP addresses. Applicable to VRMS, and VRS.</td>
</tr>
<tr>
<td>GetRunningTask</td>
<td>Gets the currently active configuration task or null. Applicable to SRM, VRMS, and VRS.</td>
</tr>
<tr>
<td>GetSyslogServers</td>
<td>Gets the syslog log forwarding information. Applicable to SRM, VRMS, and VRS.</td>
</tr>
<tr>
<td>IsReconfigureRequired</td>
<td>Checks if the reconfigure operation is required after an upgrade. Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>ListVcServices</td>
<td>Lists all the vCenters in the Platform Service Controller (PSC). Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>ReadCurrentConfig</td>
<td>Reads the specification for the currently configured SRM server. Applicable to SRM and VRMS.</td>
</tr>
<tr>
<td>SendSyslogTestMessage</td>
<td>Sends test message to all configured syslog servers. Applicable to SRM, VRMS, and VRS.</td>
</tr>
<tr>
<td>SetHbrSrvNic</td>
<td>Sets the HBR filter and management addresses. Applicable to VRMS and VRS.</td>
</tr>
<tr>
<td>ValidateConnection</td>
<td>Validates connections to the vSphere infrastructure. Applicable to SRM and VRMS.</td>
</tr>
</tbody>
</table>

### Table 2-3. Configuration Task

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CancelSrmConfiguration</td>
<td>Cancels a running configuration task. Multiple cancel requests are treated as a single cancellation request.</td>
</tr>
<tr>
<td>GetTaskInfo</td>
<td>Gets the current configuration task status.</td>
</tr>
</tbody>
</table>

### Table 2-4. Database Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangePassword</td>
<td>Changes the embedded database password.</td>
</tr>
<tr>
<td>ReadStatus</td>
<td>Checks the database status and return the version information.</td>
</tr>
</tbody>
</table>

### Table 2-5. Diagnostic Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRunningTask</td>
<td>Gets the currently active retrieve update task or null.</td>
</tr>
<tr>
<td>GenerateSystemLogBundle</td>
<td>Instructs the server to generate a system log bundle.</td>
</tr>
</tbody>
</table>
Table 2-5. Diagnostic Manager (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetrieveSystemLogBundle</td>
<td>Retrieves the log bundle using the Binary datatype.</td>
</tr>
<tr>
<td>DeleteSystemLogBundle</td>
<td>Instructs the server that this log bundle is no longer needed by the client that generated it.</td>
</tr>
</tbody>
</table>

Table 2-6. Service Instance

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangeUserPassword</td>
<td>Assigns password to the user, who is running the drconfig service.</td>
</tr>
<tr>
<td>LoginDrConfig</td>
<td>Logs on to the server by verifying user and password with the local OS.</td>
</tr>
<tr>
<td>LogoutDrConfig</td>
<td>Log out and terminate the current session.</td>
</tr>
<tr>
<td>RetrieveContent</td>
<td>Retrieves the properties of the service instance.</td>
</tr>
</tbody>
</table>

Table 2-7. Service Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrConfigStartService</td>
<td>Starts the service.</td>
</tr>
<tr>
<td>DrConfigStopService</td>
<td>Stops the service.</td>
</tr>
<tr>
<td>DrConfigServiceStatus</td>
<td>Returns a ServiceStatus object which contains the service status information about the service.</td>
</tr>
<tr>
<td>DrConfigRestartService</td>
<td>Stops the service and then restarts it.</td>
</tr>
<tr>
<td>DrConfigAllServicesStatus</td>
<td>Returns a ServiceStatus object for all the services.</td>
</tr>
<tr>
<td>IsSrmServerRunning</td>
<td>Returns the current service state of the Site Recovery Manager</td>
</tr>
</tbody>
</table>

Table 2-8. SRA Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopySraConfiguration</td>
<td>Copies the SRA configuration from a source image to a destination image.</td>
</tr>
<tr>
<td>DeleteImage</td>
<td>Stops and then deletes the containers instantiated from the given image, and deletes the image itself.</td>
</tr>
<tr>
<td>DeleteImageContainers</td>
<td>Stops and then deletes the containers which were instantiated from the given image.</td>
</tr>
<tr>
<td>GetRunningTask</td>
<td>Gets the currently active retrieve update task or null.</td>
</tr>
<tr>
<td>GetSraImages</td>
<td>Returns a collection of SRA images loaded into the docker daemon of the Site Recovery Manager Virtual Appliance.</td>
</tr>
<tr>
<td>GetImageInfo</td>
<td>Returns the image information as taken from the queryinfo SRA command.</td>
</tr>
<tr>
<td>ResetToFactorySettings</td>
<td>Reverts the SRA image’s configuration to its factory settings.</td>
</tr>
</tbody>
</table>
Table 2-9. SSL Certificate Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddCaCertificates</td>
<td>Adds certificate authority certificates to the list of validating certificates.</td>
</tr>
<tr>
<td>ClearCaCertificates</td>
<td>Completely clears Site Recovery Manager specific list of certificate authority certificates, used by the Site Recovery Manager to validate other server’s certificates.</td>
</tr>
<tr>
<td>DrConfigGenerateCSR</td>
<td>Generates a new key and CSR, and returns them for signing.</td>
</tr>
<tr>
<td>DrConfigSetCertificate</td>
<td>Sets a new certificate. Reconfigures the Site Recovery Manager if already configured. Restarts the proxy service.</td>
</tr>
<tr>
<td>DrConfigSetKeyCertificate</td>
<td>Sets a new key and certificate, reconfigures Site Recovery Manager if already configured, and then restarts the proxy service.</td>
</tr>
<tr>
<td>GetCertificateInfo</td>
<td>Lists the certificate info.</td>
</tr>
<tr>
<td>InstallSelfSignedCertificate</td>
<td>Installs self-signed certificate, reconfigures the Site Recovery Manager if already configured, and restarts the proxy service.</td>
</tr>
<tr>
<td>InstallCertificate</td>
<td>Installs the PKCS#12 certificate, reconfigures Site Recovery Manager if already configured, and restarts the proxy service.</td>
</tr>
<tr>
<td>RemoveCaCertificates</td>
<td>Removes certificate authority certificates from the list of validating certificates.</td>
</tr>
<tr>
<td>RetrieveCaCertificates</td>
<td>Gets the current SRM specific list of certificate authority certificates used by SRM to validate other server’s certificates.</td>
</tr>
<tr>
<td>ProbeSsl</td>
<td>Checks if the Site Recover Manager can establish successful SSL connection to the specified endpoint.</td>
</tr>
</tbody>
</table>

Table 2-10. Update Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrConfigCheckForUpdates</td>
<td>Checks for updates. It checks the repository for available updates.</td>
</tr>
<tr>
<td>GetRepositories</td>
<td>Gets update repos.</td>
</tr>
<tr>
<td>GetRunningTask</td>
<td>Tests the currently active retrieve update task or returns null.</td>
</tr>
<tr>
<td>InstallUpdate</td>
<td>Installs the update.</td>
</tr>
<tr>
<td>UpdateRepository</td>
<td>Changes the update repository location.</td>
</tr>
</tbody>
</table>

Managed Object Hierarchy

The following table shows the managed object hierarchy of the Site Recovery Manager Appliance Management API with the methods of each managed object in an alphabetical order.
<table>
<thead>
<tr>
<th>Managed Object</th>
<th>Remarks</th>
<th>Local Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplianceManager</td>
<td>Tools to manipulate the appliance of SRM server.</td>
<td>GetAllTimeZones, GetCurrentDateTime, GetCurrentTimeZone, GetDiskInfo, GetInfo, GetNetworkInfo, GetTimeSyncConfig, Restart, SetCurrentTimeZone, SetNetworkInfo, SetTimeSync, Stop</td>
</tr>
<tr>
<td>ConfigurationManager</td>
<td>Tools to configure SRM server</td>
<td>CheckRegistration, ClearSrmConfiguration, ConfigureSrm, ConfigureSyslogForwarding, ConfigureSyslogServers, EnableSyslogLogging, GetHbrSrvNic, GetRunningTask, GetSyslogServers, IsReconfigureRequired, ListVcServices, ReadCurrentConfig, SendSyslogTestMessage, SetHbrSrvNic, ValidateConnection</td>
</tr>
<tr>
<td>ConfigurationTask</td>
<td>Operations to configure SRM server</td>
<td>GetTaskInfo, CancelSrmConfiguration</td>
</tr>
<tr>
<td>DatabaseManager</td>
<td>Operations to configure SRM database.</td>
<td>ChangePassword, ReadStatus</td>
</tr>
<tr>
<td>DiagnosticManager</td>
<td>Describes the interface to get the Site Recovery Manager system log bundles that contains log files, cores and configuration files that are useful for diagnosis of issues.</td>
<td>DeleteSystemLogBundle, GetRunningTask, GenerateSystemLogBundle, RetrieveSystemLogBundle</td>
</tr>
<tr>
<td>ServiceInstance</td>
<td>Singleton object which provides access to the functionality of the DrConfig server.</td>
<td>ChangeUserPassword, LoginDrConfig, LogoutDrConfig, RetrieveContent</td>
</tr>
<tr>
<td>Managed Object</td>
<td>Remarks</td>
<td>Local Methods</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>ConfigServiceManager</td>
<td>Describes the operations to control appliance services.</td>
<td>DrConfigAllServicesStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigRestartService</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigServiceStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigStartService</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigStopService</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsSrmServerRunning</td>
</tr>
<tr>
<td>SraManager</td>
<td>Describes the interface for managing SRA images and containers in the SRM Configuration Service.</td>
<td>CopySraConfiguration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DeleteImage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DeleteImageContainers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetImageInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetRunningTask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetSraImages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ResetToFactorySettings</td>
</tr>
<tr>
<td>SslCertificateManager</td>
<td>Describes operations to configure certificates for the configuration service and SRM.</td>
<td>AddCaCertificates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ClearCaCertificates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigGenerateCSR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigSetCertificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DrConfigSetKeyCertificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InstallCertificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InstallSelfSignedCertificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProbeSsl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveCaCertificates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RetrieveCaCertificates</td>
</tr>
<tr>
<td>UpdateManager</td>
<td>Describes operations to update the Site Recover Manager appliance.</td>
<td>DrConfigCheckForUpdates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetRepositories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetRunningTask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InstallUpdate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UpdateRepository</td>
</tr>
</tbody>
</table>

The Site Recovery Manager Appliance Management Object Classes graphic shows the managed object class hierarchy with the methods of each managed object.
Site Recovery Manager API

The Site Recovery Manager API provides language-neutral interfaces to the Site Recovery Manager server management framework.

The API is implemented as industry-standard Web service, running on Site Recovery Manager server. The Site Recovery Manager API complies with the Web Services Interoperability Organization (WS-I) Basic Profile 1.0, which includes XML Schema 1.0, SOAP 1.1, WSDL 1.1. For more information about the WS-I Basic Profile 1.0, see:

http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

List of API Operations

The following tables provide a list of Site Recovery Manager API methods arranged in alphabetical order.

Table 2-12. Service Instance

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BreakPairing</td>
<td>Remove the connection with the remote SRM server. This operation automatically logs out from the remote site.</td>
</tr>
<tr>
<td>GetSiteName</td>
<td>Get the name of the current local site. (deprecated in 6.5)</td>
</tr>
<tr>
<td>GetPairedSite</td>
<td>Retrieve information about the remote site that is paired with this local site.</td>
</tr>
<tr>
<td>GetLocalSiteInfo</td>
<td>Get information about the local site</td>
</tr>
<tr>
<td>GetSolutionUserInfo</td>
<td>Obtain the Site Recovery Manager solution user name for the local site.</td>
</tr>
<tr>
<td>GetPairedSiteSolutionUserInfo</td>
<td>Obtain the Site Recovery Manager solution user name for the remote site.</td>
</tr>
</tbody>
</table>
### Table 2-12. Service Instance (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetLicenseInfo</td>
<td>Get assigned license information.</td>
</tr>
<tr>
<td>PairSrm</td>
<td>Establish persistent network connection with a remote SRM server. Remote SRM server must have the same VC extension key.</td>
</tr>
<tr>
<td>ProbeSsl</td>
<td>Returns (host and thumbprint) tuples for all PSC/MGMT/DR hosts to which SRM should connect under specified root PSC node.</td>
</tr>
<tr>
<td>ReconfigureConnection</td>
<td>Reconfigure RemoteSite object with connection information for remote PSC node.</td>
</tr>
<tr>
<td>RetrieveContent</td>
<td>Retrieve the properties of a service instance. Additionally the AboutInfo data object provides information about this service.</td>
</tr>
<tr>
<td>SrmLoginByTokenLocale</td>
<td>Begin a session with Site Recovery Manager Server.</td>
</tr>
<tr>
<td>SrmLoginSitesByToken</td>
<td>Log in to both the local and remote vCenter Server.</td>
</tr>
<tr>
<td>SrmLoginRemoteSiteByToken</td>
<td>Log in to remote site when escalated privileges are required and the current session has already been authenticated using SrmLoginSitesByToken.</td>
</tr>
<tr>
<td>SrmLoginLocale</td>
<td>Begin a session with Site Recovery Manager Server.</td>
</tr>
<tr>
<td>SrmLoginSites</td>
<td>Log in to both the local and remote vCenter Server.</td>
</tr>
<tr>
<td>SrmLogoutLocale</td>
<td>Log out sites and terminate the current session.</td>
</tr>
<tr>
<td>SrmLoginRemoteSite</td>
<td>Log in to remote site when escalated privileges are required on the remote site and the current session has already been authenticated using SrmLoginSitesByToken.</td>
</tr>
</tbody>
</table>

### Table 2-13. SRM ExtApi Task

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSrmExtApiTaskInfo</td>
<td>Gets the results of this task</td>
</tr>
<tr>
<td>IsSrmExtApiTaskComplete</td>
<td>Returns True if this task has been completed</td>
</tr>
</tbody>
</table>

### Table 2-14. Inventory Mapping

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddFolderMapping</td>
<td>Add a folder mapping between the primary and secondary vCenter Server.</td>
</tr>
<tr>
<td>AddNetworkMapping</td>
<td>Add a network mapping between the primary and secondary vCenter Server.</td>
</tr>
<tr>
<td>AddResourcePoolMapping</td>
<td>Add resource pool mapping between primary and secondary vCenter Server.</td>
</tr>
<tr>
<td>AddTestNetworkMapping</td>
<td>Add a test network mapping.</td>
</tr>
<tr>
<td>GetFolderMappings</td>
<td>Returns an array of the folder mappings for a specific inventory mapper.</td>
</tr>
</tbody>
</table>
### Table 2-14. Inventory Mapping (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetnetworkMappings</td>
<td>Returns an array of the network mappings for a specific inventory mapper.</td>
</tr>
<tr>
<td>GettestNetworkMappings</td>
<td>Returns an array of the test network mappings for a specific inventory mapper.</td>
</tr>
<tr>
<td>GetresourcePoolMappings</td>
<td>Returns an array of the resource pool mappings for a specific inventory mapper.</td>
</tr>
<tr>
<td>RemoveFolderMapping</td>
<td>Remove a folder mapping.</td>
</tr>
<tr>
<td>RemoveNetworkMapping</td>
<td>Remove a network mapping.</td>
</tr>
<tr>
<td>RemoveResourcePoolMapping</td>
<td>Remove a resource pool mapping.</td>
</tr>
<tr>
<td>RemoveTestNetworkMapping</td>
<td>Remove a test network mapping.</td>
</tr>
</tbody>
</table>

### Table 2-15. Storage

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateArrayManager</td>
<td>Creates ArrayManager object.</td>
</tr>
<tr>
<td>DiscoverDevices</td>
<td>Loops through all array managers, local and remote, discovering devices.</td>
</tr>
<tr>
<td>QueryArrayManagers</td>
<td>Returns a list of all the available array managers.</td>
</tr>
<tr>
<td>QueryStorageAdapters</td>
<td>List of Storage Replication Adapters (SRAs) info successfully loaded into SRM.</td>
</tr>
<tr>
<td>ReloadAdapters</td>
<td>Scans SRA installation directory and re-loads SRAs.</td>
</tr>
<tr>
<td>RemoveArrayManager</td>
<td>Deletes ArrayManager object.</td>
</tr>
</tbody>
</table>

### Table 2-16. Storage Adapter

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FetchInfo</td>
<td>Fetches basic information about the SRA.</td>
</tr>
<tr>
<td>GetAdapterConnectionSpec</td>
<td>Gets the connection parameters for the SRA provided by the user.</td>
</tr>
</tbody>
</table>

### Table 2-17. Autoprotect Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAutoprotectUser</td>
<td>Gets the user for the automatic protection.</td>
</tr>
<tr>
<td>IsActive</td>
<td>Checks if the automatic protection is globally activated (true) or deactivated (false).</td>
</tr>
<tr>
<td>SetAutoprotectUser</td>
<td>Configures the user to be used by automatic protection on this site. If not called, the default autoprotect user is used.</td>
</tr>
<tr>
<td>SetDefaultAutoprotectUser</td>
<td>Reverts the current user to the default user.</td>
</tr>
</tbody>
</table>
### Table 2-18. Folder

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetName</td>
<td>Get the name of ProtectionGroupFolder or RecoveryPlanFolder.</td>
</tr>
<tr>
<td>GetParentFolder</td>
<td>Get parent folder for a ProtectionGroupFolder or RecoveryPlanFolder.</td>
</tr>
<tr>
<td>GetChildType</td>
<td>Specifies the object types a folder may contain.</td>
</tr>
<tr>
<td>CreateFolder</td>
<td>Creates a new sub-folder with the specified name.</td>
</tr>
<tr>
<td>MoveFolder</td>
<td>Moves the specified folder into another folder.</td>
</tr>
<tr>
<td>DestroyFolder</td>
<td>Destroys the specified folder.</td>
</tr>
<tr>
<td>RenameFolder</td>
<td>Renames the specified Folder.</td>
</tr>
</tbody>
</table>

### Table 2-19. Protection

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateAbrProtectionGroup</td>
<td>Create an array-based replication (ABR) protection group. The method returns a CreateProtectionGroupTask.</td>
</tr>
<tr>
<td>CreateHbrProtectionGroup</td>
<td>This method is deprecated. It creates a host-based replication (HBR) protection group. Returns a CreateProtectionGroupTask.</td>
</tr>
<tr>
<td>CreateHbrProtectionGroup2</td>
<td>Create a new host based (that is vSphere replication) ProtectionGroup using the provided virtual machines.</td>
</tr>
<tr>
<td>CreateVvolProtectionGroup</td>
<td>Creates a new vVol protection group.</td>
</tr>
<tr>
<td>GetProtectionGroupRootFolder</td>
<td>Get the root folder for all protection groups, so as to navigate folder hierarchy. The methods below give users the ability to locate replicated resources and construct protection groups, key features of the 5.8 API.</td>
</tr>
<tr>
<td>ListProtectionGroups</td>
<td>Get a list of the protection groups that are currently configured.</td>
</tr>
<tr>
<td>ListInventoryMappings</td>
<td>Get a list of the configured inventory mappings on the protection site.</td>
</tr>
<tr>
<td>ListReplicatedDatastores</td>
<td>Get a list of the replicated datastores. (deprecated in 6.0)</td>
</tr>
<tr>
<td>ListUnassignedReplicatedDatastores</td>
<td>Get list of replicated datastores that can be used to create new protection groups.</td>
</tr>
<tr>
<td>ListUnassignedReplicatedVms</td>
<td>Get list of replicated VMs not currently assigned to a Site Recovery Manager protection group.</td>
</tr>
<tr>
<td>ProtectionListProtectedDatastores</td>
<td>Get list of replicated datastores that are protected by Site Recovery Manager.</td>
</tr>
<tr>
<td>ProtectionListProtectedVms</td>
<td>Get list of virtual machines that are protected by Site Recovery Manager.</td>
</tr>
<tr>
<td>RemoveProtectionGroup</td>
<td>Delete a protection group.</td>
</tr>
</tbody>
</table>
### Table 2-20. Protection Group Folder

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetProtectionGroup</td>
<td>Retrieves the protection group with the specified name, if any.</td>
</tr>
<tr>
<td>ListChildProtectionGroupFolders</td>
<td>Return the child ProtectionGroupFolders located in this folder.</td>
</tr>
<tr>
<td>ListChildProtectionGroups</td>
<td>Return the SrmProtectionGroup objects located in this folder.</td>
</tr>
</tbody>
</table>

### Table 2-21. Create Protection Group Task

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCreateProtectionGroupResult</td>
<td>Once task is complete, check the result to ensure that it succeeded.</td>
</tr>
<tr>
<td>GetNewProtectionGroup</td>
<td>After checking task result, obtain the newly created SrmProtectionGroup.</td>
</tr>
<tr>
<td>IsCreateProtectionGroupComplete</td>
<td>Check completeness of the task to create a new protection group.</td>
</tr>
</tbody>
</table>

### Table 2-22. Protection Group

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssociateVms</td>
<td>Associate the specified VMs with a group. This is a prerequisite for protection. Only for vSphere Replication.</td>
</tr>
<tr>
<td>AddDatastores</td>
<td>Adds datastores to the protection group. Additionally, the virtual machines on these datastores can be protected by the protection group. This can be done by calling protectVms method from this interface.</td>
</tr>
<tr>
<td>CheckConfigured</td>
<td>Check the protection group for VMs that are not configured, have configuration issues, and protected VMs that must be configured.</td>
</tr>
<tr>
<td>GetInfo</td>
<td>Retrieve basic information about this protection group.</td>
</tr>
<tr>
<td>GetPeer</td>
<td>Retrieve the peer protection group.</td>
</tr>
<tr>
<td>GetPlaceholderVmInfo</td>
<td>This method returns information for the placeholder VM for the specified protected VM.</td>
</tr>
<tr>
<td>GetRecoveryLocationSettings</td>
<td>This method returns the recovery location settings for the specified protected VM.</td>
</tr>
<tr>
<td>GetProtectionState</td>
<td>Get the current state of the protection group.</td>
</tr>
<tr>
<td>GetVvolGroupDetails</td>
<td>Gets vVol specific details for this protection group.</td>
</tr>
<tr>
<td>ListProtectedVms</td>
<td>List VMs protected in this group with information about their protection state.</td>
</tr>
<tr>
<td>ListProtectedDatastores</td>
<td>Retrieve a list of the Datastores protected by this protection group.</td>
</tr>
<tr>
<td>ListAssociatedVms</td>
<td>Retrieve a list of VMs associated with this group. Only for vSphere Replication and vVol.</td>
</tr>
</tbody>
</table>
### Table 2-22. Protection Group (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveGroup</strong></td>
<td>This method moves specified ProtectionGroup to a different folder.</td>
</tr>
<tr>
<td><strong>ProtectionGroupGetParentFolder</strong></td>
<td>Retrieve the folder that contains this protection group.</td>
</tr>
<tr>
<td><strong>ProtectionGroupListRecoveryPlans</strong></td>
<td>Retrieve a list of all Recovery Plans this protection group is a member of.</td>
</tr>
<tr>
<td><strong>ProtectionGroupQueryVmProtection</strong></td>
<td>Determine whether the specified VMs can be or currently are protected, which must be mapped to the recovery site as per ListInventoryMappings.</td>
</tr>
<tr>
<td><strong>ProtectVms</strong></td>
<td>Protect the specified VMs. The folder, resource pool, and network of each virtual machine must be mapped to the recovery site. Returns a ProtectionTask.</td>
</tr>
<tr>
<td><strong>ProtectionGroupGetOperationalLocation</strong></td>
<td>Get the effective location of the protection group.</td>
</tr>
<tr>
<td><strong>RemoveDatastores</strong></td>
<td>Removes datastores from the protection group. Virtual machines on the removed datastores are no longer protected by the protection group.</td>
</tr>
<tr>
<td><strong>ReconfigureRecoveryLocationSettings</strong></td>
<td>Reconfigures the recovery location settings for the specified protected VM.</td>
</tr>
<tr>
<td><strong>ReconfigureVvolProtectionGroup</strong></td>
<td>Reconfigure settings for this group. For a vVvol ProtectionGroup this method can reconfigure the name, description, fault domain, and associated virtual machines. If the clearAssociatedVms flag is set to true, then the associated virtual machines with this protection group will be cleared. If the flag is not set and the associated Vm's list is not empty, then the associated virtual machines will be added to this protection group replacing the old virtual machines.</td>
</tr>
<tr>
<td><strong>RecreatePlaceholder</strong></td>
<td>Recreates a placeholder VM.</td>
</tr>
<tr>
<td><strong>UnprotectVms</strong></td>
<td>Unprotect the specified VMs.</td>
</tr>
<tr>
<td><strong>UnassociateVms</strong></td>
<td>Unassociate the specified VMs with this group. Only for vSphere Replication and vVol.</td>
</tr>
</tbody>
</table>

### Table 2-23. Protection Task

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetProtectionStatus</strong></td>
<td>Get the results of ProtectVms or UnprotectVms</td>
</tr>
<tr>
<td><strong>GetTasks</strong></td>
<td>Get Task information from the vCenter Server for each virtual machine that was requested to be protected or unprotected.</td>
</tr>
<tr>
<td><strong>GetResult</strong></td>
<td>Get the results of this Task.</td>
</tr>
<tr>
<td><strong>IsComplete</strong></td>
<td>Check if this Task has finished.</td>
</tr>
</tbody>
</table>
### Table 2-24. Recovery

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateRecoveryPlan</td>
<td>Create a recovery plan.</td>
</tr>
<tr>
<td>DeleteRecoveryPlan</td>
<td>Delete a recovery plan.</td>
</tr>
<tr>
<td>GetHistory</td>
<td>Retrieve the history for a given Recovery Plan.</td>
</tr>
<tr>
<td>GetRecoveryPlanRootFolder</td>
<td>Retrieve the root folder for all Recovery Plans.</td>
</tr>
<tr>
<td>ListPlans</td>
<td>Retrieve all the Recovery Plans for Site Recovery Manager Server.</td>
</tr>
<tr>
<td>MovePlan</td>
<td>Moves the RecoveryPlan to a different folder.</td>
</tr>
</tbody>
</table>

### Table 2-25. Recovery Plan Folder

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRecoveryPlan</td>
<td>Get MoRef to recovery plan with the specified name in the RecoveryPlanFolder.</td>
</tr>
<tr>
<td>ListChildRecoveryPlanFolders</td>
<td>Return the child RecoveryPlanFolders located in the folder.</td>
</tr>
<tr>
<td>ListChildRecoveryPlans</td>
<td>Return an array of SrmRecoveryPlan objects located in the folder.</td>
</tr>
</tbody>
</table>

### Table 2-26. Recovery Plan

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddProtectionGroup</td>
<td>Add a protection group to this Recovery Plan.</td>
</tr>
<tr>
<td>AddTestNetworkMappingToRecoveryPlan</td>
<td>Add a test network mapping to a recovery plan.</td>
</tr>
<tr>
<td>AnswerPrompt</td>
<td>Answer the current prompt displayed by a Recovery Plan. Requires the Run privilege for test, or the Failover privilege for the other modes.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Cancel the specified Recovery Plan.</td>
</tr>
<tr>
<td>GetRecoverySettings</td>
<td>Retrieve the per-VM recovery settings for VMs in the Recovery Plan.</td>
</tr>
<tr>
<td>ListPrompts</td>
<td>List the current prompts that are waiting for input. When a prompt step is reached, the plan goes into the waiting state until AnswerPrompt is received. Prompts are given in the same order in which VMs are scheduled to start up.</td>
</tr>
<tr>
<td>RecoveryPlanGetPeer</td>
<td>Get the peer plan for this Recovery Plan. The returned object refers to a plan at the paired site, not the local site.</td>
</tr>
<tr>
<td>RecoveryPlanGetInfo</td>
<td>Retrieve basic information about the specified Recovery Plan.</td>
</tr>
<tr>
<td>RecoveryPlanGetParentFolder</td>
<td>Retrieve the root folder for all Recovery Plans.</td>
</tr>
<tr>
<td>RemoveProtectionGroupFromRecoveryPlan</td>
<td>Remove a protection group from a recovery plan.</td>
</tr>
<tr>
<td>RemoveTestNetworkMappingFromRecoveryPlan</td>
<td>Remove a test network mapping from a recovery plan.</td>
</tr>
<tr>
<td>RecoveryPlanGetLocation</td>
<td>Check whether the recovery plan is hosted locally or on the paired site.</td>
</tr>
</tbody>
</table>
### Table 2-26. Recovery Plan (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecoveryPlanHasRunningTask</td>
<td>Check whether there is a task that is associated with the recovery plan.</td>
</tr>
<tr>
<td>Start</td>
<td>Start the Recovery Plan in a selected mode: test, cleanupTest, recovery, or reprotect. Requires Run privilege for tests, and the Failover privilege for the others.</td>
</tr>
<tr>
<td>SetRecoverySettings</td>
<td>Modify the per-VM recovery settings for VMs in the Recovery Plan. Configure the IP address and corresponding DNS, WINS of the virtual machine, after the migration is complete, using the VmlpCustomization API.</td>
</tr>
</tbody>
</table>

### Table 2-27. Recovery History

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRecoveryResult</td>
<td>Retrieve the recovery result for a given run of a Recovery Plan.</td>
</tr>
<tr>
<td>GetResultCount</td>
<td>Retrieve total number of stored results, including Recovery and peer plans.</td>
</tr>
<tr>
<td>GetResultLength</td>
<td>Get length of XML result document for the requested recovery result.</td>
</tr>
<tr>
<td>RetrieveStatus</td>
<td>Retrieve XML document for a historical run of the specified Recovery Plan.</td>
</tr>
</tbody>
</table>

### Table 2-28. IP Subnet Mapper

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddIpMapping</td>
<td>Associates an IPMapping object with an inventory-mapped protected site network.</td>
</tr>
<tr>
<td>GetIpSubnetMappings</td>
<td>Returns an array of the IP subnet mappings for this IP Subnet Mapper.</td>
</tr>
<tr>
<td>RemoveIpMappings</td>
<td>Removes IPMappings from mapped protected site networks. This must be called on secondary (recovery) site.</td>
</tr>
</tbody>
</table>

### Table 2-29. Array Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddArrayPair</td>
<td>Creates ReplicatedArrayPair object for a given pair of storage arrays.</td>
</tr>
<tr>
<td>DiscoverArrays</td>
<td>Discovers storage arrays configured for replication by executing SRA command discoverArrays.</td>
</tr>
<tr>
<td>GetAdapter</td>
<td>Returns the corresponding storage adapter to the ArrayManager.</td>
</tr>
<tr>
<td>GetArrayDiscoveryStatus</td>
<td>Returns the status and timestamp information of latest array discovery.</td>
</tr>
<tr>
<td>GetArrayInfo</td>
<td>This method gets the list of discovered storage arrays.</td>
</tr>
</tbody>
</table>
### Table 2-29. Array Manager (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryReplicatedArrayPairs</td>
<td>Returns list of all the replicated array pairs in the ArrayManager.</td>
</tr>
<tr>
<td>ReadInfo</td>
<td>Returns information specific to the ArrayManager instance.</td>
</tr>
<tr>
<td>Reconfigure</td>
<td>Updates array manager name and connection parameters for the SRA.</td>
</tr>
<tr>
<td>RemoveArrayPair</td>
<td>Deletes specified ReplicatedArrayPair object.</td>
</tr>
</tbody>
</table>

### Table 2-30. Replicated Array Pair

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDevices</td>
<td>Returns list of storage devices configured for replication.</td>
</tr>
<tr>
<td>GetDeviceGroups</td>
<td>Returns list of consistency groups of storage devices configured for replication.</td>
</tr>
<tr>
<td>GetReplicatedDatastores</td>
<td>Returns list of datastores residing on replicated storage devices.</td>
</tr>
<tr>
<td>GetDeviceDiscoveryStatus</td>
<td>Gets the storage device discovery status.</td>
</tr>
<tr>
<td>QueryReplicatedRdms</td>
<td>Returns info for all replicated RDMs in the ReplicatedArrayPair.</td>
</tr>
</tbody>
</table>

### Table 2-31. Vvol Replication

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDomains</td>
<td>Returns a list of local vVol fault domains with their replication groups which target fault domains matching SRM peer site.</td>
</tr>
<tr>
<td>GetUnprotectedVms</td>
<td>Returns a list of unprotected vVol replicated virtual machines part of vVol replication groups that target the SRM peer site.</td>
</tr>
<tr>
<td>Rescan</td>
<td>Initiates a rescan of the server's local vVol configuration. The server keeps updating its view of the local vVol configuration periodically. This results in the newly provisioned vVol virtual machines being available for protection only after the passage of update interval. This function is called to force an update.</td>
</tr>
</tbody>
</table>

### Table 2-32. Placeholder Datastore Manager

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDatastore</td>
<td>Adds datastore to the list of placeholder datastores.</td>
</tr>
<tr>
<td>GetPlaceholderDatastores</td>
<td>Gets the list of all configured placeholder datastores.</td>
</tr>
<tr>
<td>RemoveDatastore</td>
<td>Removes datastore(s) from the list of placeholder datastores.</td>
</tr>
</tbody>
</table>
### Table 2-33. Deprecated DisasterRecoveryApi

<table>
<thead>
<tr>
<th>Method</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetApiVersion</td>
<td>Obtain the API version.</td>
</tr>
<tr>
<td>GetFinalStatus</td>
<td>Get the final status of a Recovery Plan.</td>
</tr>
<tr>
<td>Login, Logout, LoginByToken</td>
<td>Log in to and out of Site Recovery Manager Server.</td>
</tr>
<tr>
<td>ListRecoveryPlans</td>
<td>Get a list of Recovery Plans at the SRM site.</td>
</tr>
<tr>
<td>RecoveryPlanSettings</td>
<td>Get the settings of a specific Recovery Plan at the SRM site.</td>
</tr>
<tr>
<td>RecoveryPlanStart</td>
<td>Start a specific Recovery Plan in recovery or test mode.</td>
</tr>
<tr>
<td>RecoveryPlanPause</td>
<td>Pause a running Recovery Plan.</td>
</tr>
<tr>
<td>RecoveryPlanResume</td>
<td>Restart a paused Recovery Plan.</td>
</tr>
<tr>
<td>RecoveryPlanAnswerPrompt</td>
<td>Answer a prompt.</td>
</tr>
<tr>
<td>RecoveryPlanCancel</td>
<td>Cancel a Recovery Plan.</td>
</tr>
</tbody>
</table>

### New vCenter Single Sign-On APIs

A set of login-by-token functions was added to the ServiceInstance managed object. For an example of use, see the Functions for Logging Into Sites [Logging into Sites with SAML Tokens](#).

### Deprecated APIs

The version 1.0 DisasterRecoveryApi was discontinued in Site Recovery Manager 5.8 and marked deprecated in Site Recovery Manager 6.0, although a new login-by-token function was implemented for backward compatibility.

**Note** The InvalidLogin fault and others use a different namespace in DisasterRecoveryApi (drextapi.fault.InvalidLogin) than in ServiceInstance (vim.fault.InvalidLogin).

In the RemoteSite managed object, the vcHost and vcPort fields are deprecated. They are replaced by the lkpUrl (Lookup Service URL) and vcInstanceUUID (vCenter Server unique ID).

The GetSiteName method is deprecated in Site Recovery Manager 6.5. You must use LocalSiteInfo.siteInfo to get the local site name.

The SrmProtection.listReplicatedDatastores method is replaced with SrmProtection.listUnassignedReplicatedDatastores.

In the Protection managed object, the createHbrProtectionGroup method is deprecated. It is replaced by the createHbrProtectionGroup2 method.

### Managed Object Hierarchy

The following table shows the managed object hierarchy of the Site Recovery Manager API with the methods of each managed object in an alphabetical order.
<table>
<thead>
<tr>
<th>Managed Object</th>
<th>Remarks</th>
<th>Local Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayManager</td>
<td>Query information about array managers</td>
<td>AddArrayPair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DiscoverArrays</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetAdapter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetArrayDiscoveryStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetArrayInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ReadInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reconfigure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveArrayPair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QueryReplicatedArrayPairs</td>
</tr>
<tr>
<td>SrmAutomaticProtection</td>
<td>External API for automatic protection</td>
<td>GetAutoprotectUser</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsActive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetAutoprotectUser</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetDefaultAutoprotectUser</td>
</tr>
<tr>
<td>CreateRecoveryPlanTask</td>
<td>Contains the status of the operation</td>
<td>GetCreateRecoveryPlanFailure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetNewRecoveryPlan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsCreateRecoveryPlanComplete</td>
</tr>
<tr>
<td>CreateProtectionGroupTask</td>
<td>Handle an ABR or HBR protection group</td>
<td>GetCreateProtectionGroupResult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetNewProtectionGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsCreateProtectionGroupComplete</td>
</tr>
<tr>
<td>DeleteRecoveryPlanTask</td>
<td>Contains the status of the operation</td>
<td>GetDeleteRecoveryPlanFailure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsDeleteRecoveryPlanComplete</td>
</tr>
<tr>
<td>SrmRecoveryApi1</td>
<td>Old version 1.0 API, deprecated but still provided for backward compatibility</td>
<td>GetApiVersion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetFinalStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListRecoveryPlans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanAnswerPrompt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanCancel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanPause</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanResume</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanSettings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanStart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLogin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginByToken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLogout</td>
</tr>
<tr>
<td>DiscoverDevicesTask</td>
<td>Contains the status of the operation</td>
<td>GetDiscoverDevicesTaskFailures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsDiscoverDevicesTaskComplete</td>
</tr>
<tr>
<td>Folder</td>
<td>Site Recovery Manager folder class</td>
<td>CreateFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DestroyFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetChildType</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetParentFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MoveFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RenameFolder</td>
</tr>
</tbody>
</table>
### Table 2-34. Managed Object Hierarchy (continued)

<table>
<thead>
<tr>
<th>Managed Object</th>
<th>Remarks</th>
<th>Local Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtectionGroupFolder</td>
<td>Site Recovery Manager folder for protection groups</td>
<td>GetProtectionGroup, ListChildProtectionGroupFolders, ListChildProtectionGroups</td>
</tr>
<tr>
<td>ProtectionTask</td>
<td>Handle VM protection</td>
<td>GetProtectionStatus, GetResult, GetTasks, IsComplete</td>
</tr>
<tr>
<td>RemoveProtectionGroupTask</td>
<td>Handle protection group removal</td>
<td>GetRemoveProtectionGroupResult, IsRemoveProtectionGroupComplete</td>
</tr>
<tr>
<td>ReplicatedArrayPair</td>
<td>Query info about RDM devices</td>
<td>GetDevices, GetDeviceGroups, GetDeviceDiscoveryStatus, GetReplicatedDatastores, QueryReplicatedRdms</td>
</tr>
<tr>
<td>SrmExtApiTask</td>
<td>Base external API task</td>
<td>IsSrmExtApiTaskComplete, GetSrmExtApiTaskInfo</td>
</tr>
<tr>
<td>SrmRecovery</td>
<td>Query recovery plans</td>
<td>CreateRecoveryPlan, DeleteRecoveryPlan, GetHistory, GetRecoveryPlanRootFolder, ListPlans, MovePlan</td>
</tr>
<tr>
<td>Managed Object</td>
<td>Remarks</td>
<td>Local Methods</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SrmRecoveryPlan</td>
<td>Run a recovery plan</td>
<td>AddProtectionGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AddTestNetworkMappingToRecoveryPlan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AnswerPrompt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cancel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetRecoverySettings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListPrompts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanGetInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanGetPeer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanGetParentFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanGetLocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveProtectionGroupFromRecoveryPlan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecoveryPlanHasRunningTask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveTestNetworkMappingFromRecoveryPlan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetRecoverySettings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start</td>
</tr>
<tr>
<td>SrmStorageAdapter</td>
<td>Gets information about a storage adapter</td>
<td>FetchInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetAdapterConnectionSpec</td>
</tr>
<tr>
<td>SrmStorage</td>
<td>Access the storage</td>
<td>CreateArrayManager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DiscoverDevices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QueryArrayManagers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QueryStorageAdapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ReloadAdapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveArrayManager</td>
</tr>
<tr>
<td>SrmRecoveryHistory</td>
<td>Recovery plan status</td>
<td>GetRecoveryResult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetResultCount</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetResultLength</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RetrieveStatus</td>
</tr>
<tr>
<td>Managed Object</td>
<td>Remarks</td>
<td>Local Methods</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>SrmServiceInstance</td>
<td>Open or close session, get information about local and remote sites</td>
<td>BreakPairing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetLicenseInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetPairedSite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetSiteName (deprecated in 6.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PairSrm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProbeSsl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ReconfigureConnection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RetrieveContent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginLocale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginRemoteSite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginSites</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLogoutLocale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetLocalSiteInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetSolutionUserInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetPairedSiteSolutionUserInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginByTokenLocale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginRemoteSiteByToken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SrmLoginSitesByToken</td>
</tr>
<tr>
<td>SrmProtection</td>
<td>Create an ABR or HBR protection group, list inventory mappings, query datastores and VMs, and list protection groups</td>
<td>CreateAbrProtectionGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CreateHbrProtectionGroup (deprecated in 8.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CreateVvolProtectionGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetProtectionGroupRootFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListInventoryMappings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListProtectionGroups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListReplicatedDatastores (deprecated in 6.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListUnassignedReplicatedDatastores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListUnassignedReplicatedVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionListProtectedDatastores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionListProtectedVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveProtectionGroup</td>
</tr>
<tr>
<td>Managed Object</td>
<td>Remarks</td>
<td>Local Methods</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>SrmProtectionGroup</td>
<td>Add virtual machines to a protection group, get peer, query protected datastores, add datastore, and remove datastore</td>
<td>AddDatastores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AssociateVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CheckConfigured</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetInfoGetPeer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetPeer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetPlaceholderVmInfo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetProtectionState</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetRecoveryLocationSettings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetVvolGroupDetails</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListAssociatedVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListProtectedDatastores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListProtectedVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MoveGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionGroupGetOperationalLocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionGroupGetParentFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionGroupListRecoveryPlans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectionGroupQueryVmProtection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProtectVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ReconfigureRecoveryLocationSettings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ReconfigureVvolProtectionGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecreatePlaceholderVmmGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveDatastores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UnassociateVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UnprotectVms</td>
</tr>
<tr>
<td>SrmVvolReplication</td>
<td>Provide information about the local vVol topology replicated to the SRM peer site</td>
<td>GetDomains</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetUnprotectedVms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rescan</td>
</tr>
<tr>
<td>SrmPlaceholderDatastoreManager</td>
<td>Manages placeholder datastores</td>
<td>AddDatastore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoveDatastore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetPlaceholderDatastores</td>
</tr>
</tbody>
</table>

The SRM Object Classes graphic shows the Site Recovery Manager managed object class hierarchy with the methods of each managed object.
Figure 2-2. Site Recovery Manager API Object Classes

Logging into Sites with SAML Tokens

Site Recovery Manager release 6.0 improves security by obtaining a security assertion markup language (SAML) token from the vCenter Single Sign-On service for both the local and remote sites.
Table 2-35. Functions for Logging Into Sites

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSolutionUserInfo</td>
<td>Obtain the UUID of Site Recovery Manager Server and the Site Recovery Manager solution user name.</td>
</tr>
<tr>
<td>SrmLoginByTokenLocale</td>
<td>After obtaining a token from vCenter Single Sign-On, begin session with the local Site Recovery Manager Server</td>
</tr>
<tr>
<td>GetPairedSiteSolutionUserInfo</td>
<td>Obtain the remote UUID of Site Recovery Manager Server and the solution user name</td>
</tr>
<tr>
<td>SrmLoginRemoteSiteByToken</td>
<td>After obtaining remote token, begin session with the paired Site Recovery Manager Server</td>
</tr>
<tr>
<td>SrmLoginSitesByToken</td>
<td>Log in to both local and remote Site Recovery Manager Server, passing both SAML tokens</td>
</tr>
</tbody>
</table>

The following figure shows the sequence of calling for LoginSitesByToken

**Figure 2-3. Calling Sequence for LoginSitesByToken**

Order of operations

1. Obtain local token from the vCenter Single Sign-On service located on the local Platform Services Controller.
2. Get remote site information from Site Recovery Manager, and extract the URL of remote LookupService.
3. Use remote LookupService to find the remote vCenter Single Sign-On service.
4. Obtain remote access SAML token from vCenter Single Sign-On service located on the remote Platform Services Controller.
5. Make the SrmLoginSitesByToken call locally to Site Recovery Manager.
WSDL Programming Environments

You can program Web services and read WSDL files using the C# language with Visual Studio .NET, or using the Java language with the Axis framework or the JAX-WS framework. You can program Web services using many other languages and frameworks, but they are beyond the scope of this manual.

Java JAX-WS Framework

The SDK provides sample code that uses the Java Development Kit (JDK) 1.6 with the JAX-WS framework bundled with the JDK 1.6. The build scripts generate Java stubs from the Site Recovery Manager specific WSDL.

**Note** For Site Recovery Manager Appliance Management APIs, use JDK 1.8.0_202.

C# and Visual Studio

The Site Recovery Manager SDK provides sample C# .NET code prepared for use with Visual Studio 2008, which you can convert for use with Visual Studio 2010 and perhaps later versions as well.

**Note** For Site Recovery Manager Appliance Management APIs, use Microsoft Visual Studio 2017.

Java Axis Framework

The Site Recovery Manager SDK provides legacy sample code that requires Java SE 1.5 or later and Apache Axis 1.4. Samples are set up for stub generation on Windows or on Linux.

**Note** For Site Recovery Manager Appliance Management APIs, use Apache Axis2.

Managed Objects as WSDL

The WSDL Programming Components graphic shows the WSDL programming components used by various language frameworks.
Site Recovery Manager managed objects and methods are derived from classes and methods in the vSphere API, also known as the virtual machine object description language (VMODL). In the SDK, the Site Recovery Manager interfaces are mixed in with vSphere interfaces. For specific information about vSphere interfaces, see the vSphere API Reference manual, which is in the vSphere Web Services SDK under the VMware vSphere Management SDK.

Accessing Site Recovery Manager APIs

The Site Recovery Manager Appliance Management API and Site Recovery Manager API provides language-neutral interfaces for configuring Site Recovery Manager Server or OS-specific settings, managing protection groups, and recovery plans. Array-based replication, vSphere Replication, and vVols are supported.

Location of the API

The Site Recovery Manager 8.3 API is located at the following endpoints:

- Site Recovery Manager Appliance Management API
  
  https://<FQDN_Server_or_IP_Address>:5480/configureserver/sdk

- Site Recover Manager APIs for Windows
  
  https://<FQDN_Server_or_IP_Address>:9086/vcdr/extapi/sdk

- Site Recover Manager APIs for Photon Virtual Appliance (VA)
  
  https://<FQDN_Server_or_IP_Address>:443/drserver/vcdr/extapi/sdk

All services use this single network port, and all communications are TLS encrypted. SSLv3 is disabled for security reasons. The API is implemented as an industry-standard Web service running on Site Recovery Manager Server.
The API complies with the Web Services Interoperability Organization (WS-I) Basic Profile 1.0, which includes XML Schema 1.0, SOAP version 1.1, and WSDL version 1.1. For details about WS-I Basic Profile 1.0, see the http://www.ws-i.org website.

**Obtaining WSDL for APIs**

- **WSDL for Site Recovery Manager Appliance Management API:** Request the file `drconfig-service.wsdl` from the server root path.
  
  \[https://<FQDN_Server_or_IP_Address>:5480/drconfig-service.wsdl\]

- **WSDL for Site Recovery Manager API:** Request the file `srm-Service.wsdl` from the server root path.
  
  - On a Windows server, the root path can be \[https://<FQDN_Server_or_IP_Address>:9086/srm-Service.wsdl\]
  
  - On a Virtual Appliance, the root path can be \[https://<FQDN_Server_or_IP_Address>:443/drserver/srm-Service.wsdl\]

**Associated vCenter Servers for Site Recovery Manager API**

As of SRM 6.0, Platform Services Controller and vCenter Server are associated with the Site Recovery Manager Server at both the local (protected) and the remote (recovery) sites.

Platform Services Controller can be embedded in vCenter Server, or it can be hosted on a separate machine. Platform Services Controller performs three services: Lookup, vCenter Single Sign-On, and Licensing.

The vCenter Server performs tagging and authorization for Site Recovery Manager. A system administrator installs the Site Recovery Manager plug-in at both local and remote sites to control the site’s Site Recovery Manager Server through vCenter Server.

Managed object `SrmServiceInstance` provides functions for local and remote site discovery. You obtain the local site information with `getLocalSiteInfo`, and obtain the local solution user with `GetSolutionUserInfo`.

The local Platform Services Controller LookupService does not know anything about services on the remote site. You obtain the remote site name with `GetPairedSite` and obtain the remote solution user with `GetPairedSiteSolutionUserInfo`. The RemoteSite object contains the URL of the remote LookupService, and the UUID of the remote vCenter Server.
The SDK Installation and Setup chapter describes how to unpack and use the software development kit (SDK).

This chapter includes the following topics:

- Contents of the SDK Package
- SDK Directory Structure
- Download and Setup

Contents of the SDK Package

The Site Recovery Manager SDK is delivered as a ZIP archive (VMware-{version}-<build>.zip file.

You can obtain the SDK package by navigating to http://www.vmware.com/support/developer/srm-api and clicking the Download SDK link. You must provide an email address or customer number, with a valid password, for the authentication on the Site Recovery Manager download site.

The package contains:

- Sample code demonstrating common use cases for programmatically managing the Site Recovery Manager server. The sample code includes Java and C# source code files. See the following Readme files for information about building and using the samples:
  - doc/srm/readme_dotnet.htm
  - doc/srm/readme_java.htm
  - doc/srm/readme_jaxws.htm

- Sample code demonstrating common use cases for programmatically configuring the Site Recovery Manager Virtual Appliance. The sample code includes Java and C# source code files. See the following Readme files for information about building and using the samples:
  - doc/drconfig/readme_jaxws.htm
  - doc/drconfig/readme_java.htm
  - doc/drconfig/readme_dotnet.htm
- The WSDL and XML schema files that define the Site Recovery Manager API and Site Recovery Manager Appliance Management API.

- Batch files and shell scripts to automate the process of generating client-side stubs, and for rebuilding the sample applications. For C# developers, the Microsoft Visual Studio project files (.sln) are included.

- Documentation, including VMware Site Recovery Manager API Reference Guide and VMware Site Recovery Manager Appliance Management API Reference Guide, that provides a language-neutral descriptive information about object type definitions, properties, and method signatures for the VMware Site Recovery Manager API 8.3.

## SDK Directory Structure

After you unzip the Site Recovery Manager SDK, the following directories and sub-directories appear. Many of the sub-directories contain helpful readme files.

### Table 3-1. SDK Directory Structure

<table>
<thead>
<tr>
<th>Directory or File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/doc</td>
<td>Contains SDK README files and reference documentation for the SDK.</td>
</tr>
<tr>
<td>/doc/srm/ReferenceGuide</td>
<td>API Reference for the Site Recovery Manager API. To view the API Reference, open index.html with a Web browser.</td>
</tr>
<tr>
<td>/doc/SDK_Terms_and_Conditions.*</td>
<td>End-user license agreement for the Site Recovery Manager SDK.</td>
</tr>
<tr>
<td>/samples</td>
<td>Top-level directory for language-specific versions of sample client applications.</td>
</tr>
<tr>
<td>samples/srm/DotNet</td>
<td>Directory containing command scripts to generate the .NET proxy classes and Web service stubs for the VMware Site Recovery Manager API. The GeneratingStubs.txt file gives helpful notes about how to generate stubs with your own namespace for Visual Studio 2008.</td>
</tr>
<tr>
<td>samples/srm/DotNet/cs</td>
<td>Directory containing Visual Studio 2008 solution (.sln) file and sub-directories with C# AppUtil support code and RecoveryPlan.cs sample application with project (.csproj) file VMware Site Recovery Manager API.</td>
</tr>
<tr>
<td>/samples/srm/JAXWS</td>
<td>Directory containing Java source code for the JAX-WS framework for the VMware Site Recovery Manager API. Sample program RecoveryPlanList.java is in the com/vmware/samples/recovery subdirectory. Shell scripts and batch files are provided to build and run the sample program.</td>
</tr>
</tbody>
</table>
### Table 3-1. SDK Directory Structure (continued)

<table>
<thead>
<tr>
<th>Directory or File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/samples/srm/Axis</td>
<td>Directory containing Java source code for the Axis framework for the VMware Site Recovery Manager API. Sample program RecoveryPlanList.java is in the <code>java\com\vmware\samples\recovery</code> subdirectory. Shell scripts and batch files are provided to build and run the sample program.</td>
</tr>
<tr>
<td>/wsdl/srm/srm.wsdl</td>
<td>The Web Services Description Language (WSDL) file containing definition of the VMware Site Recovery Manager API.</td>
</tr>
<tr>
<td>/wsdl/srm/srm-Service.wsdl</td>
<td>A WSDL file defining the Web services endpoint at which the VMware Site Recovery Manager API is available. This file references the srm.wsdl with an import statement, so you will use the appropriate generation tool with srm-Service.wsdl (rather than srm.wsdl directly).</td>
</tr>
<tr>
<td>/wsdl/srm/*.xsd</td>
<td>XML schema definition files (six).</td>
</tr>
<tr>
<td>/samples/drconfig/Axis</td>
<td>Directory containing batch files and Axis source code for the VMware Site Recovery Manager Appliance Management API.</td>
</tr>
<tr>
<td>/samples/drconfig/DotNet/cs</td>
<td>Directory containing batch files, plus several other subdirectories containing Windows C# sample applications (in the appropriate namespace structure), the Microsoft Visual Studio project files (.sln files) for the VMware Site Recovery Manager Appliance Management API.</td>
</tr>
<tr>
<td>/samples/drconfig/JAXWS</td>
<td>Directory containing batch files and JAXWS source code for the VMware Site Recovery Manager Appliance Management API.</td>
</tr>
<tr>
<td>/wsdl/drconfig/drconfig.wsdl</td>
<td>The Web Services Description Language (WSDL) file containing definition of the VMware Site Recovery Manager Appliance Management API.</td>
</tr>
<tr>
<td>/wsdl/drconfig/drconfig-service.wsdl</td>
<td>A WSDL file defining the Web services endpoint at which the VMware Site Recovery Manager Appliance Management API is available. The drconfig-service.wsdl file references the drconfig.wsdl with an import statement, so you will use the appropriate generation tool with drconfig-service.wsdl (rather than drconfig.wsdl directly).</td>
</tr>
</tbody>
</table>

### Download and Setup

Setting up your environment to develop client applications with the SDK involves several steps, but if you are already developing vSphere applications, some of the steps are unnecessary.
Procedure

1. Select a programming language (C# or Java) for the Web services client application development. You can use Linux or Windows for the Java development. C# development is done on Windows.

2. Identify the target VMware Site Recovery Manager server (or servers) to use for development. A “target server” is a Site Recovery Manager server that your client application manages.

3. Install, or verify the presence of, the development environment appropriate for your programming language.

   - For C#, you need one of the Microsoft development environments, such as Visual Studio 2008 or Microsoft Visual C#. Use Microsoft Visual Studio 2008 or later, which includes the required .NET Framework. For more information, visit the MSDN website.

     **Note** For Site Recovery Manager Appliance Management APIs, use Microsoft Visual Studio 2017.

   - You can use Java Standard Edition (SE) 6.0 or 7.0. VMware recommends Java Development Kit (JDK) 1.7.0_45 or later. For more information, visit the Oracle Java website. Open JDK works also.

     **Note** For Site Recovery Manager Appliance Management APIs, use JDK 1.8 Update 202.

4. Obtain the appropriate Web services client tools (XML parser, WSDL-to-proxy-code generation tools, and runtime) for your programming language.

   - For C#, you need Microsoft .NET Framework 2.0 or 1.1. If you already use Microsoft development tools, it is likely you already have this. You can obtain the .NET Framework 2.0 from MSDN. You also need the .NET 2.0 Software Development Kit, which includes the WSDL-to-stub generation tool (wsd1.exe) and the command-line C# compiler (csc.exe), both of which get called from the gensrmstubs.cmd script. You can get the .NET 2.0 Software Development Kit from Microsoft: [http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=19988](http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=19988).

   - For Java with JAX-WS, you can use the JAX-WS framework that includes the JDK.

     **Note** For Site Recovery Manager APIs, use Java 1.4. For Site Recovery Manager Appliance Management APIs, use Java 1.8.

   - For Java with Axis, you need the Apache Axis 1.4 client-side Web service libraries. For documentation and downloads, visit the Axis Apache website.

     **Note** For Site Recovery Manager Appliance Management APIs, use Apache Axis2.
The Site Recovery Manager SDK package includes sample code demonstrating common use cases for programmatically configuring Site Recovery Manager Virtual Appliance. The sample code includes Java and C# source code files for building a simple command-line tool. This command-line tool can configure srm-va, retrieve the srm-va configuration, and clear the srm-va configuration.

This chapter includes the following topics:

- About C# .NET Samples
- About Java JAX-WS Samples
- About Java Axis Samples

About C# .NET Samples

This section describes how to build and run the sample code that uses C# .NET for the Site Recovery Manager Appliance Management API. The samples have been developed to work with the Microsoft Visual Studio 2017. They are located in subdirectories contained in the following SDK directory:

`/samples/srm/DotNet`

Build Sample Code with Visual Studio 2017

You can build C# .NET samples using Microsoft Visual Studio 2017.

Procedure

1. Open a Developer Command Prompt for Visual Studio 2017 from the Windows Start Menu.
2. Navigate to the `SDK\samples\drconfig\DotNet` subdirectory.
3. At the command prompt, type `BuildSamples.cmd` to run the build commands.

Run Sample Code from Visual Studio 2017
Procedure
2. Open the DrConfigSamples.sln file.
3. Change the Project Properties to specify the command-line arguments:
   a. From the Project menu, select Properties to display the Property Pages dialog box.
   b. In the Project_Name Property Pages dialog box, select Configuration Properties—Debugging in the left pane.
   c. In the right pane (under Start Options), select Command Line Arguments.
   d. To save the changes, click OK.
4. Run the sample at the command prompt.

About Java JAX-WS Samples
This section describes how to build and run the sample code that uses the JAX-WS bindings for the Site Recovery Manager Appliance Management API. The samples have been developed to work with the JAX-WS bundled with the JDK 1.8.0_202. They are located in subdirectories contained in the following SDK directory:

SDK/samples/drconfig/JAXWS/com/vmware

Build JAX-WS Sample Code
The build scripts (build.bat or build.sh) generate Site Recovery Manager Appliance Management API Java stubs from the Site Recovery Manager Appliance Management API WSDL, compile the generated stubs, and compile the sample programs.

Procedure
1. Set the JAVAHOME environment variable to the base directory of your installed JDK 1.8. For example,
   - On Linux, this can be /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64, or /, depending on your java location
   - On Windows, this can be C:\Program Files\Java\jdk1.8.0_202
2. Change directory to sdk/samples/drconfig/JAXWS and run the build.sh script (or on Windows, the build.bat file) to generate the Site Recovery Manager Appliance Management API Java stubs from the drconfig-Service.wsdl definitions, generate the Java stubs, and compile the sample Java code into class files, from which jars will be created (drconfig.jar, samples.jar and vim25.jar).
Note the WSDL file dependency: JAX-WS requires a WSDL file for the stub generation and compilation. To manage this dependency, the build script performs the following operations:

a. Calls the wsimport JDK tool to generate the Site Recovery Manager Appliance Management Java stubs from the Site Recovery Manager Appliance Management API WSDL file (drconfig-service.wsdl).

b. Specifies the wsimport -wsdlLocation command-line option to identify the WSDL file location.

c. Copies the WSDL file and related schema files into the drconfig.jar file.

The script compiles the Sample Java code, and imports the generated stubs. It uses the drconfig.jar built by the build.sh script. The WSDL file must be in the same location that was specified by the -wsdlLocation command-line option. To establish this location, the build script modifies the DrConfigService class to reference the WSDL location inside the JAR file. The Sample.jar will be modified to have the class path reference to the dependant JAR files.

**Run JAX-WS Sample Code**

After building, you can run the sample program that uses JAX-WS bindings for the Site Recovery Manager Appliance Management API. The program was developed to work with the JAX-WS framework that is bundled with the JDK 1.8.0_202

**Procedure**

1. Change directory to `sdk\samples\drconfig\JAXWS`, where the JAR files are located, if you are not already there.

2. Define VMKEYSTORE as the path to the Java key store. This can be the default Java keystore, or custom defined. This is needed to securely access Site Recovery Manager Server.

   ```
   export VMKEYSTORE=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/jre/lib/security/cacerts
   ```
   or your custom java keystore location

   ```
   set VMKEYSTORE=C:\Program Files\Java\jdk1.8.0_202\jre\lib\security\cacerts
   ```

   For more information about VMKEYSTORE, see [SSL Certificates](#).

3. Call the run script or batch file to execute the sample program. This sample program prints a usage summary if you do not specify any options or if you specify --help on the command line.

**Clean up JAX-WS Sample Code**

To explicitly delete the JAX-WS Sample Code, use either a script or a batch file.

**Procedure**

1. Change directory to `sdk\samples\drconfig\JAXWS`, if you are not already there.
Run the `clean.sh` script or the `clean.bat` batch file.

About Java Axis Samples

This section describes how to build and run the VMware sample code that uses the AXIS2 web services for the Site Recovery Manager Appliance Management API. The samples have been developed to work with the AXIS2 version axis2-1.7.9, as for the JAVA use JDK 1.8.0_202. The samples are located in the subdirectories contained in the SDK/samples/drconfig/Axis/java/com/vmware/samples directory.

Build JAVA AXIS Sample Code

The build scripts (build.bat or build.sh) generate Site Recovery Manager Appliance Management API Java stubs from the Site Recovery Manager Appliance Management API WSDL, compile the generated stubs, and compile the sample programs.

Procedure

1. Make sure that the Java development kit and Apache Axis2 are installed and functioning.
2. Start the Linux terminal (shell) or Windows command prompt.
3. Set the environment variables as shown in the Java and Axis environment variables table.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description and Usage Notes</th>
<th>Example Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXIS2_HOME</td>
<td>Complete path to the Apache Axis2 installation top-level directory. Must be set prior to using the build.bat script.</td>
<td><code>C:\apache\axis2-1.7.9</code></td>
</tr>
<tr>
<td>JAVAHOME</td>
<td>Path to the binary directory for the Java JDK.</td>
<td><code>C:\Program Files\Java\jdk1.8.0_202</code></td>
</tr>
</tbody>
</table>

4. Change directory to `sdk/samples/drconfig/Axis/java` and run the `build.sh` script (or on Windows, the `build.bat` file) to compile the sample Java code into a class files.

Run Java Axis Sample Code

Source code build files are located in the `samples\drconfig\Axis\java\com\vmware\samples` directory, as extracted from the ZIP archive.

Procedure

1. Change directory to `sdk/samples/drconfig/JAXWS`, where the JAR files are located.
2. When running the samples, if you are going to use your own certificates, then you must set the `VMKEYSTORE` environment variable to your Java keystore location. 

   ```
   export VMKEYSTORE=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.e17_7.x86_64/jre/lib/security/cacerts OR custom java keystore location and set VMKEYSTORE=C:\Program Files\Java\jdk1.8.0_202\jre\lib\security\cacert
   ```
3 After building the sample program, you can call it with the run script. For example, if you are building the ConfigureSrm sample, you can call it using the following command:

```
run.bat com.vmware.samples.drconfig.ConfigureAppliance configure --drconfigaddr https://FQDN_or_IP_Address:5480/configureserver/sdk --drconfiguser admin --pscuri pscfqdn:443 --pscuser Administrator@vsphere.local
```

**Clean up JAVA AXIS Sample Code**

To explicitly delete the JAVA AXIS Sample Code, use either a script or a batch file.

**Procedure**

1. Change directory to `SDK/samples/drconfig/Axis/java`, if you are not already there.
2. Run the clean.sh script or the clean.bat batch file.
SDK Samples for Site Recovery Manager API

The Site Recovery Manager SDK package includes sample code demonstrating common use cases for programmatically creating protection groups and initiating test, recovery, and reprotection operations.

This chapter includes the following topics:

- About the C# .NET Samples
- About Java JAX-WS Samples
- About Java Axis Samples

About the C# .NET Samples

Currently the SDK includes recovery plan sample code that you can build on .NET. You can build C# .NET samples using Microsoft Visual C# 2008 Express or Microsoft Visual Studio 2008.

If you have a later version of Visual Studio, it might ask to convert the solution and project files before proceeding. An earlier version of this SDK supported Visual Studio 2005. Visual Studio 2003 was never supported because of performance issues (.NET took a long time to instantiate the VimService class).

Build Sample Code with Visual Studio 2008

You can build C# .NET samples using Microsoft Visual C# 2008 Express or Microsoft Visual Studio 2008.

Procedure


2. Start the Windows command prompt.

   On 64-bit Windows systems, run C:\Windows\SysWOW64\cmd.exe so the sample programs execute under Windows 32-bit on Windows 64-bit (WOW64).

3. Navigate to the SDK\samples\DotNet sub-directory.

4. At the command prompt, type Build2008.cmd to execute the build commands.
Build Sample Code with Visual C# 2008 Express

You can build C# .NET samples using Microsoft Visual C# 2008 Express or Microsoft Visual Studio 2008. If you have a later version of Visual Studio, it might ask to convert the solution and project files before proceeding.

Procedure

1. Select the default (Full) Microsoft Visual C# 2008 Express installation.
2. If you installed Visual C# 2008 Express in the default location, skip this step. Otherwise:
   a. Create the System environment variable VSINSTALLDIR.
   b. Set the VSINSTALLDIR environment variable to the location of the Microsoft Visual Studio tools, in the Common7 sub-directory of the Microsoft Visual C# 2008 Express installation. Default locations are shown below. Use quotation marks around directory names that contain spaces, as these do.
      - “C:\Program Files\Microsoft Visual Studio 9.0\Common7"
      - “C:\apps\Microsoft Visual Studio 9.0\Common7”

      If Visual C# Express is installed in its default folder C:\Program Files\Microsoft Visual Studio 9.0, you do not need to create or set the VSINSTALLDIR environment variable.
4. Navigate to the SDK\samples\DotNet sub-directory.
5. At the command prompt, type Build2008.cmd to execute the build commands.

   The build process generates the RecoveryPlan sample program, which lists all recovery plans and optionally gets state for the specified recovery plan. A sample build can be executed from the \bin or \debug directory of a project. You can also run samples from within Visual Studio, at the .NET command prompt. To display help text for any application, you can run the application without any parameters.

Run Sample Code from Visual Studio

You can build C# .NET samples using Microsoft Visual C# 2008 Express or Microsoft Visual Studio 2008.

Procedure

3 Change the Project Properties to specify the command line arguments:
   a From the Project menu, select Properties to display the Property Pages dialog.
   b In the Project_Name Property Pages dialog, select Configuration Properties—Debugging on the left.
   c In the right-hand pane (under Start Options), select Command Line Arguments.
   d Click OK to save your changes.
4 Run the sample code at the command prompt.

Run C# Sample Code

You can build C# .NET samples using Microsoft Visual C# 2008 Express or Microsoft Visual Studio 2008.

Procedure

1 After you generate the sample program, you can run it as follows: RecoveryPlan --url <webserviceurl> --username <user> --password <passwd> --planname <plan>
   The RecoveryPlan program lists all recovery plans and optionally gets the state for the plan specified after the --planname option.
2 You can remove build files by running the clean.bat batch script.

About Java JAX-WS Samples

This section describes how to build and run the sample program that uses JAX-WS bindings for the Site Recovery Manager API.

The program was developed to work with the JAX-WS framework that is bundled with the JDK 1.6 and later Java source code is located in the samples/JAXWS/com/vmware/samples/recovery directory, as extracted from the ZIP archive.

Build JAX-WS Sample Code

The sample program that uses JAX-WS bindings for the Site Recovery Manager API was developed to work with the JAX-WS framework that is bundled with the JDK 1.6 and later.

Procedure

1 Set the JAVAHOME environment variable to the base directory of your installed JDK.
   On Linux this could be /usr/lib/jvm/java-7-openjdk-i386 for example.
   On Windows this could be C:\Program Files\Java\jdk1.7.0_65 for example.
2. Change directory to sdk/samples/JAXWS and run the build.sh script (or on Windows, the build.bat file) to generate the Site Recovery Manager API Java stubs from the srm-Service.wsdl definitions, generate the Java stubs, and compile the sample Java code into a class file.

Note the WSDL file dependency: JAX-WS requires a WSDL file for stub generation and compilation. To manage this dependency, the build script performs the following operations:

- It calls the wsimport JDK tool to generate Java stubs from the srm-Service.wsdl SRM WSDL file.
- It specifies the wsimport -wsdlLocation command line option to identify the WSDL file location.
- It copies the WSDL file and related schema files into the srm.jar file.

To compile Java code that imports the generated stubs and uses the srm.jar built by the build.sh script, the WSDL file must be in the same location that was specified by the -wsdlLocation command line option. To establish this location, the build script modifies the SrmService class to reference the WSDL location inside the JAR file. Then you just need to add the srm.jar file to your class path.

Run JAX-WS Sample Code

After building, you can run the sample program that uses JAX-WS bindings for the Site Recovery Manager API. The program was developed to work with the JAX-WS framework that is bundled with the JDK 1.6 and later.

Procedure

1. Change directory to sdk\samples\JAXWS, where the JAR files are located, if you are not already there, and set CLASSPATH. Sometimes %CLASSPATH% has already been set system wide. Example settings for Linux and Windows are export CLASSPATH=/usr/lib/jvm/java-7-openjdk-i386/lib and set CLASSPATH=%JAVAHOME%\lib\.

2. Define VMKEYSTORE as the path to the Java key store. This is needed to securely access Site Recovery Manager Server.

   export VMKEYSTORE=/usr/share/mime/application/x-java-keystore.xml
   set VMKEYSTORE=C:\cygwin\usr\share\mime\application\x-java-keystore.xml

   For more information about VMKEYSTORE, see SSL Certificates.

3. Call the run script or batch file. This sample program prints its usage summary, as if you specified --help on the command line, run.sh com.vmware.samples.recovery.RecoveryPlanList or run.bat com.vmware.samples.recovery.RecoveryPlanList.

4. As you can see from the usage message, the RecoveryPlanList sample code requires a user name and password for log in to the Site Recovery Manager administrator account. You need to pass in additional options: --username srmadmin --password secret --planname myRecoveryPlan
Clean Up JAX-WS Sample Code

You can clean up the JAX-WS Sample Code by using either a script or a batch file.

Procedure

1. Change directory to `sdk\samples\JAXWS`, if you are not already there.
2. Run the `clean.sh` script or the `clean.bat` batch file.

About Java Axis Samples

This section describes how to build and run the sample program that uses Axis Web services for the Site Recovery Manager API.

Axis can be downloaded from the Apache Web site, or as the `libaxis-java` package in some Linux distributions. Axis works with JDK 1.6 and later. Source code build files are located in the `samples/Axis/java` directory, as extracted from the ZIP archive.

Build Java Axis Sample Code

You can build the sample program that uses Axis Web services for the Site Recovery Manager API. Axis can be downloaded from the Apache Web site, or as the `libaxis-java` package in some Linux distributions. Axis works with JDK 1.6 and later. Source code build files are located in the `samples/Axis/java` directory, as extracted from the ZIP archive.

Procedure

1. Make sure the Java development kit and Apache Axis are installed and functioning.
2. Start the Linux terminal (shell) or Windows command prompt.
3. Set the environment variables as shown in the Java and Axis environment variables table.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description and Usage Notes</th>
<th>Example Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXISHOME</td>
<td>Complete path to the top-level Axis installation directory. Must be set before using the build scripts.</td>
<td>C:\Apache\axis1.4 /usr/share/java</td>
</tr>
<tr>
<td>JAVAHOME</td>
<td>Path to the binary directory for the Java JDK.</td>
<td>C:\Java \jdk1.7.0_65 /usr/lib/jvm/java-7-openjdk-i386</td>
</tr>
</tbody>
</table>

4. Change directory to `sdk/samples/Axis/java` and run the `build.sh` script (or on Windows, the `build.bat` file) to compile the sample Java code into a class file.
If the build script produces error messages about missing classes (could not find or load a class), edit the script and change the LOCALCLASSPATH line so path names refer to the proper jar file versions. Some Java archives contain symbolic links where a generic file points to a specific version of the jar file.

The script takes time to build the RecoveryPlanList sample program, which lists all recovery plans, or optionally gets state for a specified recovery plan.

**Note** The sample program was written for Axis version 1. It may require modifications for version 2.

### Run Java Axis Sample Code

Source code build files are located in the samples/Axis/java directory, as extracted from the ZIP archive.

**Procedure**

1. Change directory to sdk/samples/JAXWS, where the JAR files are located, if you are not already there, and set CLASSPATH. Example settings for Linux and Windows are

   ```
   export CLASSPATH=/usr/lib/jvm/java-7-openjdk-i386/lib and set CLASSPATH=%JAVAHOME%\lib
   ``

   Sometimes CLASSPATH has already been set system wide.

2. Define VMKEYSTORE as the path to the Java key store. This is needed to securely access a Site Recovery Manager Server, export VMKEYSTORE=/usr/share/mime/application/x-java-keystore.xml and set VMKEYSTORE=C:\cygwin\usr\share\mime\application\x-java-keystore.xml

3. After you build the sample program, you can call it with the run script as follows, run.sh

   ```
   com.vmware.samples.recovery.RecoveryPlanList --url <srm-URL> --username <user> --password <passwd>
   ```

   If you include the --ignorecert option, the sample code runs the following to get around an untrusted server certificate:

   ```
   System.setProperty("org.apache.axis.components.net.SecureSocketFactory",
   "org.apache.axis.components.net.SunFakeTrustSocketFactory");
   ```

### Clean Up Java Axis Sample Code

You can clean up the JAVA Axis sample code by either running a script or a batch file.

**Procedure**

1. Change directory to sdk/samples/Axis/java, if you are not already there.

2. Run the clean.sh script or the clean.bat batch file.
Logical Usage Order - Site Recovery Manager Appliance Management API

This chapter contains descriptions for Site Recovery Manager Appliance Management APIs.

The API descriptions in this chapter follow the logical usage order of List of API Operations. In examples below, MoRef indicates a String that references a managed object.

This chapter includes the following topics:

- Appliance Manager
- Configuration Manager
- Configuration Task
- Database Manager
- Diagnostic Manager
- Service Instance
- Service Manager
- SRA Manager
- SSL Certificate Manager
- Update Manager

Appliance Manager

This section describes the tools to manipulate the appliance of Site Recover Manager server. These operations are applicable on SRM, VRMS, and VRS deployments.

GetAllTimeZones

This method gets all available time zones. It returns a list representing all the available time zones.

Synopsis

```java
String[] getAllTimeZones();
```
 Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetCurrentDateTime**

This method gets the current date and time of the appliance. It returns a vmodl.DateTime object with the appliance date and time.

**Synopsis**

```java
DateTime getCurrentDateTime();
```

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetCurrentTimeZone**

This method gets the current time zone of the appliance. It returns a string representing the current time zone.

**Synopsis**

```java
String getCurrentTimeZone();
```

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetDiskInfo**

This method retrieves appliance disks information. It returns an array of DiskInfo objects, which contains of the appliance's disk information.

**Synopsis**

```java
DiskInfo[] getDiskInfo();
```

DiskInfo contains appliance disk info. It has the following fields:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Appliance hostname.</td>
</tr>
<tr>
<td>partitionName</td>
<td>Partition name. For example, root, swap, core, log, etc.</td>
</tr>
<tr>
<td>description</td>
<td>ALocalized description.</td>
</tr>
<tr>
<td>usedSize</td>
<td>Disk used size in bytes</td>
</tr>
<tr>
<td>totalSize</td>
<td>Total disk size in bytes</td>
</tr>
</tbody>
</table>

### Faults

- **RuntimeFault**

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### GetInfo

This method retrieves appliance information. It returns an `ApplianceInfo` object which contains information about the appliance.

#### Synopsis

```java
ApplianceInfo getInfo();
```

`ApplianceInfo` contains information about the appliance. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>Appliance hostname</td>
</tr>
<tr>
<td>srmBuild</td>
<td>Appliance product (SRM) build number</td>
</tr>
<tr>
<td>srmProduct</td>
<td>Appliance product name</td>
</tr>
<tr>
<td>srmVersion</td>
<td>Appliance product (SRM) version</td>
</tr>
<tr>
<td>vaFullVersion</td>
<td>Virtual appliance full version - version + build number</td>
</tr>
<tr>
<td>vaBuildNumber</td>
<td>Virtual appliance build number</td>
</tr>
<tr>
<td>systemInfo</td>
<td>System information</td>
</tr>
</tbody>
</table>

### Faults

- **RuntimeFault**

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### GetNetworkInfo

This method retrieves appliance network information. It returns a `NetworkInfo` object which contains network information about the appliance.
Synopsis

```java
NetworkInfo getNetworkInfo();
```

NetworkInfo contains network information of the appliance. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DnsInfo dns</td>
<td>DNS information about the appliance.</td>
</tr>
<tr>
<td>InterfaceInfo[] interfaces</td>
<td>Class that contains network interface information about the appliance.</td>
</tr>
</tbody>
</table>

DnsInfo contains information about the DNS configuration of the appliance. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DnsMode</td>
<td>Enum that describes the source of DNS servers. It enumerates the following values:</td>
</tr>
<tr>
<td></td>
<td>- DHCP: DNS servers addresses that are obtained from a DHCP server.</td>
</tr>
<tr>
<td></td>
<td>- STATIC: The DNS servers addresses are specified explicitly.</td>
</tr>
<tr>
<td>dnsMode</td>
<td>DNS mode. The value should be one of DnsMode enum.</td>
</tr>
<tr>
<td>hostname</td>
<td>Hostname</td>
</tr>
<tr>
<td>servers</td>
<td>Servers. This value is ignored while in DHCP mode.</td>
</tr>
</tbody>
</table>

InterfaceInfo contains network interface information about the appliance. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Interface name. For example, &quot;nic0&quot;, &quot;nic1&quot;.</td>
</tr>
<tr>
<td>interfaceStatus</td>
<td>Interface status. The value should be one of the InterfaceStatus Enums.</td>
</tr>
<tr>
<td>mac</td>
<td>MAC address. For example, 00:0C:29:94:BB:5A.</td>
</tr>
<tr>
<td>IPv4Info ipv4</td>
<td>IPv4 Address information.</td>
</tr>
<tr>
<td>IPv6Info ipv6</td>
<td>IPv6 Address information.</td>
</tr>
</tbody>
</table>

IPv4Info defines the IPv4 configuration state of a network interface. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interfaceName</td>
<td>Interface name. For example, &quot;nic0&quot;, &quot;nic1&quot;.</td>
</tr>
<tr>
<td>mode</td>
<td>Address assignment mode. Value should be one of the IPv4Mode enums.</td>
</tr>
<tr>
<td>address</td>
<td>IPv4 address, for example, &quot;10.20.80.191&quot;. Value not needed when DHCP mode.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>prefix</td>
<td>IPv4 CIDR prefix, for example, 24. This value is not required while in DHCP mode.</td>
</tr>
<tr>
<td></td>
<td>For more information, see <a href="http://www.oav.net/mirrors/cidr.html">http://www.oav.net/mirrors/cidr.html</a> for netmask-to-prefix conversion.</td>
</tr>
<tr>
<td>defaultGateway</td>
<td>IPv4 address of the default gateway. This Value is not required while in DHCP mode.</td>
</tr>
</tbody>
</table>

IPv4Mode defines different IPv4 modes. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td>IPv4 address is automatically assigned by a DHCP server.</td>
</tr>
<tr>
<td>STATICMODE</td>
<td>IPv4 address is static.</td>
</tr>
<tr>
<td>UNCONFIGURED</td>
<td>The IPv4 protocol is not configured.</td>
</tr>
</tbody>
</table>

IPv6Info contains IPv6 info on a particular interface. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interfaceName</td>
<td>Network interface. For example, &quot;nic0&quot; to configure.</td>
</tr>
<tr>
<td>dhcp</td>
<td>Address assigned by a DHCP server. This option can be set to true in parallel with autoconf and static IPv6 addresses.</td>
</tr>
<tr>
<td>autoconf</td>
<td>Address is assigned by Stateless Address Autoconfiguration (SLAAC). This option can be set to true in parallel with dhcp and static IPv6 addresses.</td>
</tr>
<tr>
<td>IPv6Address[] addresses</td>
<td>A list of addresses to be statically assigned. Values can be available in parallel with set dhcp and autoconf.</td>
</tr>
<tr>
<td>defaultGateway</td>
<td>Default gateway for static IP address assignment. This configures the global IPv6 default gateway on the appliance with the specified gateway address and interface. This gateway replaces the existing default gateway configured on the appliance. However, if the gateway address is link-local, then it is added for that interface. This does not support configuration of multiple global default gateways through different interfaces.</td>
</tr>
</tbody>
</table>

IPv6Address is used for naming an IPv6 address. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>IPv6 address, for example, fc00:10:20:83:20c:29ff:fe94:bb5a.</td>
</tr>
<tr>
<td>prefix</td>
<td>IPv6 CIDR prefix, for example, 64.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetTimeSyncConfig**

This method gets the appliance time sync mode. It returns a TimeSyncInfo object representing the timeSyncMode.

**Synopsis**

```java
TimeSyncInfo getTimeSyncConfig();
```

TimeSyncInfo has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeSyncMode</td>
<td>The TimeSyncMode defines time synchronization modes. It enumerates the following:</td>
</tr>
<tr>
<td></td>
<td>- DISABLED: Time synchronization is disabled.</td>
</tr>
<tr>
<td></td>
<td>- NTP: NTP-based time synchronization.</td>
</tr>
<tr>
<td></td>
<td>- HOST: VMware Tool-based time synchronization.</td>
</tr>
<tr>
<td>timeSyncMode</td>
<td>The Time Synchronization Mode. Must be one of the enum values.</td>
</tr>
<tr>
<td>ntpServers</td>
<td>NTP servers. Used only when timeSyncMode is NTP.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**Restart**

This method restarts the appliance.

**Synopsis**

```java
void restart();
```

**Faults**

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**SetCurrentTimeZone**

This method sets the appliance time zone.
Synopsis

```java
void setCurrentTimeZone(String timeZone)
```

timeZone parameter represents the time zone.

**Faults**
- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**SetNetworkInfo**

This method sets the appliance network information.

Synopsis

```java
void setNetworkInfo(NetworkInfo networkInfo)
```

networkInfo parameter is a NetworkInfo object that contains network information of the appliance.

NetworkInfo contains network information of the appliance. For more information see GetNetworkInfo.

**Faults**
- InvalidArgument
- InvalidNetworkConfiguration
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**SetTimeSync**

This method sets appliance time sync information.

Synopsis

```java
void setTimeSync(TimeSyncInfo timeSyncInfo)
```

timeSyncInfo is a TimeSyncInfo that contains the time sync information of the appliance. For more information, see GetTimeSyncConfig.
Faults

- HostUnreachableFault
- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

Stop

This method stops the appliance.

Synopsis

```java
void stop();
```

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

Configuration Manager

This section describes the tools to configure the SRM server.

GetRunningTask

This method gets the currently active configuration task or null.

Synopsis

```java
ConfigurationTask getRunningTask();
```

ConfigurationTask is a managed object that provides operations to configure the SRM server. For more information, see Configuration Task.

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.
CheckRegistration

This method checks whether the given extension key is already registered in SSO, lookup service, and as a vCenter extension. Returns a list of existing registrations for the requested extension key.

Synopsis

```java
Registration[] checkRegistration(String adminUser, @secret String adminPassword, ConnectionSpec connection, String extensionKey)
```

CheckRegistration has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adminUser</td>
<td>Name of a user with sufficient privileges to perform checks.</td>
</tr>
<tr>
<td>adminPassword</td>
<td>Password for the administration user.</td>
</tr>
<tr>
<td>connection</td>
<td>The connection specification. For more information about connection ConnectionSpec see ClearSrmConfiguration.</td>
</tr>
<tr>
<td>extensionKey</td>
<td>extensionKey for which checks should be performed. This value is returned in the result: Registration.extensionKey</td>
</tr>
</tbody>
</table>

Registration contains information about existing LS++, SSO, or vCenter registration. It has the extensionKey field which is the requested extension key.

Faults

- InvalidLogin
- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

ClearSrmConfiguration

This method clears the SRM server configuration with the vSphere infrastructure. Returns a task object that can be used to monitor the operation.

Synopsis

```java
ConfigurationTask clear(ConfigurationSpec config)
```

config parameter is a ConfigurationSpec data object that contains SRM configuration specification.

ConfigurationSpec has the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>siteName</td>
<td>The SRM site name. If not set the site name is read from the current SRM configuration if it exists.</td>
</tr>
<tr>
<td>hostName</td>
<td>The SRM server FQDN. Used when registering with infrastructure and management nodes. If not set, the DNS name will be used.</td>
</tr>
<tr>
<td>extensionKey</td>
<td>The SRM extension key. If not set the default extension key value of &quot;com.vmware.vcDr&quot; will be used.</td>
</tr>
<tr>
<td>clockToleranceSeconds</td>
<td>The allowed server clock tolerance in seconds. If not set the default value of 3 seconds will be used. This parameter is used only when validating the VC server where SRM will be registered. Clock difference between SRM virtual appliance and vCenter Server should not exceed this value, otherwise the validation (or configuration) fails.</td>
</tr>
<tr>
<td>ConnectionSpec</td>
<td>The connection specification. If not set the connection parameters will be read from the current SRM configuration if it exists.</td>
</tr>
<tr>
<td>adminUser</td>
<td>The name of a user with sufficient privileges to perform configuration tasks on the infrastructure and management nodes as well as SSO service configuration tasks on the infrastructure node.</td>
</tr>
<tr>
<td>adminPassword</td>
<td>Password for the administrator user.</td>
</tr>
<tr>
<td>deleteSrmData</td>
<td>Delete SRM data. Used only when clearing SRM configuration. If set to true, existing SRM database will be deleted.</td>
</tr>
<tr>
<td>extraConfig</td>
<td>Additional configuration settings in XML format. These settings are used to upgrade database.</td>
</tr>
<tr>
<td>localSrmUuid</td>
<td>UUID of the local SRM Server. This is out parameter returned by ConfigurationManager.readCurrentConfig()</td>
</tr>
<tr>
<td>organization</td>
<td>Organization name.</td>
</tr>
<tr>
<td>description</td>
<td>Plugin description.</td>
</tr>
<tr>
<td>adminEmail</td>
<td>Admin email.</td>
</tr>
<tr>
<td>moId</td>
<td>Managed Object ID of this VM.</td>
</tr>
</tbody>
</table>

**ConnectionSpec** is a structure that contains connection information for a service. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uri</td>
<td>The PSC node URI. FQDN + optional port. If port not specified 443 will be used.</td>
</tr>
<tr>
<td>thumbprint</td>
<td>Thumbprint of the PSC node's certificate. When the correct value is provided all security checks of the certificate are off.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vcInstanceId</td>
<td>Identifier of the MGMT node to register with. If not specified the configuration service will assume embedded environment is used and will look for MGMT node services at the PSC node address.</td>
</tr>
<tr>
<td>vcThumbprint</td>
<td>Thumbprint of the MGMT node's certificate.</td>
</tr>
</tbody>
</table>

`ConfigurationTask` is a managed object that provides operations to configure the SRM server. For more information, see `Configuration Task`.

### Faults

- `InvalidArgument`
- `RuntimeFault`
- `ServiceBusy`
- `SrmAlreadyRunning`

For more information about the faults, see `Faults in Site Recovery Manager Appliance Management API`.

### ConfigureSrm

This method configures the SRM server and connects it to the vSphere infrastructure.

#### Synopsis

```
ConfigurationTask configure(ConfigurationSpec config)
```

`config` parameter is a `ConfigurationSpec` data object that contains SRM configuration specification. For more information, see `ClearSrmConfiguration`.

`ConfigurationTask` is a managed object that provides operations to configure the SRM server. For more information, see `Configuration Task`.

### Faults

- `InvalidArgument`
- `RuntimeFault`
- `ServiceBusy`
- `SrmAlreadyRunning`

For more information about the faults, see `Faults in Site Recovery Manager Appliance Management API`.
ConfigureSyslogForwarding

This method sets syslog log forwarding. When `enable` is set to true, this method adds a rule to rsyslog configuration for given apps and restarts rsyslog service. With this rule, all the logs written by these apps to syslog are forwarded to the specified server. Note that the rule will be either appended or updated in the syslog configuration file and this does not remove any existing rules for other apps. When `enable` is set to false, the existing rules for given apps added with this method are deleted.

Synopsis

```java
void configureSyslogForwarding(boolean enable, @optional String[] appNames, SyslogForwardInfo info)
```

ConfigureSyslogForwarding has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>True to enable log forwarding, false to disable.</td>
</tr>
<tr>
<td>appNames</td>
<td>List of application names for which log forwarding should be configured.</td>
</tr>
<tr>
<td></td>
<td>Values must be one of the enums AppName. This value is not used since @version3.</td>
</tr>
<tr>
<td>info</td>
<td>Structure that holds log forwarding information.</td>
</tr>
</tbody>
</table>

SyslogForwardInfo contains information for syslog log forwarding. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Enumerates the list of available protocols to use. The available constants are tcp, udp, and relp.</td>
</tr>
<tr>
<td>host</td>
<td>IP address or FQDN of the syslog server to which the logs will be forwarded.</td>
</tr>
<tr>
<td>port</td>
<td>Port of the syslog server to which the logs will be forwarded.</td>
</tr>
<tr>
<td>protocol</td>
<td>Protocol to use for log forwarding. The value must be one of the Protocol enums.</td>
</tr>
</tbody>
</table>

Faults

- `InvalidArgument`
- `RuntimeFault`
- `SystemError`

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.
ConfigureSyslogServers

This method sets the syslog log forwarding. It configures all the SRM applications to forward to the specified servers. Receiving null deletes all the configurations. The new spec completely replaces the old configured servers.

Synopsis

```java
void configureSyslogServers(@optional SyslogForwardInfo[] info)
```

info parameter is a list of SyslogForwardInfo objects.

SyslogForwardInfo object contains information for syslog log forwarding. For more information, see ConfigureSyslogForwarding.

Faults

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

EnableSyslogLogging

This method enables or disables logging to syslog. When enabling syslog logging for SRM, this method call must be preceded by configure call which will generate the SRM configuration file.

Synopsis

```java
void enableSyslogLogging(boolean enable, @optional String[] appNames)
```

EnableSyslogLogging has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>True to enable logging to syslog and false to disable.</td>
</tr>
<tr>
<td>appNames</td>
<td>List of application names for which logging to syslog should be set. Values must be one of the enumsAppName. App must be stopped before calling this method. This value not used since @version3.</td>
</tr>
</tbody>
</table>

Faults

- InvalidArgument
- RuntimeFault
- SrmAlreadyRunning
- SrmNotConfigured
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetSyslogServers**

This method gets the syslog log forwarding information. Returns fwdInfo which is a list of SyslogForwardInfo objects that contain information about all the syslog servers.

**Synopsis**

```java
SyslogForwardInfo[] getSyslogServers()
```

SyslogForwardInfo contains information for syslog log forwarding. For more information, see ConfigureSyslogForwarding.

**Faults**

- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**IsReconfigureRequired**

This method checks if the reconfigure operation is required after an upgrade. Returns the boolean True if the reconfiguration is required. Reconfiguration is required after an upgrade when the Site Recovery Manager has been configured prior to the upgrade operation. This flag is not set after the upgrade if the Site Recovery Manager was never configured.

**Synopsis**

```java
boolean isReconfigureRequired();
```

**Faults**

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**ListVcServices**

This method lists all the vCenters in the Platform Service Controller (PSC).

**Synopsis**

```java
VcInfo[] listVcServices(String pscUri, @optional String pscThumbprint, @optional String serviceId)
```

listVcServices has the following parameters:
### VcInfo contains vCenter information. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>vCenter URL</td>
</tr>
<tr>
<td>serviceId</td>
<td>Service ID</td>
</tr>
<tr>
<td>nodeId</td>
<td>Node ID</td>
</tr>
<tr>
<td>productVersion</td>
<td>Product version.</td>
</tr>
</tbody>
</table>

### Faults

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### ReadCurrentConfig

This method reads the specification for the currently configured SRM server. Returns null if the SRM is not yet configured. If a configuration task is currently in progress this method returns its config parameter.

**Synopsis**

```java
ConfigurationSpec readCurrentConfig();
```

ConfigurationSpec is a data object that contains SRM configuration specification. For more information, see [ClearSrmConfiguration](#).

### Faults

- DatabaseConnectionFault
- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).
SendSyslogTestMessage

This method sends a test message to all the configured syslog servers. It sends a specified message to the syslog servers. The operation to send a message cannot get the result for the connection, so the client must manually verify that the syslog server logs contain the specified string. The test message is hardcoded.

Synopsis

```java
void sendSyslogTestMessage(String message)
```

message parameter is the content of the test message.

Faults

- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

ValidateConnection

This method validates connections to the vSphere infrastructure. If there is no explicit and valid vcInstanceId in the connection spec, no vCenter validation is done. Only the infranode is validated.

Synopsis

```java
void validateConnection(String adminUser, @secret String adminPassword, ConnectionSpec connection)
```

ValidateConnection has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adminUser</td>
<td>The name of a user with sufficient privileges to perform configuration tasks on the infrastructure and management nodes and SSO service configuration tasks on the infrastructure node.</td>
</tr>
<tr>
<td>adminPassword</td>
<td>Password for the administration user.</td>
</tr>
<tr>
<td>connection</td>
<td>The connection specification. For more information about ConnectionSpec, see ClearSrmConfiguration.</td>
</tr>
</tbody>
</table>

Faults

- InvalidArgument
- InvalidLogin
- RuntimeFault
- SsoTokenNotAcquired
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**SetHbrSrvNic**

Sets the HBR filter and management addresses.

**Synopsis**

```c
void setHbrSrvNic(HbrFilterInfo filterInfo)
```

setHbrSrvNic has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filterInfo</td>
<td>A HbrFilterInfo data object representing the filter and management addresses.</td>
</tr>
</tbody>
</table>

HbrFilterInfo data object type describes the HBR filter and management traffic. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filterIp</td>
<td>Filter IP address. Can be empty.</td>
</tr>
<tr>
<td>managementIp</td>
<td>Management IP address. Can be empty.</td>
</tr>
</tbody>
</table>

**Faults**

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**GetHbrSrvNic**

Gets the HBR filter and management IP addresses.

**Synopsis**

```c
HbrFilterInfo getHbrSrvNic()
```

getHbrSrvNic returns HbrFilterInfo data object representing the filter and management addresses. For more information, see SetHbrSrvNic.
Faults

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

Configuration Task

This section describes the operations to configure the SRM server. These operations are applicable according to the API that return this data object, generally SRM and VRMS.

GetTaskInfo

This method gets the current configuration task status.

Synopsis

```java
ConfigurationTaskInfo getTaskInfo();
```

ConfigurationTaskInfo provides status information for a configuration task. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskType</td>
<td>Enumerates the list of following task types:</td>
</tr>
<tr>
<td></td>
<td>- clear</td>
</tr>
<tr>
<td></td>
<td>- configuration</td>
</tr>
<tr>
<td></td>
<td>- installUpdate</td>
</tr>
<tr>
<td></td>
<td>- retrieveUpdate</td>
</tr>
<tr>
<td></td>
<td>- generateBundle</td>
</tr>
<tr>
<td></td>
<td>- getSraImages</td>
</tr>
<tr>
<td>complete</td>
<td>Flag to indicate whether the task has completed.</td>
</tr>
<tr>
<td>progress</td>
<td>Overall configuration progress.</td>
</tr>
<tr>
<td>currentOp</td>
<td>Current operation name.</td>
</tr>
<tr>
<td>currentOpProgress</td>
<td>Current operation progress.</td>
</tr>
<tr>
<td>.startTime</td>
<td>Time stamp when the task started running.</td>
</tr>
<tr>
<td>completeTime</td>
<td>Time stamp when the task was completed.</td>
</tr>
<tr>
<td>cancelled</td>
<td>Flag to indicate whether the client requested cancellation of the task.</td>
</tr>
<tr>
<td>error</td>
<td>If the configuration state is 'failed', then this property contains the fault code.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>The result of the task.</td>
</tr>
<tr>
<td>type</td>
<td>Type of the task. Can be one of the TaskType enum values.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**CancelSrmConfiguration**

This method cancels a running configuration task. Multiple cancel requests are treated as a single cancellation request. Cancelling a completed or an already cancelled task throws ServiceIdle exception. If successfully canceled, the StateInfo.error property is set to RequestCanceled. A cancel operation is asynchronous. The operation may return before the task is cancelled. After invoking it, the configuration task continues to remain working, and it can still succeed. In such a situation, the StateInfo.cancelled property is set.

**Synopsis**

```java
void cancel()
```

**Faults**

- RuntimeFault
- ServiceIdle

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**Database Manager**

This section describes the operations to configure SRM database. These operations are applicable only on SRM deployments.

**ReadStatus**

This method checks the database status and returns the version information.

**Synopsis**

```java
DatabaseStatus readStatus()
```
DatabaseStatus provides the database status information. It contains currently installed database version and the database version of the currently running service. It also contains the currently used ODBC driver version and name. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drVersion</td>
<td>Database version. This can be unset if the database is empty.</td>
</tr>
<tr>
<td>runtimeVersion</td>
<td>DB manager runtime version.</td>
</tr>
<tr>
<td>upgradeRequired</td>
<td>Flag set if the versions do not match and upgrade is required.</td>
</tr>
<tr>
<td>driverName</td>
<td>ODBC driver name.</td>
</tr>
<tr>
<td>driverVersion</td>
<td>ODBC driver version.</td>
</tr>
<tr>
<td>licenseAssetId</td>
<td>License asset identifier. Unset if the SRM was not started.</td>
</tr>
</tbody>
</table>

**Faults**

- ReadDbStatusFault
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

**ChangePassword**

This method changes the embedded database password.

**Synopsis**

```java
void changePassword(@secret String oldPassword, @secret String newPassword)
```

ChangePassword has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oldPassword</td>
<td>Old password for the database.</td>
</tr>
<tr>
<td>newPassword</td>
<td>New password for the database.</td>
</tr>
</tbody>
</table>

**Faults**

- ChangePasswordFault
- RuntimeFault
- SrmAlreadyRunning

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.
Diagnostic Manager

This section describes the interface to get the Site Recovery Manager system log bundles that contains log files, cores, and configuration files that are useful for the diagnosis of issues. These operations are applicable on SRM, VRMS, and VRS deployments.

GetRunningTask

This method gets the currently active diagnostic task or null.

Synopsis

```java
ConfigurationTask getRunningTask();
```

ConfigurationTask is a managed object that provides operations to configure the SRM server. For more information, see Configuration Task.

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

GenerateSystemLogBundle

This method instructs the server to generate a system log bundle. A log bundle includes log files and other configuration information that can be used to investigate failures. The bundle can be retrieved with calls to `retrieveSystemLogBundle(String, long, long)` method.

The Site Recovery Manager maintains five outstanding bundles. This method returns Configuration Task object that can be used to monitor the operation. `ConfigurationTask.getTaskInfo` method returns the current task status.

Synopsis

```java
ConfigurationTask generateSystemLogBundle()
```

`ConfigurationTask.getTaskInfo` returns `SystemLogBundleInfo`. It has the following fields:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The unique key for this bundle.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the file generated by the Diagnostics Manager. The value can be either &quot;zip&quot;, &quot;gz&quot;, or &quot;txt&quot;. This value is used by the client to construct a file with the proper extension.</td>
</tr>
<tr>
<td>timeCreated</td>
<td>The date and time when the bundle was generated. This is the local server time in UTC.</td>
</tr>
<tr>
<td>size</td>
<td>The size in bytes of the diagnostic bundle file.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 checksum for the bundle file. Can be used to verify its contents after page assembly using <code>retrieveSystemLogBundle(String, long, long)</code>.</td>
</tr>
<tr>
<td>url</td>
<td>The URL to download the bundle.</td>
</tr>
</tbody>
</table>

**Faults**
- RuntimeFault
- ServiceBusy

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**RetrieveSystemLogBundle**

This method retrieves the log bundle using the Binary data type. The bundle is downloaded in chunks which the client is supposed to assemble into a single file. The MD5 checksum in `SystemLogBundleInfo` object can be used to verify that the retrieved file is correctly assembled. It returns a binary object containing the requested chunk of the bundle file. Its size may be less than the `maxPageSize`.

**Synopsis**

```java
Binary retrieveSystemLogBundle(String key, long offset, long maxPageSize)
```

`RetrieveSystemLogBundle` has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The <code>SystemLogBundleInfo.key</code> returned by a call to <code>generateSystemLogBundle()</code> method.</td>
</tr>
<tr>
<td>offset</td>
<td>Byte offset into the bundle file to the start of the chunk that is to be returned.</td>
</tr>
<tr>
<td>maxPageSize</td>
<td>Maximum size in bytes of the chunk to be returned. This parameter can be used to control the size and number of chunks while retrieving a bundle. The maximum page size can also be limited internally by the server so the size of the returned chunk may be smaller than <code>maxPageSize</code> even if the bundle file contains more bytes.</td>
</tr>
</tbody>
</table>

**Faults**
- OutOfBounds
- RuntimeFault
- SystemLogBundleNotFound
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**DeleteSystemLogBundle**

This method instructs the server that a specific log bundle is no longer needed by the client that generated it. The server can then safely delete the file.

Calling this method after each call to `GenerateSystemLogBundle` method helps the server work more efficiently.

**Synopsis**

```java
void deleteSystemLogBundle(String key)
```

`DeleteSystemLogBundle` has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| key       | SystemLogBundleInfo.key returned by a call to `generateSystemLogBundle()` method. For more information, see `GenerateSystemLogBundle`.

**Faults**

- RuntimeFault
- SystemLogBundleNotFound

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**Service Instance**

The ServiceInstance managed object is the singleton object which provides access to the functionality of the Site Recovery Manager Appliance Management Server. Authorization and authentication support are limited to local OS users and method only privileges.

Do not specify the parameter privileges as they do not work. The authorization code currently recognizes only the following privileges:

- **System.Anonymous**: No authorization required. It can be called with an unauthenticated session.
- **System.View or System.Read**: Authorization is required. Call `Login` before invoking a method that requires this privilege.
- **VcDr.Internal.com.vmware.vcDr.InternalAccess**: Administrator access is required to invoke a method annotated with this privilege.

**Note** If a method requires a privilege not listed in the list above, then it is treated as if the administrator access privilege is required.
RetrieveContent

This method retrieves the properties of the service instance. It returns the properties belonging to the service instance, including various object managers.

Synopsis

```java
ServiceInstanceContent retrieveContent();
```

`ServiceInstanceContent` is the properties of the `ServiceInstance` class. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>about</td>
<td>Information about this service.</td>
</tr>
<tr>
<td>applianceManager</td>
<td>ApplianceManager instance. For more information, see Appliance Manager.</td>
</tr>
<tr>
<td>cfgManager</td>
<td>ConfigurationManager instance. For more information, see Configuration Manager.</td>
</tr>
<tr>
<td>dbManager</td>
<td>DatabaseManager instance. For more information, see Database Manager.</td>
</tr>
<tr>
<td>diagnosticManager</td>
<td>DiagnosticManager instance. For more information, see Diagnostic Manager.</td>
</tr>
<tr>
<td>serviceManager</td>
<td>ServiceManager instance. For more information, see Service Manager.</td>
</tr>
<tr>
<td>sslCertificateManager</td>
<td>SslCertificateManager API. For more information, see SSL Certificate Manager.</td>
</tr>
<tr>
<td>updateManager</td>
<td>UpdateManager API. For more information, see Update Manager.</td>
</tr>
<tr>
<td>sraManager</td>
<td>sraManager instance. For more information, see SRA Manager.</td>
</tr>
<tr>
<td>propertyCollector</td>
<td>Property Collector for external API</td>
</tr>
</tbody>
</table>

AboutInfo is a data object type that describes system information including the name, type, version, and build number. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Short form of the product name.</td>
</tr>
<tr>
<td>fullName</td>
<td>Complete product name, including the version information.</td>
</tr>
<tr>
<td>vendor</td>
<td>Name of the vendor of this product.</td>
</tr>
<tr>
<td>version</td>
<td>Dot-separated version string. For example, &quot;1.2&quot;.</td>
</tr>
<tr>
<td>build</td>
<td>Build string for the server on which this call is made. For example, x.y.z-num. This string does not apply to the API.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating System type and architecture.</td>
</tr>
<tr>
<td>productLineId</td>
<td>Product ID. Unique identifier of the product line.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiType</td>
<td>Indicates whether or not the service instance represents a standalone host.</td>
</tr>
<tr>
<td>apiVersion</td>
<td>The version of the API as a dot-separated string. For example, &quot;1.0.0&quot;.</td>
</tr>
<tr>
<td>instanceUuid</td>
<td>A globally unique identifier associated with this service instance.</td>
</tr>
<tr>
<td>deployment</td>
<td>Deployment type of the appliance. It enumerates the following:</td>
</tr>
<tr>
<td></td>
<td>- SRM</td>
</tr>
<tr>
<td></td>
<td>- VRMS</td>
</tr>
<tr>
<td></td>
<td>- VRS</td>
</tr>
<tr>
<td></td>
<td>- Combined</td>
</tr>
</tbody>
</table>

### Faults
- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### LoginDrConfig

This method logs you on to the server by verifying user and password with the local Operating System.

### Synopsis

```java
void login(String userName, @secret String password, @optional String locale)
```

login has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>ID of the user who is logging on to the server.</td>
</tr>
<tr>
<td>password</td>
<td>Password of the user logging on to the server.</td>
</tr>
<tr>
<td>locale</td>
<td>Locale for this session.</td>
</tr>
</tbody>
</table>

### Faults
- InvalidArgument
- InvalidLocale
- InvalidLogin
- NoPermission
- RuntimeFault
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

LogoutDrConfig
This method logs out and terminates the current session.

Synopsis

```java
void logout();
```

Faults
- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

ChangeUserPassword
This method assigns password to the user who is running the configuration service.

Synopsis

```java
void ChangePassword(String userName, @secret String currentPassword, @secret String newPassword)
```

ChangePassword has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>User for whom the password should be changed. Currently, this parameter is not taken into account and the password is changed only for the SRM user.</td>
</tr>
<tr>
<td>currentPassword</td>
<td>Current user password.</td>
</tr>
<tr>
<td>newPassword</td>
<td>New user password.</td>
</tr>
</tbody>
</table>

Faults
- ChangePasswordFault
- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

Service Manager

This section describes the operations to control appliance services. These operations are applicable on SRM, VRMS, and VRS deployments.

**Note** IsSrmServerRunning can be executed on VRMS deployment but it will return false.
IsSrmServerRunning

This method returns the current service state of the Site Recovery Manager.

Synopsis

```java
boolean isSrmServerRunning();
```

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

DrConfigStartService

This method starts the service. It does nothing if the service is already running.

Synopsis

```java
void Start(String service)
```

service is the parameter for the service to start. The value must be one of the following Service enums:

- srm
- db
- rsyslog
- sshd
- drclient
- envoy_proxy
- hbrsrv
- hmsdb
- hms

Faults

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.
DrConfigStopService

This method stops the service. It does nothing if the service is not running.

Synopsis

```java
void Stop(String service)
```

`service` is the parameter for the service to be stopped. Value must be one of the `service` enums. For more information, see DrConfigStartService.

Faults

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

DrConfigServiceStatus

This method returns status information of the service.

Synopsis

```java
ServiceStatusInfo getServiceStatus(String service)
```

`ServiceStatusInfo` is the information for service. It has the following fields:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartUpType</td>
<td>List of startup types. Enumerates the following:</td>
</tr>
<tr>
<td></td>
<td>Automatic</td>
</tr>
<tr>
<td></td>
<td>Manual</td>
</tr>
<tr>
<td>serviceId</td>
<td>The service id. This should correspond to the <code>ServiceManager::Service enum.</code></td>
</tr>
<tr>
<td>name</td>
<td>Name of the service.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the service.</td>
</tr>
<tr>
<td>startupType</td>
<td>Startup type. Value must be one of the enums.</td>
</tr>
<tr>
<td>isRunning</td>
<td>Boolean if the service is running.</td>
</tr>
</tbody>
</table>

Faults

- InvalidArgument
- RuntimeFault
- SystemError
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**DrConfigRestartService**

This method stops the service and then starts it. If the service is not running, it will be started.

**Synopsis**

```java
void Restart(String service)
```

`service` is the parameter for service that must be restarted. Value must be one of the `service` enums. For more information, see DrConfigStartService.

**Faults**

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**DrConfigAllServicesStatus**

This method returns `ServiceStatusInfo` object for all the services.

**Synopsis**

```java
ServiceStatusInfo[] getAllServicesStatus()
```

`ServiceStatusInfo` is the information about the service. For more information about `ServiceStatusInfo`, see DrConfigServiceStatus.

**Faults**

- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**SRA Manager**

This section describes the interface for managing SRA images and containers in the SRM Configuration Service. These operations are applicable only on SRM deployments.
GetRunningTask

This method gets the currently active SRA task or null.

Synopsis

```java
ConfigurationTask getRunningTask();
```

ConfigurationTask is a managed object that provides operations to configure the SRM server. For more information, see Configuration Task.

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

GetSraImages

This method returns a collection of SRA images loaded into the docker daemon of the Site Recovery Manager Virtual Appliance. It returns the ConfigurationTask object that can be used to monitor the operation. ConfigurationTask.getTaskInfo() method returns status information for the current task. ConfigurationTaskInfo.result returns a collection of SraImage objects.

Synopsis

```java
ConfigurationTask getSraImages()
```

ConfigurationTask is a managed object that provides operations to configure the SRM server. For more information see, Configuration Task.

SraImage describes an SRA image loaded into docker. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Information about the SRA as taken from ‘queryInfo’ command. It has the following values:</td>
</tr>
<tr>
<td></td>
<td>- uuid: Universally unique identifier of the SRA which is preserved on SRA upgrades.</td>
</tr>
<tr>
<td></td>
<td>- name: Name of the adapter.</td>
</tr>
<tr>
<td></td>
<td>- version: Version of the adapter.</td>
</tr>
<tr>
<td></td>
<td>- vendor: Storage Vendor who owns the adapter.</td>
</tr>
<tr>
<td></td>
<td>- helpUrl: URL for on-line documentation for the adapter.</td>
</tr>
<tr>
<td>imageId</td>
<td>Sha256 id of the image.</td>
</tr>
<tr>
<td>SraContainer[]</td>
<td>List of containers instantiated from the image.</td>
</tr>
<tr>
<td>repoTags</td>
<td>Docker repository tags the image is known by in the [REPOSITORY[:TAG]] format.</td>
</tr>
</tbody>
</table>
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>SRA image info as taken from the ‘queryInfo’ SRA command. This can be null if there was an error retrieving the info from the SRA. In this case <code>error</code> will be populated with the error that occurred.</td>
</tr>
<tr>
<td>error</td>
<td>Populated when there is an error executing the ‘queryInfo’ SRA command. The <code>info</code> field will be null in this scenario.</td>
</tr>
</tbody>
</table>

SraContainer describes an SRA container instantiated from an SRA docker image. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>containerId</td>
<td>The container id.</td>
</tr>
<tr>
<td>imageId</td>
<td>The sha256 image id of the image from which the container was instantiated.</td>
</tr>
<tr>
<td>MountPoint[] mountPoints</td>
<td>An array of mount points the container has, if any.</td>
</tr>
</tbody>
</table>

MountPoint describes a `<em>bind</em>` mount of the SRA container. For more information see [https://docs.docker.com/storage/bind-mounts/](https://docs.docker.com/storage/bind-mounts/).

MountPoint has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>The source of the mount point, i.e this is the path outside of the container.</td>
</tr>
<tr>
<td>destination</td>
<td>The destination path where the source is mounted at. This is the path inside the container.</td>
</tr>
</tbody>
</table>

### Faults

- RuntimeFault
- ServiceBusy
- SraOperationsDisabled

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### DeleteImage

This method stops and then deletes the containers instantiated from the given image, and deletes the image itself. It returns True if the image is deleted, and False if the image does not exist.

#### Synopsis

```java
boolean deleteImage(String imageId)
```

The `imageId` of the image that should be deleted. This can be either the sha256 id of the image, or the image repository tag in the `[REPOSITORY[:TAG]]` format.
Faults

- CannotExecuteDockerCommand
- DockerCommandFailed
- RuntimeFault
- SraOperationsDisabled

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

DeleteImageContainers

This method stops and then deletes the containers which were instantiated from the given image. It returns True if the image's containers are deleted, and False if the image does not exist or if the image does not have containers.

Synopsis

```java
boolean deleteImageContainers(String imageId)
```

imageId is the id of the image whose containers must be stopped. This can be either the sha256 id of the image or the image repository tag in the [REPOSITORY[:TAG]] format.

Faults

- CannotExecuteDockerCommand
- DockerCommandFailed
- RuntimeFault
- SraOperationsDisabled

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

GetImageInfo

This method returns the image information as taken from the queryInfo SRA command. Because the docker images are immutable, the queryInfo SRA command is executed only once, the first time the information is requested. Subsequent calls return a cached value which lives for the duration of the Site Recovery Manager Configuration Service lifetime.

Synopsis

```java
Info getImageInfo(String imageId)
```

imageId is the id of the image for which the information is being requested. This can be either the sha256 id of the image, or the image repository tag in the [REPOSITORY[:TAG]] format.
SraImage.Info is the information about the SRA as taken from the ‘queryInfo’ command. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>Universally unique identifier of the SRA which is preserved on SRA upgrades.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>version</td>
<td>Version of the adapter.</td>
</tr>
<tr>
<td>vendor</td>
<td>Storage Vendor who owns the adapter.</td>
</tr>
<tr>
<td>helpUrl</td>
<td>URL for on-line documentation for the adapter.</td>
</tr>
</tbody>
</table>

Faults

- CannotCreateSraLogDirectory
- CannotExecuteDockerCommand
- DockerCommandFailed
- DockerImageDoesNotExist
- RuntimeFault
- SraOperationsDisabled

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

CopySraConfiguration

This method copies the SRA configuration from a source image to a destination image.

Synopsis

```java
void copySraConfiguration(String fromImage, String toImage)
```

copySraConfiguration has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fromImage</td>
<td>Docker image id of the source image. It can be either in the [REPOSITORY:[TAG]] format or in the canonical sha256 format.</td>
</tr>
<tr>
<td>toImage</td>
<td>Docker image id of the destination image. It can be either in the [REPOSITORY:[TAG]] format or in the canonical sha256 format.</td>
</tr>
</tbody>
</table>

Faults

- CannotCreateSraLogDirectory
- CannotExecuteDockerCommand
ResetToFactorySettings

This method reverts the SRA image's configuration to its factory settings. Any changes to the configuration made since the image was loaded into docker will be lost.

Synopsis

```java
void resetToFactorySettings(String imageId)
```

imageId is the docker image id of the source image. It can be either in the [REPOSITORY:[TAG]] format or in the canonical sha256 format.

Faults

- CannotExecuteDockerCommand
- DockerCommandFailed
- DockerImageDoesNotExist
- RuntimeFault
- SraOperationsDisabled

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

SSL Certificate Manager

This section describes operations to configure certificates for the configuration service and SRM. These operations are applicable on SRM, VRMS, and VRS deployments.

The following methods that are used for setting or installing a new SRM certificate restarts the envoy service. When you are using the following APIs, if you encounter a ConnectionLost error, ignore it.

- SetCertificate
- SetKeyCertificate
- InstallSelfSignedCertificate
- InstallCertificate
Wait for the restart process to complete and reconnect to the SRM Configuration Service to make new calls.

**ProbeSsl**

This method checks if the Site Recovery Manager can establish successful SSL connection to the specified endpoint. It returns CertificateInfo that describes if this SRM server can validate the certificate coming from the specified endpoint.

**Synopsis**

```java
CertificateInfo probeSsl(String uri)
```

`uri` is the URL of the endpoint to probe.

CertificateInfo provides information about X509 certificate. It has the following fields:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificate</td>
<td>PEM encoded X509 certificate.</td>
</tr>
<tr>
<td>dnsName</td>
<td>DNS name of the server extracted from the certificate. The client is expected to use this DNS name to establish a secure connection to the server.</td>
</tr>
<tr>
<td>isTrusted</td>
<td>True if the SRM server can successfully validate the certificate without using thumbprints and false otherwise.</td>
</tr>
<tr>
<td>issuedTo</td>
<td>IP or FQDN of the receiver of the certificate.</td>
</tr>
<tr>
<td>issuedBy</td>
<td>IP or FQDN of the issuer. The one who signed the certificate.</td>
</tr>
<tr>
<td>expiresOn</td>
<td>Date on which the certificate expires.</td>
</tr>
</tbody>
</table>

**Faults**

- ConnectionRefusedFault
- DnsLookupFault
- HostUnreachableFault
- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**DrConfigGenerateCSR**

This method generates a new key and CSR, and returns the CSR for signing. It returns the certificate signing request in PEM format.
Synopsis

```java
String generateCSR(@optional CsrData requestData);
```

`requestData` parameter is the `CsrInfo` with desired CSR parameters.

`CsrData` is a data object. It provides information about about X509 certificate. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commonName</td>
<td>Common name to be set in the certificate. Usually the fully qualified domain name for server. If not set, data provided in FQDN (OS hostname) will be used.</td>
</tr>
<tr>
<td>organization</td>
<td>Exact legal name of the organization. Do not use an abbreviation. If not set, VMware default will be used.</td>
</tr>
<tr>
<td>organizationUnit</td>
<td>Section within the organization. If not set, VMware default will be used.</td>
</tr>
<tr>
<td>locality</td>
<td>City where organization is legally located. If not set, VMware default will be used.</td>
</tr>
<tr>
<td>state</td>
<td>State or province where organization is legally located. Do not use an abbreviation. If not set, VMware default will be used.</td>
</tr>
<tr>
<td>country</td>
<td>Two letter ISO abbreviation for organization country. If not set, VMware default will be used.</td>
</tr>
<tr>
<td>emailAddress</td>
<td>Email address to contact the organization. If not set, this will be empty.</td>
</tr>
<tr>
<td>fqdn</td>
<td>List of comma-separated FQDN strings to be used for SAN extensions. If not set, the OS hostname will be used.</td>
</tr>
<tr>
<td>ip</td>
<td>List of comma-separated IP strings to be used for SAN extensions. If not set, this will be empty.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

DrConfigSetCertificate

This method sets a new certificate. It reconfigures the Site Recovery Manager if already configured and restarts the proxy service. If you encounter an error while using this method, ignore the error and then reconnect to the management service.

Synopsis

```java
void setCertificate(String certificate, @optional String[] caChain)
```
setCertificate has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificate</td>
<td>New server certificate that must be used. It should be in PEM format.</td>
</tr>
<tr>
<td>caChain</td>
<td>List of intermediate CA certificates used to sign the server certificate. It should be in PEM format.</td>
</tr>
</tbody>
</table>

**Faults**

- CertificateBadKeyPair
- CertificateCaNotAllowed
- CertificateDnsMismatch
- CertificateHasExpired
- CertificateInvalidKeyLength
- CertificateMd5NotAllowed
- CertificateNotYetValid
- InvalidArgument
- PrivateKeyNotFound
- RuntimeFault
- SystemError

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**DrConfigSetKeyCertificate**

This method sets a new key and certificate, reconfigures Site Recovery Manager if already configured and then restarts the proxy service. If you encounter an error while using this method, ignore the error and then reconnect to the management service.

**Note**  This method does not allow you to password protect the private key. Consider using the following methods to set the key and certificate:

- generateCSR() method with the setCertificate() method.
- installCertificate() method

**Synopsis**

```java
void setKeyCertificate(String key, String certificate, @optional String[] caChain)
```

setKeyCertificate has the following parameters:
### Parameter and Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>New server private key to use in PEM format.</td>
</tr>
<tr>
<td>certificate</td>
<td>New server certificate to use in PEM format.</td>
</tr>
<tr>
<td>caChain</td>
<td>List of intermediate CA certificates, used to sign the server certificate in PEM format. During connect in 'Certificate' message, the server sends this chain and server certificate. The chain may or may not include the root CA.</td>
</tr>
</tbody>
</table>

### Faults

- CertificateBadKeyPair
- CertificateCaNotAllowed
- CertificateDnsMismatch
- CertificateHasExpired
- CertificateInvalidKeyLength
- CertificateMd5NotAllowed
- CertificateNotYetValid
- InvalidArgument
- RuntimeFault
- SystemError

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### AddCaCertificates

This method adds certificate authority certificates to the list of validating certificates. This list is used to validate server certificates when the Site Recovery Manager acts as a client.

#### Synopsis

```java
void addCaCertificates(@Optional String[] certs);
```

certs is an array of CA certificates. Each item contains a single certificate in PEM format.

#### Faults

- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).
**RemoveCaCertificates**

This method removes certificate authority certificates from the list of validating certificates. If a certificate requested for removal is not found, this will be noop.

**Synopsis**

```java
void removeCaCertificates(@optional String[] certs)
```

certs parameter is an array of CA certificates. Each item contains a single certificate in the PEM format.

**Faults**

- InvalidArgument
- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**RetrieveCaCertificates**

This method gets the current list of certificate authority certificates used by SRM to validate other server's certificates. It returns the collection of PEM encoded CA certificates. The list includes all system level certificates.

**Synopsis**

```java
String[] retrieveCaCertificates();
```

**Faults**

- RuntimeFault

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**ClearCaCertificates**

This method completely clears the certificate authority certificates installed on Site Recovery Manager Virtual Appliance.

**Synopsis**

```java
void clearCaCertificates();
```

**Faults**

- RuntimeFault
For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**InstallSelfSignedCertificate**

This method installs self-signed certificate, reconfigures the Site Recovery Manager if already configured, and restarts the proxy service. If you encounter an error while using this method, ignore the error and then reconnect to the management service.

**Synopsis**

```java
void installSelfSignedCertificate(CsrData csrData)
```

csrData is the CsrData object that should be used to sign the certificate.

CsrData is a data object. It provides information about about X509 certificate. For more information, see DrConfigGenerateCSR.

**Faults**

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

**InstallCertificate**

This method installs the PKCS#12 certificate, reconfigures the Site Recovery Manager if already configured, and restarts the proxy service. If you encounter an error while using this method, ignore the error and then reconnect to the management service.

**Synopsis**

```java
void installCertificate(String pkcs, @optional @secret String pkcsPassword)
```

InstallCertificate has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkcs</td>
<td>Certificate as string. Base64 encoded.</td>
</tr>
<tr>
<td>pkcsPassword</td>
<td>Password for the certificate.</td>
</tr>
</tbody>
</table>

**Faults**

- CertificateBadKeyPair
- CertificateCaNotAllowed
- CertificateDnsMismatch
- CertificateHasExpired
- CertificateInvalidKeyLength
CertificateMd5NotAllowed
CertificateNotYetValid
RuntimeFault
SystemError

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

GetCertificateInfo

This method lists the certificate info. Returns a CertificateInfo object that contains information about the certificate.

Synopsis

```java
CertificateInfo getCertificateInfo();
```

CertificateInfo provides information about X509 certificate. For more information, see ProbeSsl.

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

Update Manager

This section describes operations to update the Site Recovery Manager appliance. These operations are applicable on SRM, VRMS, and VRS deployments.

GetRunningTask

This method gets the currently active update task or null.

Synopsis

```java
ConfigurationTask getRunningTask();
```

ConfigurationTask is a managed object that provides operations to configure the SRM server. For more information, see Configuration Task.

Faults

- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.
**UpdateRepository**

This method changes the update repository location.

**Synopsis**

```java
void updateRepository(UpdateRepoInfo info)
```

`info` parameter is a `UpdateRepoInfo` object that contains information about the update repository location.

`UpdateRepoInfo` contains appliance update repository information. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Repository url.</td>
</tr>
<tr>
<td>mountedIso</td>
<td>True if the mounted iso is used. If this value is set, everything else is ignored.</td>
</tr>
<tr>
<td>username</td>
<td>User name</td>
</tr>
<tr>
<td>password</td>
<td>Password</td>
</tr>
</tbody>
</table>

**Faults**

- `InvalidArgument`
- `RuntimeFault`
- `SystemError`

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

**DrConfigCheckForUpdates**

This method checks the repository for available updates. It returns a task object used for monitoring the operation. It returns a null if no updates are available.

**Synopsis**

```java
ConfigurationTask checkForUpdates()
```

`ConfigurationTask` is a managed object that provides operations to configure the SRM server. For more information, see [Configuration Task](#).

`ConfigurationTask.getTaskInfo()` method returns status information for the current task. `ConfigurationTaskInfo.result` returns an `UpdateInfo` data object which to be used in `installUpdate` method.

`UpdateInfo` contains the appliance update repository information. It has the following fields:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateType</td>
<td>Enumerates the following update types:</td>
</tr>
<tr>
<td></td>
<td>- <strong>FEATURE</strong> - Update contains a new feature.</td>
</tr>
<tr>
<td></td>
<td>- <strong>SECURITY</strong> - Update contains security fixes.</td>
</tr>
<tr>
<td></td>
<td>- <strong>FIX</strong> - Update contains other fixes.</td>
</tr>
<tr>
<td></td>
<td>- <strong>MULTIPLE</strong> - Update contains multiple types of modifications.</td>
</tr>
<tr>
<td>UpdateSeverity</td>
<td>Enumerates the following levels of importance:</td>
</tr>
<tr>
<td></td>
<td>- <strong>MODERATE</strong> - Update is moderate.</td>
</tr>
<tr>
<td></td>
<td>- <strong>IMPORTANT</strong> - update is important.</td>
</tr>
<tr>
<td></td>
<td>- <strong>CRITICAL</strong> - Update is critical.</td>
</tr>
<tr>
<td>version</td>
<td>Update version string.</td>
</tr>
<tr>
<td>type</td>
<td>Update type. Value must be one of UpdateType enums.</td>
</tr>
<tr>
<td>releaseDate</td>
<td>Release date.</td>
</tr>
<tr>
<td>rebootRequired</td>
<td>Reboot required.</td>
</tr>
<tr>
<td>severity</td>
<td>Severity. Value must be one of UpdateSeverity enums.</td>
</tr>
<tr>
<td>summary</td>
<td>Update summary.</td>
</tr>
<tr>
<td>eula</td>
<td>End-user license agreement.</td>
</tr>
</tbody>
</table>

### Faults

- **RuntimeFault**
- **ServiceBusy**

For more information about the faults, see [Faults in Site Recovery Manager Appliance Management API](#).

### InstallUpdate

This method installs the update. It returns a task object used for monitoring the operation. CheckForUpdates must be called before this operation. This stops all the services to update them.

### Synopsis

```csharp
ConfigurationTask installUpdate(@optional UpdateInfo info)
```

info parameter is a UpdateInfo object that contains information for the update that should be installed. If null, installs the latest update.

UpdateInfo contains the appliance update repository information. For more information on UpdateInfo, see [DrConfigCheckForUpdates](#).

ConfigurationTask is a managed object that provides operations to configure SRM server. For more information, see [Configuration Task](#).
Faults
- InvalidArgument
- RuntimeFault
- UpdateNotAvailableFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.

GetRepositories
This method gets update repos. It returns a `UpdateRepoInfo` object representing the current repository and the default repository. It returns only one object if no custom repository is set.

Synopsis

```java
UpdateRepoInfo[] getRepositories();
```

`UpdateRepoInfo[]` contains the appliance update repository information. For more information on `UpdateRepoInfo`, see UpdateRepository.

Faults
- RuntimeFault

For more information about the faults, see Faults in Site Recovery Manager Appliance Management API.
This chapter contains descriptions for Site Recovery Manager APIs.

The API descriptions in this chapter follow the logical usage order of List of API Operations. In examples below, MoRef indicates a String that references a managed object.

**Note** In the various examples provided in this chapter, the srmPortType is a class instance which should be setup from the SDK examples. For more information, see About Java JAX-WS Samples.

This chapter includes the following topics:

- Service Instance
- SrmExtApiTask
- SRM Folder
- Inventory Mappings
- Autoprotect Manager
- Protection
- Protection Group Folder
- Create Protection Group Task
- Protection Group
- Protection Task
- Recovery
- Recovery Plan Folder
- Recovery Plan
- Recovery History
- IP Subnet Mapper
- Storage Adapter
- Storage
- ArrayManager
Service Instance

Site Recovery Manager methods take a managed object reference _this, which references the SessionManager used for making method calls. Programs obtain _this by retrieving content of the ServiceInstance, which is accomplished by creating a new managed object reference of type SrmServiceInstance.

**Example: C# code to create SrmServiceInstance**

```csharp
public SvcConnection(string svcRefVal)
{
    ...
    _svcRef = new ManagedObjectReference();
    _svcRef.type = "SrmServiceInstance";
    _svcRef.Value = svcRefVal;
}
```

The Java code is similar to the C# code.

**Example: Java code to create SrmServiceInstance**

```java
final ManagedObjectReference _svcRef = new ManagedObjectReference();
    _svcRef.setType("SrmServiceInstance");
    _svcRef.setValue("SrmServiceInstance");
```

**GetSiteName**

Gets the name of the current site.

**Note**  
The method is deprecated. You should not rely on it in production code, as it is not guaranteed to provide valid information in future releases. Instead, you should use `GetLocalSiteInfo` method.

**Synopsis**

```java
String getSiteName()
```

String representing the local Site Recovery Manager site name.

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.
GetPairedSite

This function gets the remote site paired with this site. A remote site may be acting as the secondary site for this site, or may be acting as the primary site with this site as its secondary. Most of the initial Site Recovery Manager calls work for everyone (System.Anonymous privilege) but GetPairedSite requires the System.View role, so it must be called after login to the local site. Also GetPairedSiteSolutionUserInfo is useless without the remote paired site, so the two must be called together.

Synopsis

```csharp
RemoteSite getPairedSite()
```

The `RemoteSite` class contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>a String with the name of the site.</td>
</tr>
<tr>
<td>uuid</td>
<td>a String with the UUID of the site.</td>
</tr>
<tr>
<td>vclHost</td>
<td>a String with the DNS name or IP address of the remote vCenter Server.</td>
</tr>
<tr>
<td></td>
<td>Note This property has been deprecated. You should not rely on this property in production code, as it is not guaranteed to contain valid information in future releases. Instead, you should use lkpUrl and vcInstanceUuid to locate services on the remote site.</td>
</tr>
<tr>
<td>vcInstanceUuid</td>
<td>a String with instance UUID of the vCenter Server associated with the remote Site Recovery Manager.</td>
</tr>
<tr>
<td>vcPort</td>
<td>an integer with the port used for VMOMI access to the remote vCenter Server.</td>
</tr>
<tr>
<td></td>
<td>Note This property has been deprecated. You should not rely on this property in production code, as it is not guaranteed to contain valid information in future releases. Instead, you should use lkpUrl and vcInstanceUuid to locate services on the remote site.</td>
</tr>
<tr>
<td>vcUrl</td>
<td>a String with the URL of the remote vCenter Server</td>
</tr>
<tr>
<td>connected</td>
<td>a boolean that is true when the sites are connected and false when the sites are disconnected.</td>
</tr>
<tr>
<td>lkpUrl</td>
<td>a String with the URL of the remote LookupService server.</td>
</tr>
</tbody>
</table>

Faults

- RemoteSiteNotEnabled if the remote site is not enabled.
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.
Example: Example

```java
com.vmware.srm.SrmRemoteSite remoteSite = srmPortType.getPairedSite(ManagedObjectReference _this);
Where ManagedObjectReference _this = _svcRef;
```

RetrieveContent

Retrieves properties of the service instance.

Synopsis

```java
ServiceInstanceContent retrieveContent()
```

The ServiceInstanceContent class contains the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>about</td>
<td>shows information about this service.</td>
</tr>
<tr>
<td>apiVersion</td>
<td>represents the version of this API.</td>
</tr>
<tr>
<td>inventoryMapping</td>
<td>an external API to InventoryMappings. For more information, see Inventory Mappings.</td>
</tr>
<tr>
<td>protection</td>
<td>an external API to Protection. For more information, see Protection.</td>
</tr>
<tr>
<td>recovery</td>
<td>an external API to Recovery. For more information, see Recovery.</td>
</tr>
<tr>
<td>storage</td>
<td>an external API to Storage. For more information, see Storage.</td>
</tr>
<tr>
<td>autoprotectManager</td>
<td>an external API to Automatic Protection. For more information, see Autoprotect Manager.</td>
</tr>
<tr>
<td>ipSubnetMapper</td>
<td>an external API to IP Subnet Mapper. For more information, see IP Subnet Mapper.</td>
</tr>
<tr>
<td>vvolReplication</td>
<td>an external API to Vvol Protection. For more information, see Vvol Replication.</td>
</tr>
<tr>
<td>propertyCollector</td>
<td>Property Collector for external API</td>
</tr>
<tr>
<td>placeholderDatastoreManager</td>
<td>External API to PlaceholderDatastoreManager</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetLocalSiteInfo

The getLocalSiteInfo method gets information about the local site.
Synopsis

LocalSiteInfo getLocalSiteInfo()

Returns information about the local site such as site name, UUID, and URLs of the local LookupService server and vCenter Server.

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Solution User Information

At install time Site Recovery Manager creates a solution user at the local and remote sites. This improves security by avoiding use of administrator@vsphere.local or the Windows administrator. Programs can obtain both solution users before Site Recovery Manager login because the privilege required for these functions is System.Anonymous.

GetSolutionUserInfo

Obtain the Site Recovery Manager solution user name and site UUID for the local site. You must call the GetSolutionUserInfo method before logging in (with SrmLoginByTokenLocale for example) if you wish to delegate a token to the Site Recovery Manager solution user.

Synopsis

SolutionUserInfo getSolutionUserInfo()

SolutionUserInfo.siteUuid is a string with the immutable unique identifier of the Site Recovery Manager server.

SolutionUserInfo.username is the name of the Site Recovery Manager solution user.

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetPairedSiteSolutionUserInfo

Obtain the Site Recovery Manager solution user name and site UUID for the paired remote site.

Synopsis

SolutionUserInfo getPairedSiteSolutionUserInfo()

SolutionUserInfo.siteUuid is a string with the immutable unique identifier of the remote Site Recovery Manager Server.

SolutionUserInfo.username is the name of the remote Site Recovery Manager solution user.
Faults

- RuntimeFault
- RemoteSiteNotEnabled, if the remote site is not enabled.

See Faults in Site Recovery Manager API for more details.

**SAML Token Authentication**

In the 6.0 release, Site Recovery Manager interacts with both vCenter Server (as before) and Platform Services Controller (PSC).

The PSC contains a Lookup Service to locate other services, a Licensing Service to replace the VIM license manager, and Single Sign On (SSO) service for authentication and token acquisition.

The vCenter Server management node contains a new Tag Manager to create categories and tags for Storage DRS or SPBM, and a new Authz service to replace the VIM authorization manager.

*Figure 7-1. Management (M) and PSC (N) nodes*

When programs connect to the local or remote SRM server, they must obtain a SAML token from the SSO service on the local or remote PSC. (The older login functions implicitly obtain a SAML token.)
**SrmLoginByTokenLocale**

This function begins a session with Site Recovery Manager Server.

**Synopsis**

```java
void loginByToken(String samlToken, @optional String locale)
```

- `samlToken` is an XML encoded security assertion markup language (SAML) token for authenticating login to the SRM server. The token should either be a bearer token or a holder of key token delegated to the Site Recovery Manager solution user.
- `locale` is the optional locale for this session.

**Faults**

- `AlreadyLoggedInFault` if there is already an established session.
- `ConnectionLimitReached` if the configured connection limit has been reached.
- `InvalidLogin` if the given token is not valid.
- `InvalidLocale` if the requested locale is invalid or unavailable.
- `InvalidTokenLifetime` if the SSO token is either expired or not yet valid.
- `NoPermission` if the user does not have permissions.
- `RuntimeFault`

See [Faults in Site Recovery Manager API](#) for more details.

---

**SrmLoginSitesByToken**

Log in to both the local and remote vCenter Servers using the provided tokens. This function is needed when escalated privileges are required to perform operations on the remote site, such as protecting virtual machines.

**Synopsis**

```java
void loginSitesByToken(String samlToken, String remoteSamlToken, @optional String locale)
```

- `samlToken` is an XML encoded SAML token for authenticating login to the Site Recovery Manager server. The token should either be a bearer token or a holder of key token delegated to the Site Recovery Manager solution user.
- `remoteSamlToken` is an XML encoded SAML token for authenticating login to the remote Site Recovery Manager server. The token should either be a bearer token or a holder of key token delegated to the remote Site Recovery Manager solution user.
- `locale` is the optional locale for this session.
Faults

- AlreadyLoggedInFault if there is already an established session.
- ConnectionLimitReached if the configured connection limit has been reached.
- InvalidLogin if the given token is not valid.
- InvalidLocale if the requested locale is invalid or unavailable.
- NoPermission
- RemoteSiteNotEnabled if the remote site is not enabled.
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

SrmLoginRemoteSiteByToken

Log in to the remote Site Recovery Manager server using the provided credentials. This function may be called when escalated privileges are required on the remote site and the current session has already been authenticated by login.

Synopsis

```java
void loginRemoteSiteByToken(String remoteSamlToken,
                             @optional String locale)
```

remoteSamlToken is an XML encoded SAML token for authenticating login to the Site Recovery Manager server. The token should either be a bearer token or a holder of key token delegated to the remote Site Recovery Manager solution user.

locale is the optional locale for this session.

Faults

- AlreadyLoggedInFault if there is already an established session.
- ConnectionDownFault
- ConnectionLimitReached if the configured connection limit has been reached.
- InvalidLogin if the given token is not valid.
- InvalidLocale if the requested locale is invalid or unavailable.
- NotAuthenticated if there is no session
- RemoteSiteNotEnabled if the remote site is not enabled.
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.
Credential Based Authentication

In the 6.0 release, login functions in earlier Site Recovery Manager releases have been implemented on top of the SAML token authentication functions. The password-based login functions are now offered for convenience.

Programs can connect to the local Site Recovery Manager Server using the SrmLoginLocale function, or to Site Recovery Manager Server instances at both the (local) protected site and the (remote) recovery site using the SrmLoginSites API.

SrmLoginLocale

This method logs in to the Site Recovery Manager server. The Connect public method requires the URL of a Site Recovery Manager server and authentication credentials. The SrmLoginLocale method takes the _srcRef managed object reference from SrmServiceInstance, and fails if the user name and password combination is invalid, or if the user is already logged in. In these examples, a locale string could be provided instead of the null parameter.

Synopsis

```csharp
void SrmLoginLocale(String username, String password, @optional String locale)
```

username is the user name authorized for access to the local vCenter Server.

password is the password for that user on the local vCenter Server.

locale is the name of the locale for this session.

Faults

- AlreadyLoggedInFault
- ConnectionLimitReached
- InvalidLocale
- InvalidLogin
- NoPermission
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: C# code for Site Recovery Manager login

```csharp
protected SrmService _service;
protected SrmServiceInstanceContent _sic;
protected ManagedObjectReference _svcRef;
...
public void Connect(string url, string username, string password)
{
    _service = new SrmService();
    _service.Url = url;
    _service.Timeout = 600000;
}
The Java code is similar to the C# code but uses a service locator.

**Example: Java code for Site Recovery Manager login**

```java
private static SrmPortType srmPort;
private static SrmServiceInstanceContent serviceContent;
private static boolean isConnected = false;
...
srmPort = srmService.getSrmPort();
Map<String, Object> ctxt =
((BindingProvider) srmPort).getRequestContext();
ctxt.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, url);
ctxt.put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
serviceContent = srmPort.retrieveContent(_svcRef);
srmPort.srmLoginLocale(_svcRef, userName, password, null);
isConnected = true;
```

Subsequent methods in the Site Recovery Manager are called as a subclass of _service, for example _service.ListPlans() in C# or srmpoint.listPlans() in Java.

**SrmLoginSites**

The SrmLoginSites API is very similar to SrmLoginLocale, except it takes an additional user name and password combination for the remote (usually recovery) site. The SrmServiceInstance _this is obtained from the local (usually protected) site.

**Synopsis**

```java
void SrmLoginSites(String username, String password, String remoteUser, String remotepass, @optional String locale)
```

- **username** is the user name authorized for access to the local vCenter Server
- **password** is the password for that user on the local vCenter Server
- **remoteUser** is the user name authorized for access to the remote vCenter Server
- **remotePass** is the password for that user on the remote vCenter Server
- **locale** is the name of the locale for this session

**Faults**

- AlreadyLoggedInFault
- ConnectionLimitReached
- InvalidLocale
- InvalidLogin
- NoPermission
- RemoteSiteNotEnabled
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**Example: C# and Java for double SRM login**

```csharp
///<C#
_service.SrmLoginSites(_svcRef, username, password, remoteuser, remotepass, locale);
//srmPort.srmLoginSites(_svcRef, username, password, remoteuser, remotepass, locale);
```

**SrmLogoutLocale**

This method logs out of the Site Recovery Manager server and terminates the current session. It takes the same managed object reference as for `SrmLoginLocale`, and should be called with other methods to clean up a connection.

**Synopsis**

```csharp
void SrmLogoutLocale();
```

**Faults**

- NotAuthenticated
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**Example: C# code to log out**

```csharp
public void Disconnect() {
    if (_service != null) {
        _service.SrmLogoutLocale(_svcRef);
        _service.Dispose();
        _service = null;
        _sic = null;
    }
}
```

The Java code is simpler than the C# code.

**Example: Java code to log out**

```java
private static void disconnect() throws Exception {
    if (isConnected) {
        srmPort.srmLogoutLocale(_svcRef);
    }
    isConnected = false;
}
```
SrmLoginRemoteSite

Log in to the remote Site Recovery Manager server using the provided credentials. This function may be invoked when escalated privileges are required on the remote site and the current session has already been authenticated using login.

Synopsis

```java
void SrmLoginRemoteSite(String remoteUser, String remotePassword, @optional String locale)
```

- `remoteUser` is the username to be used to login to the remote VirtualCenter server.
- `remotePassword` is the password to be used to login to the remote VirtualCenter server.
- `locale` is the locale for this session.

Faults

- `AlreadyLoggedInFault`
- `ConnectionLimitReached`
- `InvalidLocale`
- `InvalidLogin`
- `RemoteSiteNotEnabled`
- `RuntimeFault`

See Faults in Site Recovery Manager API for more details.

GetLicenseInfo

This method gets the assigned license information.

Synopsis

```java
LicenseInfo getLicenseInfo();
```

`LicenseInfo` is a data object. It encapsulates information about a license. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>editionKey</code></td>
<td>The license edition</td>
</tr>
<tr>
<td><code>costUnit</code></td>
<td>The cost unit for this license. <code>costUnit</code> enumerates the following:</td>
</tr>
<tr>
<td></td>
<td>- <code>cpuPackage</code>: One license is required per CPU package</td>
</tr>
<tr>
<td></td>
<td>- <code>vm</code>: One license is required per VM</td>
</tr>
<tr>
<td></td>
<td>- <code>unknown</code>: Unknown cost unit</td>
</tr>
<tr>
<td><code>total</code></td>
<td>Total capacity of the license</td>
</tr>
<tr>
<td><code>used</code></td>
<td>Number of units currently used from this asset</td>
</tr>
<tr>
<td><code>productName</code></td>
<td>The product name for this license</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>productVersion</td>
<td>The product version for this license</td>
</tr>
<tr>
<td>expiryDays</td>
<td>Number of days left until the license expires.</td>
</tr>
<tr>
<td>expiryDate</td>
<td>Expiration date of the license.</td>
</tr>
<tr>
<td>inUseFeatures</td>
<td>The features in this license.</td>
</tr>
</tbody>
</table>

**Faults**
- RuntimeFault

See **Faults in Site Recovery Manager API** for more details.

**ProbeSsl**

This method returns (host and thumbprint) tuples for all PSC/MGMT/DR hosts to which SRM should connect under specified root PSC node.

**Synopsis**

```java
@optional HostThumbprintInfo[] probeSsl(String host, @optional int port, @optional String vcHost)
```

The `probeSsl` method returns an array of `HostThumbprintInfo` structures. Each entry contains host address and required thumbprint to establish connection to the services on that host. The `probeSsl` method has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host name or IP address of the root PSC node.</td>
</tr>
<tr>
<td>port</td>
<td>Port of the root PSC node. When not provided the default value of 443 is used.</td>
</tr>
<tr>
<td>vcHost</td>
<td>Host where the VC is running. In case of embedded environment this field should not be specified. In this case the host on which the VC is running is the PSC host.</td>
</tr>
</tbody>
</table>

`HostThumbprintInfo` is a data object that contains host name/address and required thumbprint to establish connection to the services on that host. It has the following fields:
Field | Description
---|---
host | FQDN or IP address of the host.

Example of SSL command to extract SHA-2 fingerprint from host is:
```bash
openssl s_client -connect <host>:<port> < /dev/null 2>/dev/null |
openssl x509 -fingerprint -sha256 -noout -in /dev/stdin
```

Faults

- RuntimeFault
- SitePairingFault

See [Faults in Site Recovery Manager API](#) for more details.

**PairSrm**

This method establishes the persistent network connection with a remote SRM server. Remote SRM server must have the same VC extension key.

**Synopsis**

```java
@task RemoteSite pairSrm(ConnectionSpec spec);
```

This method returns the drextapi.Task that contains the paired drextapi.RemoteSite. For more information, see [GetPairedSite](#).

`ConnectionSpec` contains the connection information used to connect to PSC node and locate peer services. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host name or IP address of the PSC node.</td>
</tr>
<tr>
<td>port</td>
<td>Port of the PSC node. When not provided the default value of 443 is used.</td>
</tr>
<tr>
<td>vcHost</td>
<td>Host where the VC is running. In case of embedded environment this field should not be specified. In this case the host on which the VC is running is the PSC host.</td>
</tr>
</tbody>
</table>
### HostThumbprintInfo[] thumbprints

Thumbprints for PSC/MGMT/DR hosts to which SRM should connect when certificate validation fails. For more information, see `ProbeSsl`.

### creds

Credentials to be used to authenticate to the remote PSC node. Credentials has the following fields:

- **String user** - The name of a user with sufficient privileges to perform configuration tasks on the infrastructure and management nodes as well as SSO service configuration tasks on the infrastructure node.
- **@secret String** - Password for the user.

---

**Note** If the returned task fails, its error field may contain one of the following:

- **drextapi.site.fault.SelfPairFault** - If this SRM is already registered with the specified PSC/MGMT node.
- **drextapi.site.fault.SitePairingFault** - Contains the original error when a network connection cannot be established, there are invalid arguments, local site name is the same as remote SRM's name, another pair operation is in progress or there was other error while pairing sites.
- **drextapi.site.fault.AlreadyPairedFault** - If any of the SRM servers involved with the pairing is already paired.
- **drextapi.fault.RemoteSiteNotInitialized** - If the remote site is not correctly initialized.

### Faults

- **RuntimeFault**

See [Faults in Site Recovery Manager API](#) for more details.

### BreakPairing

This method removes the connection with the remote SRM server. This operation will automatically logs out from the remote site.

#### Synopsis

```java
@task void breakPairing(RemoteSite site)
```

The `breakPairing` method returns the `drextapi.Task` to track the operation. For more information, see `SrmExtApiTask`. `breakPairing` has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>site</td>
<td>The remote SRM server to remove. For more information, see <code>GetPairedSite</code>.</td>
</tr>
</tbody>
</table>
Note  If the returned task fails, its error field may contain:

- drextapi.fault.SitePairingFault - If another pairing operation is in progress, site still has protected objects or a paired remote site cannot be found.

Faults

- InvalidArgument
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

ReconfigureConnection

This method reconfigures the RemoteSite object with connection information for remote PSC node and propagates the changed values to the paired site.

Synopsis

```java
@task void reconfigureConnection(RemoteSite site, ConnectionSpec spec)
```

This method returns the drextapi.Task to track the operation. For more information, see SrmExtApiTask. ReconfigureConnection has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>site</td>
<td>RemoteSite object to reconfigure. For more information, see GetPairedSite.</td>
</tr>
<tr>
<td>spec</td>
<td>The ConnectionSpec to use for the remote PSC node connection.</td>
</tr>
</tbody>
</table>

Note  Task result contains the paired drextapi.RemoteSite. For more information, see GetPairedSite.

If the returned task fails, its error field may contain one of the following:

- drextapi.site.fault.SelfPairFault - If this SRM is already registered with the specified PSC/MGMT node.
- drextapi.site.fault.SitePairingFault - Contains the original error when a network connection cannot be established, there are invalid arguments, the remote site name or network connections policy is violated, either this SRM or the SRM registered with the given PSC/MGMT node is not paired with the other, another pair operation is in progress or there was other error while repairing sites.

Faults

- InvalidArgument
- RuntimeFault
See Faults in Site Recovery Manager API for more details.

**SrmExtApiTask**

A base external API task.

**IsSrmExtApiTaskComplete**

This method checks whether the task has been completed.

**Synopsis**

```java
boolean isComplete()
```

Returns True if this task has been completed.

**GetSrmExtApiTaskInfo**

This method gets detailed results of the completed task.

**Synopsis**

```java
TaskInfo getTaskInfo()
```

TaskInfo is a data object that contains all information about a task. For more information see TaskInfo in the Site Recovery Manager API Reference Guide.

**SRM Folder**

This section covers the Site Recovery Manager API methods for Folders. This is the base class for all folder types in SRM APIs.

**GetName**

Retrieves the name of this folder object, given existence of a ProtectionGroupFolder or RecoveryPlanFolder.

**Synopsis**

```java
String getName()
```

**Faults**

- RuntimeFault

See for Faults in Site Recovery Manager API more details.
Example: Example for GetName

```java
java.lang.String name = srmPortType.getName(ManagedObjectReference _this);
```

where ManagedObjectReference _this = _folderRef;
where _folderRef can be taken from:
```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
ManagedObjectReference _folderRef = srmPortType.getProtectionGroupRootFolder(ManagedObjectReference _this);
where ManagedObjectReference _this = _protectionRef;
```

GetParentFolder

Gets a reference to the parent folder. Folder extends and is inherited by Site Recovery Manager from the vSphere API, where it is a managed object acting as a container to store and organize inventory objects, such as protection groups and recovery plans.

**Synopsis**

```java
Folder getParentFolder( )
```

Returns the parent Folder as a managed object. This value is null for the root object.

**Faults**

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

GetChildType

Specifies the object types a folder may contain. When you create a folder, it inherits its childType from the parent folder in which it is created.

**Synopsis**

```java
@optional TypeName[] getChildType();
```

childType is an array of strings.

**Faults**

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

CreateFolder

Creates a new sub-folder with the specified name.
Because of the dual-server nature of SRM, the sites must be connected when creating folders. Any % (percent) character used in this name parameter must be escaped, unless it is used to start an escape sequence. Clients may also escape any other characters in this name parameter. This method requires VcDr.ProtectionProfile.com.vmware.vcDr.Create privilege on the containing folder to create a protection group folder and VcDr.RecoveryProfile.com.vmware.vcDr.Create to create a recovery plan folder.

**Synopsis**

```java
@task Folder createFolder(String folderName)
```

createFolder returns a task instance to monitor the asynchronous operation of this method. Folder object is returned as task result. createFolder contains the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name to be given the new folder. An entity name must be a non-empty string of less than 80 characters. The slash (/), backslash () and percent (%) will be escaped using the URL syntax. For example, %2F.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain one of the following:

- `vim.fault.DuplicateName` - if another object in the same folder has the target name.
- `vim.fault.InvalidName` - if the name is not a valid entity name.
- `drestapi.fault.ConnectionDownFault` - if the sites are not connected.

**Faults**

- `vim.fault.InvalidArgument`
- `RuntimeFault`

See [Faults in Site Recovery Manager API](#) for more details.

**MoveFolder**

Moves this folder into another folder.

The objects that can be moved into a folder depends on the parent folder's type (as defined by the parent folder's `childType()` property). For a folder constructed for recovery plans, only recovery plans and recovery folders can be moved into the folder. For a folder constructed to hold `ProtectionGroups`, only `ProtectionGroups` and protection folders can be moved into the folder.

For protection group folders this method requires VcDr.ProtectionProfile.com.vmware.vcDr.Edit on the moved folder, on the current parent folder, and on the destination folder.

For recovery plan folders this method requires VcDr.RecoveryProfile.com.vmware.vcDr.Edit on the moved folder, on the current parent folder, and on the destination folder.
Synopsis

```java
@task void moveFolder(Folder destination);
```

`moveFolder` returns a task instance to monitor the asynchronous operation of this method. `moveFolder` has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td>Folder that should become the new parent of this folder.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain one of the following:

- `drextapi.fault.DuplicateName` - a folder with the same name already exists within the destination folder.
- `drextapi.fault.IllegalMove` - if a cycle would be created by this move. For example, moving this folder into one of its children folders would create a cycle.
- `drextapi.fault.ImmutableFolder` - if move is initiated on the root folder.
- `vim.fault.NotSupported` - if the folder is being moved into a folder whose `childType()` property is not set to the appropriate value. For example, a folder cannot be moved into a folder whose `ChildType` property value does not contain `Dr::Folder`.

Faults

- `InvalidType`
- `ManagedObjectNotFound`
- `RuntimeFault`

See [Faults in Site Recovery Manager API](#) for more details.

DestroyFolder

Destroys the specified `Folder`.

This method requires `VcDr.ProtectionProfile.com.vmware.vcDr.Delete` privilege for protection group folders and `VcDr.RecoveryProfile.com.vmware.vcDr.Delete` for recovery plan folders.

Synopsis

```java
@task void destroyFolder();
```

`destroyFolder` returns a task instance to monitor the asynchronous operation of this method. If a task fails, its error field may contain one of the following:

- `drextapi.fault.NotEmpty` - if this folder still contains child items.
- `drextapi.fault.ConnectionDownFault` - if the sites are not connected.
Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

RenameFolder

Renames the specified Folder. This method requires VcDr.ProtectionProfile.com.vmware.vcDr.Edit privilege for protection group folders and VcDr.RecoveryProfile.com.vmware.vcDr.Edit for recovery plan folders.

Synopsis

```java
@task void renameFolder(String newName)
```

renameFolder returns a task instance to monitor the asynchronous operation of this method.

renameFolder has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newName</td>
<td>The new folder name.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain the following:

- drextapi.fault.ConnectionDownFault - if the sites are not connected.

Faults

- vim.fault.InvalidArgument
- RuntimeFault
- StringArgumentTooLong

See Faults in Site Recovery Manager API for more details.

Inventory Mappings

This section covers the Site Recovery Manager API methods for inventory (resource) mapping.

Resource mappings are the pairings of resources between the protected and recovery sites. In other words, mapping the networks, resource pools, datacenters and so forth on the protected site to their counterparts on the recovery site. This is done so that virtual machines will recover in the correct places on the recovery site. Previously this was done only in the UI, but APIs have been added to automate these mappings.

AddFolderMapping

Adds a folder mapping between a folder on the primary vCenter Server and another folder on the secondary vCenter Server.
Synopsis

void addFolderMapping(vim.Folder primaryFolder, vim.Folder secondaryFolder)

primaryFolder is a read-only MoRef to the folder on the protection site (must be local).
secondaryFolder is a read-only MoRef to the folder on the recovery site (must be remote).

Faults

- ConnectionDownFault
- InvalidPrimaryFolder
- InvalidSecondaryFolder
- NotAuthenticated
- RuntimeFault
- UnknownPrimaryFolder
- UnknownSecondaryFolder

See Faults in Site Recovery Manager API for more details.

RemoveFolderMapping

The removeFolderMapping method removes a folder mapping. The method does not check whether
the folders in the mapping exist on the protected and recovery sites. You can use the method to
remove broken mappings.

Synopsis

void removeFolderMapping(Folder primaryFolder)

The primaryFolder parameter specifies the folder on the primary site whose mapping must be
deleted.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery
Manager API.

AddNetworkMapping

Adds a network mapping between a network on the primary vCenter Server and another
network on the secondary vCenter Server.

Synopsis

primaryNetwork is a read-only MoRef to the network on the protection site (must be local).
secondaryNetwork is a read-only MoRef to the network on the recovery site (must be remote).

Faults
- ConnectionDownFault
- InvalidPrimaryNetwork
- InvalidSecondaryNetwork
- NotAuthenticated
- RuntimeFault
- UnknownPrimaryNetwork
- UnknownSecondaryNetwork

See Faults in Site Recovery Manager API for more details.

RemoveNetworkMapping
The removeNetworkMapping method removes a network mapping. The method does not check whether the networks in the mapping exist on the protected and recovery sites. You can use the method to remove broken mappings.

Synopsis
```java
void removeNetworkMapping(Network primaryNetwork)
```

The primaryNetwork parameter specifies the primary site network whose mapping must be deleted.

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

AddResourcePoolMapping
Adds a resource pool mapping between a resource pool on the primary vCenter Server and another on the secondary vCenter Server.

Synopsis
```java
```

addResourcePoolMapping contains the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primaryPool</td>
<td>resource pool on the protection site (must be local)</td>
</tr>
<tr>
<td>secondaryPool</td>
<td>resource pool on the recovery site (must be remote)</td>
</tr>
</tbody>
</table>

### Faults
- ConnectionDownFault
- NotAuthenticated
- RuntimeFault
- UnknownPrimaryResourcePool
- UnknownSecondaryResourcePool

See [Faults in Site Recovery Manager API](#) for more details.

### RemoveResourcePoolMapping

The `removeResourcePoolMapping` method removes a resource pool mapping. The method does not check whether the pools in the mapping exists on the protected and recovery sites.

#### Synopsis

```java
void removeResourcePoolMapping(ResourcePool primaryPool)
```

The `primaryPool` parameter specifies the resource pool on the primary site whose mapping must be deleted.

#### Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

### AddTestNetworkMapping

The `addTestNetworkMapping` method adds a test network mapping between a network on the secondary site and a network on the secondary site that is used for testing.

#### Synopsis

```java
void addTestNetworkMapping(@readonly Network secondaryNetwork, 
                          @readonly Network destinationTestNetwork)
```

`addTestNetworkMapping` has the following fields:
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secondaryNetwork</td>
<td>Specifies a network on the remote recovery site</td>
</tr>
<tr>
<td>destinationTestNetwork</td>
<td>Specifies a test network on the remote recovery site</td>
</tr>
</tbody>
</table>

### Faults
- ConnectionDownFault
- InvalidSecondaryNetwork
- RemoteSiteNotAuthenticated
- RuntimeFault
- UnknownSecondaryNetwork

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

### RemoveTestNetworkMapping

The `removeTestNetworkMapping` method removes the test network mapping. The method does not check whether the networks in the mapping exist on the secondary site. You can use the method to remove broken mappings.

**Synopsis**

```java
void removeTestNetworkMapping(vim.Network secondaryNetwork)
```

The `secondaryNetwork` parameter specifies the network on the secondary site whose mapping must be removed.

### Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

### GetFolderMappings

This method returns an array of the folder mappings for this inventory mapper. If one of the mappings does not have a secondary object, it means that the user does not have enough permissions to see that object on the secondary VC.

**Synopsis**

```java
FolderMapping[] getFolderMappings();
```

`FolderMapping[]` is an array of mappings from a folder on the primary site to a folder on the secondary site with the following fields:
Field | Description
--- | ---
primaryObject | Folder on the primary site
secondaryObject | Folder on the secondary site

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetNetworkMappings
This method returns an array of the network mappings for this inventory mapper. If one of the mappings does not have a secondary object, it indicates that the user does not have enough permissions to see that object on the secondary VC.

Synopsis
```java
NetworkMapping[] getNetworkMappings();
```

`NetworkMapping[]` is an array of mappings from a network on the primary site to a network on the secondary site with the following fields:

Field | Description
--- | ---
primaryObject | Network on the primary site
secondaryObject | Network on the secondary site

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetResourcePoolMappings
This method returns an array of the resource pool mappings for this inventory mapper. If one of the mappings does not have a secondary object, it means that the user does not have enough permissions to see that object on the secondary VC.

Synopsis
```java
ResourcePoolMapping[] getResourcePoolMappings();
```

`ResourcePoolMapping[]` is an array of mappings from a resource pool on the primary site to a resource pool on the secondary site with the following fields:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primaryObject</td>
<td>Resource pool on the primary site</td>
</tr>
<tr>
<td>secondaryObject</td>
<td>Resource pool on the secondary site</td>
</tr>
</tbody>
</table>

#### Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

### GetTestNetworkMappings

This method returns an array of the test network mappings for this inventory mapper.

#### Synopsis

```java
TestNetworkMapping[] getTestNetworkMappings();
```

TestNetworkMapping[] is an array of mappings from a network on the recovery site to a network that should be used for testing. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Network on the recovery site</td>
</tr>
<tr>
<td>testNetwork</td>
<td>Network to be used while in the testing mode</td>
</tr>
</tbody>
</table>

#### Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

### Autoprotect Manager

This section presents methods to enable automatic protection and related configuration settings.

#### SetAutoprotectUser

This method configures the user to be used by the automatic protection on this site. If not called, the default autoprotect user is used.

#### Synopsis

```java
void setAutoprotectUser(String user)
```

The `user` parameter specifies the name of the user that will be used by automatic protection.
Faults

- RuntimeFault
- vmomi.fault.InvalidArgument
- vim.fault.NoPermission

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetAutoprotectUser

This method gets the user for automatic protection. Returns the current user account to be used by automatic protection on this site.

Synopsis

```java
String getAutoprotectUser()
```

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

SetDefaultAutoprotectUser

This method reverts the current user to the default user.

Synopsis

```java
void setDefaultAutoprotectUser()
```

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

IsActive

Checks if the automatic protection is globally activated (true) or deactivated (false). Automatic protection is active when both SRM servers in a pair support AutoprotectManager and the connection to the remote site is healthy. Automatic protection is not active when a peer SRM server does not support AutoprotectManager or the connection to the remote site is broken.

Synopsis

```java
boolean IsActive()
```
Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

Protection

This section covers the Site Recovery Manager API methods for protection groups and virtual machine replication.

In SRM 5.8, new APIs appeared to create protection groups, assign them to recovery plans, and protect virtual machines using array-based or host-based replication. The new APIs provide three types of functionality for vSphere disaster recovery operations:

1. **Infrastructure**
   - workflows to create protection groups
   - workflows to create inventory mappings between matching objects
   - workflows to add protection groups to recovery plans
2. **Virtual machine (VM) protection**
   - workflows to protect VMs using a pre-configured array-based protection group
   - workflows to protect VMs using a pre-configured host-replicated protection group
3. **Virtual machine (VM) recovery settings**
   - recovery priority
   - per-VM callouts
   - final power state

ListProtectionGroups

This method lists the configured protection groups.

Synopsis

```plaintext
SrmProtectionGroup[] listProtectionGroups()
```

$SrmProtectionGroup[]$ is an array of managed object references to all the SrmProtectionGroup managed objects that are currently configured. For more information, see [Protection Group](#).

Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.
Example: Method to list protection groups

```java
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionRef;
where _protectionRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
```

ListInventoryMappings

This method returns the configured inventory mappings. You establish placeholder datastores as described in the VMware Site Recovery Manager Documentation. You establish inventory mappings using the AddFolderMapping, AddNetworkMapping, and AddResourcePoolMapping methods documented in this manual, or using procedures described in the VMware Site Recovery Manager Documentation.

The destinations of each of the resources (network, resource pool, and folder) are not available in the data structure that is returned. If necessary, use the following:

-   GetNetworkMappings
-   GetResourcePoolMappings
-   GetFolderMappings

Synopsis

```java
Protection.InventoryMappingInfo listInventoryMappings( )
```

inventoryMappingInfo is a list of inventory mappings from the protected site to the recovery site:

-   folders is a list of mapped VirtualMachine Folders.
-   networks is a list of mapped virtual machine Networks and dvPortgroups.
-   pools is a list of mapped Resource Pools.
-   testNetworks is a list of mapped networks to test networks.

Faults

-   RuntimeFault

See Faults in Site Recovery Manager API for more details.
Example: Method to list inventory mappings

```java
SrmProtectionInventoryMappingInfo info = srmPortType.listInventoryMappings(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionRef;
where _protectionRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
```

### ListReplicatedDatastores

This method queries and lists replicated but unprotected datastores. A datastore is replicated if it contains any virtual machines in a protection group.

**Note**  This method is deprecated. As an alternative, use ListUnassignedReplicatedDatastores.

### Synopsis

```java
vim.Datastore[] listReplicatedDatastores()
```

Datastore[] is a list of replicated datastores on this site that can be used to create new protection groups.

### Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Method to list replicated datastores

```java
List < ManagedObjectReference > datastores =
    srmPortType.listReplicatedDatastores(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionRef;
where _protectionRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
```

### GetProtectionGroupRootFolder

Returns a reference to the top-level container for protection groups.

### Synopsis

```java
ProtectionGroupFolder getProtectionGroupRootFolder()
```

ProtectionGroupFolder is the top-level folder for protection groups. For more information, see Protection Group Folder.
Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

### ListUnassignedReplicatedDatastores

Gets a list of replicated datastores that can be used to create new protection groups.

**Synopsis**

```java
vim.Datastore[] listUnassignedReplicatedDatastores()
```

`Datastore[]` is a list of all datastores on this site that are replicated but not currently protected by Site Recovery Manager.

Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

### ProtectionListProtectedDatastores

Get a list of the replicated datastores that are protected by Site Recovery Manager.

**Synopsis**

```java
vim.Datastore[] ProtectionListProtectedDatastores()
```

`Datastore[]` is a list of all datastores on this site that are replicated and protected by Site Recovery Manager.

Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

### ListUnassignedReplicatedVms

Gets a list of replicated VMs that are currently not assigned to a Site Recovery Manager protection group.

**Synopsis**

```java
vim.VirtualMachine[] listUnassignedReplicatedVms(String replicationType)
```

`listUnassignedReplicatedVms` has the following fields:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicationType</td>
<td>an enumeration defined in SrmProtectionGroup. Valid values are san for ABR replicated virtual machines, vr for HBR replicated virtual machines, and vvol for Vvol replicated virtual machines.</td>
</tr>
<tr>
<td>VirtualMachine[]</td>
<td>an enumeration defined in SrmProtectionGroup. Valid values are san for ABR replicated virtual machines, vr for HBR replicated virtual machines, and vvol for Vvol replicated virtual machines.</td>
</tr>
</tbody>
</table>

### Faults

- **InvalidArgument**
- **RuntimeFault**

See [Faults in Site Recovery Manager API](#) for more details.

### ProtectionListProtectedVms

Get a list of virtual machines that are protected by the Site Recovery Manager.

**Synopsis**

```java
vim.VirtualMachine[] ProtectionListProtectedVms()
```

VirtualMachine[] is a list of protected virtual machines.

**Faults**

- **RuntimeFault**

See [Faults in Site Recovery Manager API](#) for more details.

### CreateAbrProtectionGroup

Create a new storage array-based ProtectionGroup using the provided datastores. This method does not automatically protect VMs on the storage array. Programs must call `ProtectVms()` separately for VMs on the storage array to be protected. If you have a replicated datastore with an existing VM on it and then create an ABR group, the VM will not be auto protected. Workaround is to add another VM to the datastore after the ABR protection group is created. This causes both the VMs to be protected with autoprotect.

**Note**  The protection group name cannot be the same as the folder in which it will be created.

**Synopsis**

```java
CreateProtectionGroupTask createAbrProtectionGroup(Folder location, String name, @optional String description, vim.Datastore[] datastores)
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>folder in which to create the protection group</td>
</tr>
<tr>
<td>name</td>
<td>the name of the protection group</td>
</tr>
<tr>
<td>description</td>
<td>an optional description of the protection group</td>
</tr>
<tr>
<td>datastores</td>
<td>array of datastores to add to the new protection group</td>
</tr>
</tbody>
</table>

Returns CreateProtectionGroupTask to monitor the asynchronous operation of this method. For more information, see Create Protection Group Task.

**Faults**
- InvalidArgument
- InvalidType
- ReplicationProviderFault
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**Example: Example for CreateAbrProtectionGroup**

```java
ManagedObjectReference abrGroupRef = srmPortType.createAbrProtectionGroup(ManagedObjectReference _this,
    ManagedObjectReference location,
    String name,
    String description,
    List<ManagedObjectReference> datastores);
```

Where ManagedObjectReference _this = _protectionRef; where _protectionRef can be taken from:

```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
```

The following exceptions are presented by the CreateProtectionGroupTask instance that is returned by the CreateAbrProtectionGroup and CreateHbrProtectionGroup methods:
- ConnectionDownFault if the other site involved in the operation could not be contacted.
- DuplicateName if a group with this name already exists.
- StringArgumentTooLong if the size of either name or description in the settings parameter is too long.

**CreateHbrProtectionGroup**

Create a host based replication (vSphere replication) protection group using the provided VMs. This method does not automatically protect VMs on the storage array. Programs must call ProtectVms() separately for VMs on the storage array to be protected.

**Note** The protection group name cannot be the same as the folder in which it will be created.
Synopsis

CreateProtectionGroupTask createHbrProtectionGroup(Folder location, String name, @optional String description, vim.VirtualMachine[] vms)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>folder in which to create the protection group</td>
</tr>
<tr>
<td>name</td>
<td>the name of the protection group</td>
</tr>
<tr>
<td>description</td>
<td>an optional description of the protection group</td>
</tr>
<tr>
<td>vms</td>
<td>virtual machines to associate with the new protection group. The virtual machine list cannot be empty.</td>
</tr>
</tbody>
</table>

Returns CreateProtectionGroupTask to monitor the asynchronous operation of this method. For more information, see Create Protection Group Task.

Faults

- InternalError
- InvalidArgument
- InvalidType
- ReplicationProviderFault
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: CreateHbrProtectionGroup

```java
ManagedObjectReference hbrGroupRef = srmPortType.createHbrProtectionGroup(ManagedObjectReference _this, ManagedObjectReference location, String name, String description, List<ManagedObjectReference> vms);
```

Where ManagedObjectReference _this = _protectionRef;
where _protectionRef can be taken from:
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();

The following exceptions are presented by the CreateProtectionGroupTask instance that is returned by the CreateAbrProtectionGroup and CreateHbrProtectionGroup methods:

- ConnectionDownFault if the other site involved in the operation could not be contacted.
- DuplicateName if a group with this name already exists.
- StringArgumentTooLong if the size of either name or description in the settings parameter is too long.
CreateHbrProtectionGroup2

This method creates a new host based (that is vSphere replication) Protection Group using the provided virtual machines. The list of virtual machines can be empty.

**Note** The protection group name cannot be the same as the folder in which it will be created.

**Synopsis**

```
CreateProtectionGroupTask createHbrProtectionGroup2(
    drextapi.Folder location,
    String name,
    @optional String description,
    @optional vim.VirtualMachine[] vms)
```

createHbrProtectionGroup2 returns CreateProtectionGroupTask to monitor the asynchronous operation of this method. For more information, see Create Protection Group Task.

createHbrProtectionGroup2 method has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Folder in which to create the protection group.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the protection group.</td>
</tr>
<tr>
<td>description</td>
<td>An optional description of the protection group.</td>
</tr>
<tr>
<td>vms</td>
<td>VirtualMachines to associate with the new protection group. ProtectVms must be called for these VMs to be protected.</td>
</tr>
</tbody>
</table>

**Faults**

- InvalidArgument
- InternalError
- InvalidType
- ReplicationProviderFault
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

CreateVvolProtectionGroup

Creates a new VVol ProtectionGroup. Returns a task instance to monitor the asynchronous operation of this method.

**Synopsis**

```
CreateProtectionGroupTask createVvolProtectionGroup(
    drextapi.Folder location,
    String name,
```
createVvolProtectionGroup has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Folder in which the protection group should be created</td>
</tr>
<tr>
<td>name</td>
<td>Name of the protection group</td>
</tr>
<tr>
<td>description</td>
<td>Description of the protection group</td>
</tr>
<tr>
<td>replicationGroups</td>
<td>List of the replication groups to be configured for the protection group. Only the virtual machines replicated by these replication groups can be protected in this protection group.</td>
</tr>
</tbody>
</table>

**Faults**
- InternalError
- InvalidArgument
- ReplicationProviderFault
- RuntimeFault

See [Faults in Site Recovery Manager API](Site+Recovery+Manager+API) for more details.

**RemoveProtectionGroup**

The `removeProtectionGroup` method unprotects the VMs in the protection group and deletes the group.

**Synopsis**

```
RemoveProtectionGroupTask removeProtectionGroup(ProtectionGroup group)
```

The group parameter specifies the protection group that must be deleted.

RemoveProtectionGroupTask is a task object that contains information about the status of the operation. Site Recovery Manager Server retains the object for 30 minutes after the task finishes.

**Faults**
- ConnectionDownFault
- ProtectionGroupNotEmpty
- ReplicationProviderFault
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](Site+Recovery+Manager+API).
Protection Group Folder

This section presents methods to navigate folder hierarchy and retrieve specific protection groups.

ListChildProtectionGroupFolders

Returns the child Protection Group Folders located within this folder.

Synopsis

```java
ProtectionGroupFolder[] listChildProtectionGroupFolders()
```

ProtectionGroupFolder[] is the array of Protection Group Folders within this folder. Protection Group Folders for which the current session does not have the System.View privilege are removed from the result set. For more information, see Protection Group.

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

ListChildProtectionGroups

Returns the child Protection Groups located within this folder.

Synopsis

```java
ProtectionGroup[] listChildProtectionGroups()
```

ProtectionGroup[] is the array of Protection Groups within this folder. Protection Groups for which the current session does not have the System.View privilege are removed from the result set. For more information, see Protection Group.

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetProtectionGroup

Retrieves the protection group with the specified name, if any.

Synopsis

```java
ProtectionGroup getProtectionGroup(String name)
```

name is the name of the protection group.
Returns a specific ProtectionGroup with the given name. For more information, see Protection Group.

**Faults**

- ProtectionGroupNotFound
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

## Create Protection Group Task

This chapter presents methods to track progress and completion of create protection group calls.

### IsCreateProtectionGroupComplete

This function checks completeness of the operation. To get the result, see GetCreateProtectionGroupResult.

**Synopsis**

```csharp
boolean IsCreateProtectionGroupComplete()
```

True if this task has been completed.

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

### GetCreateProtectionGroupResult

This function gets the TaskInfo object containing the detailed results. To check completeness, see IsCreateProtectionGroupComplete.

**Synopsis**

```csharp
TaskInfo GetCreateProtectionGroupResult()
```

Returns the details results of this task.

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.
GetNewProtectionGroup

After calling CreateAbrProtectionGroup or CreateHbrProtectionGroup or createVvolProtectionGroup to create a protection group, this call makes a new one and fills in the protection group with the final result of the operation. To get the task results, see GetCreateProtectionGroupResult. To check status, see IsCreateProtectionGroupComplete.

Synopsis

```csharp
ProtectionGroup GetNewProtectionGroup()
```

Returns the newly created ProtectionGroup.

Faults

- InvalidState
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Protection Group

For array based replication, Site Recovery Manager organizes datastore groups to collect files associated with protected virtual machines. You configure array based replication by associating datastore groups with protection groups. All virtual machines in a datastore group replicate files together, and all virtual machines recover together.

Configure the host based replication (vSphere replication) for one virtual machine by associating it with a protection group, or you can configure multiple virtual machines by associating their folder or datacenter with a protection group.

You should configure vvol replication by associating replication groups with the protection groups.

Use the methods AssociateVms and UnassociateVms with the host based replication, but not with array based and vvol replications.

GetInfo

This method retrieves basic information about the specified protection group.

To get an SrmProtectionGroup managed object reference, see ListProtectionGroups.

Synopsis

```csharp
ProtectionGroup.Info getInfo()
```

ProtectionGroup.Info is information about the protection group. It has the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>protection group description</td>
</tr>
<tr>
<td>name</td>
<td>protection group name</td>
</tr>
<tr>
<td>type</td>
<td>either san for array based replication, vr for vSphere replication or vvol for vVol replication</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**Example: Example for GetInfo**

```java
SrmProtectionGroupInfo info = srmPortType.getInfo(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
- SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
- ManagedObjectReference _protectionRef = content.getProtection();
- List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
- _protectionGroupRef = groups.get(0);

**ProtectionGroupGetParentFolder**

Given a protection group, gets the parent folder.

**Synopsis**

```java
ProtectionGroupFolder ProtectionGroupGetParentFolder( )
```

ProtectionGroupFolder – extends Folder; can hold Protection Group and Protection Group Folder.

**Faults**

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**GetPeer**

Given an SrmProtectionGroup on the local site, this method retrieves the SrmProtectionGroup at the peer site.

**Synopsis**

```java
ProtectionGroup.Peer getPeer( )
```

ProtectionGroup.Peer is the peer protection group from the remote site.

Peer is the peer protection group object at the paired site. It has the following fields:
Field | Description
--- | ---
group | ProtectionGroup at that site.
state | Last known state of this protection group. If the connection between sites is down, this value might be out-of-date.

### Faults
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

### Example: Example for GetPeer

```java
SrmProtectionGroupPeer peerGroup = srmPortType.getPeer(ManagedObjectReference _this);
```

Where `ManagedObjectReference _this = _protectionGroupRef;`  
where `_protectionGroupRef` can be taken from:  
- `SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);`  
- `ManagedObjectReference _protectionRef = content.getProtection();`  
- `List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);`  
- `_protectionGroupRef = groups.get(0);`

### ListProtectedVms

Retrieves the list of virtual machines currently protected in the specified protection group, with information about their placeholder VM and protection state.

### Synopsis

```java
ProtectionGroup.ProtectedVm[] listProtectedVms( )
```

`ProtectedVm[]` is an array of `ProtectedVm` data objects with the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faults</td>
<td>any faults associated with this protected virtual machine</td>
</tr>
<tr>
<td>needsConfiguration</td>
<td>the protected virtual machine needs to be configured or repaired</td>
</tr>
<tr>
<td>peerProtectedVm</td>
<td>the protected virtual machine identifier on the remote site</td>
</tr>
<tr>
<td>peerState</td>
<td>the protection state on the remote site</td>
</tr>
<tr>
<td>protectedVm</td>
<td>the protected virtual machine identifier on the local site</td>
</tr>
<tr>
<td>state</td>
<td>the protection state of this particular virtual machine</td>
</tr>
<tr>
<td>vm</td>
<td>the locally protected virtual machine (this reference is valid after reprotect or revert operations)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>vmName</td>
<td>the name of the locally protected virtual machine.</td>
</tr>
<tr>
<td>drVmIdentity</td>
<td>A DR-service (SRM) specific globally unique identifier for the virtual machine associated with this ProtectedVm object and similarly for the virtual machine associated with this ProtectedVm object's peer (e.g. the recovered virtual machine). This identifier is generated by an SRM server; it does not necessarily correspond to any identifier in vCenter or any other service. The value of this property is the same in the peer ProtectedVm managed object. The value is immutable and is maintained on both sites after failover and reprotect + failback. After this virtual machine is unprotected (ProtectedVm object is removed) the value of this identity may be reused but only by a ProtectedVm instance that is subsequently protecting the same virtual machine. <strong>Note</strong> There is no guarantee that a virtual machine's drVmIdentity value remains the same even if it is unprotected and then reprotected. Maintaining this value is currently only a best-effort operation.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**Example: Example for ListProtectedVms**

```java
List < SrmProtectionGroupProtectedVm > protectedVms = srmPortType.listProtectedVms(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
  SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
  ManagedObjectReference _protectionRef = content.getProtection();
  List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
  _protectionGroupRef = groups.get(0);

**ListProtectedDatastores**

This method retrieves the list of datastores that are protected by the specified protection group. A datastore can be a VMFS volume, a NAS directory, or a local file system path.

**Synopsis**

```java
vim.Datastore[] listProtectedDatastores( )
```

Returns Datastore[] is an array of all Datastore objects protected by this protection group.
Faults

- RuntimeFault
- vmodl.fault.NotSupported if this protection group is not a SAN group.

See Faults in Site Recovery Manager API for more details.

Example: Example for ListProtectedDatastores

```java
List<ManagedObjectReference> datastores = srmPortType.listProtectedDatastores(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
List<ManagedObjectReference> groups = srmPortType.listProtectionGroups(_protectionRef);
    _protectionGroupRef = groups.get(0);
```

ListAssociatedVms

This method retrieves the virtual machines currently associated with a specified vSphere Replication protection group.

For the method to get a list of protection groups, see ListProtectionGroups.

Synopsis

```java
vim.VirtualMachine[] listAssociatedVms()
```

VirtualMachine[] is an array listing the associated virtual machines.

Faults

- RuntimeFault
- vmodl.fault.NotSupported, if this protection group is not a VR group.

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for ListAssociatedVms

```java
List<ManagedObjectReference> vms = srmPortType.listAssociatedVms(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
List<ManagedObjectReference> groups = srmPortType.listProtectionGroups(_protectionRef);
    _protectionGroupRef = groups.get(0);
```
GetProtectionState

Gets current state of the specified protection group. Not to be confused with GetProtectionStatus which returns a virtual machine’s (un)protect status, not the state of an entire protection group.

Synopsis

```
ProtectionGroup.ProtectionState getProtectionState()
```

ProtectionState is an enumeration for the protection group state:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>failedOver</td>
<td>the protection group has been failed over to the remote site</td>
</tr>
<tr>
<td>partiallyRecovered</td>
<td>the protection group is partially recovered</td>
</tr>
<tr>
<td>ready</td>
<td>the protection group is in a ready state</td>
</tr>
<tr>
<td>recovered</td>
<td>the protection group has been recovered</td>
</tr>
<tr>
<td>recovering</td>
<td>the protection group is in the process of being recovered</td>
</tr>
<tr>
<td>shadowing</td>
<td>this protection group is shadowing the remote site group that is in a ready state</td>
</tr>
<tr>
<td>testing</td>
<td>the protection group is currently being tested</td>
</tr>
</tbody>
</table>

**Note**  While these states share a common type, they are specific to whether this Group is the Protection Group itself or the mirror of the group on the remote site. The Protection Group States are ready and failedOver. The Mirror States partiallyRecovered, recovering, testing, shadowing, and recovered.

Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

Example: Example for GetProtectionState

```
SrmProtectionGroupProtectionState state = srmPortType.getProtectionState(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:

```
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
```

ProtectionGroupListRecoveryPlans

This method retrieves a list of all the recovery plans that this protection group is a member of.
Synopsis

RecoveryPlan[] ProtectionGroupListRecoveryPlans(

RecoveryPlan[] is an array of all the Recovery Plans that this protection group belongs to. For more information, see Recovery Plan.

Fault

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Example for ProtectionGroupListRecoveryPlans

List < ManagedObjectReference > plans = srmPortType.protectionGroupListRecoveryPlans(ManagedObjectReference _this);

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
  SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
  ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);

ProtectionGroupQueryVmProtection

Determine whether the specified virtual machines are currently protected, or can be protected. To protect a Virtual Machine, its folder, resource pool, and network must be mapped from the protected site to the recovery site.

To get a list of currently configured mappings, see ListInventoryMappings. You can also query replicated datastores with ListReplicatedDatastores.

Synopsis

ProtectionGroup.VmProtectionInfo[] ProtectionGroupQueryVmProtection(vim.VirtualMachine[] vms)

vms[] is an array of managed object references to VirtualMachine objects.

VmProtectionInfo[]is an array of VmProtectionInfo data objects with the following fields:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faults</td>
<td>any faults encountered while processing queryVmProtection for this virtual machine</td>
</tr>
<tr>
<td>peerProtectedVm</td>
<td>the protected virtual machine identifier on the remote site</td>
</tr>
<tr>
<td>protectedVm</td>
<td>the protected virtual machine identifier on the local site</td>
</tr>
<tr>
<td>protectionGroup</td>
<td>the group this virtual machine is a member of, if it is protected</td>
</tr>
<tr>
<td>Fields</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>protectionGroupName</td>
<td>the name of this virtual machine’s protection group, if it is protected</td>
</tr>
<tr>
<td>recoveryPlanNames</td>
<td>the name(s) of any recovery plans the virtual machine will be recovered in</td>
</tr>
<tr>
<td>recoveryPlans</td>
<td>any recovery plans the virtual machine will be recovered in</td>
</tr>
<tr>
<td>status</td>
<td>the current protection status of the virtual machine</td>
</tr>
<tr>
<td>vm</td>
<td>the virtual machine for which protection status is being returned</td>
</tr>
</tbody>
</table>

### Faults

- **RuntimeFault**

See [Faults in Site Recovery Manager API](#) for more details.

### Example: Example for ProtectionGroupQueryVmProtection

```java
List<SrmProtectionGroupVmProtectionInfo> protection = srmPortType.protectionGroupQueryVmProtection(
    ManagedObjectReference _this,
    List<ManagedObjectReference> vms);

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
    List<ManagedObjectReference> groups = srmPortType.listProtectionGroups(_protectionRef);
    _protectionGroupRef = groups.get(0);
```

### ProtectVms

This method adds virtual machines to a protection group.

With array-based replication, the protection group is determined by datastore location of the virtual machines. With host-based replication (vSphere replication), you can use the `AssociateVms` method to place virtual machines into a protection group. With vVol replication, the protection group is determined by replication groups of the virtual machines. To protect a Virtual Machine, its folder, resource pool, and network must be mapped from the protected site to the recovery site. To get a list of currently configured mappings, see `ListInventoryMappings`. If a few or all the VMs in the list are already protected, then the operation succeeds.

### Synopsis

```java
ProtectionTask protectVms(ProtectionGroup.VmProtectionSpec[] vms)
```

vms[] is a list of virtual machines to protect. If some of the virtual machines from the list or all of them are already protected, the operation succeeds.
VmProtectionSpec is a spec describing how a virtual machine to be protected. It has the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@optional VirtualMachine vm</td>
<td>The virtual machine to be protected</td>
</tr>
<tr>
<td>VmRecoverySpec recoverySpec</td>
<td>Information relevant at the recovery site to which the VM will be protected. Allows configuration of protection per VM instead of using global inventory mappings.</td>
</tr>
</tbody>
</table>

VmRecoverySpec is a spec describing virtual machine recovery locations. It has the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@optional PlaceholderVmLocation placeholderVmLocation</td>
<td>Location in which to create the placeholder VM.</td>
</tr>
<tr>
<td>@optional RecoveryLocationSettings recoveryLocationSettings</td>
<td>Recovery location settings.</td>
</tr>
</tbody>
</table>

PlaceholderVmLocation is a data object. It contains information about where a placeholder VM should be created. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folder</td>
<td>Folder in which the placeholder VM should be created. If unset, the inventory mapper will be queried for a suitable location.</td>
</tr>
<tr>
<td>hostSystem</td>
<td>Host in which the placeholder VM should be created. If unset, SRM will attempt to pick one based on the resource pool mapping (this works only if the resource pool unambiguously designates a single host, or if it designates a DRS cluster).</td>
</tr>
<tr>
<td>resourcePool</td>
<td>Resource pool in which the placeholder VM should be created. If unset, the inventory mapper will be queried for a suitable location.</td>
</tr>
<tr>
<td>placeholderDatastore</td>
<td>Datastore in which the placeholder VM should be created. If unset, the placeholder datastore manager will be queried for a suitable location.</td>
</tr>
</tbody>
</table>

RecoveryLocationSettings is a spec that provides user-editable settings regarding where to find virtual machine components at recovery time. It has the following properties:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@optional DeviceInfo[] protectedDevices</td>
<td>Information about devices for which the user has supplied recovery-time information.</td>
</tr>
<tr>
<td>@optional DeviceInfo[] excludedDevices</td>
<td>Information about devices for which the user does not want to be present in the recovered VM. If autoExcludeMediaDevices advance settings is enabled, the list includes the media devices that are auto-excluded by SRM.</td>
</tr>
<tr>
<td>@optional String changeVersion</td>
<td>Change version control. When reconfiguring an existing settings this value must be set and must match the most recent value. This means that first #getRecoveryLocationSettings should be called. Then its result should be updated and passed to #reconfigureRecoveryLocationSettings. For newly protected VMs, leave it unset.</td>
</tr>
</tbody>
</table>

DeviceInfo is a data object with the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Device key</td>
</tr>
</tbody>
</table>

NetworkDeviceInfo extends DeviceInfo and has the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vim.Network network</td>
<td>Reference to a recovery network managed object to which to attach the NIC device.</td>
</tr>
</tbody>
</table>

FileDeviceInfo extends DeviceInfo and has the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@optional vim.Datastore datastore</td>
<td>Reference to the datastore the directory is located on. If the directory is not located on a datastore, this is omitted.</td>
</tr>
<tr>
<td>path</td>
<td>Directory path.</td>
</tr>
<tr>
<td>fileName</td>
<td>Name of the file under the directory.</td>
</tr>
</tbody>
</table>

NetworkDeviceInfo and FileDeviceInfo should be used to construct RecoveryLocationSettings.

ProtectionTask is the task object to monitor the status of requested virtual machines. For more information, see Protection Task.

If a task fails, its error field may contain one of the following:

- drextapi.fault.ConnectionDownFault - if the remote SRM could not be reached.
- drextapi.fault.CannotProtectFTSecondaryVm - if the VM is a fault tolerance secondary VM.
- drextapi.fault.DeviceBackingConflict - if the caller specified a device locator, or explicitly excluded, a device which the provider would like to protect.
- drextapi.fault.DevicesNotResolved - if any of the VM's devices were neither protected by the provider, nor was a backing locator provided by the caller, nor was the device explicitly excluded.

- drextapi.fault.InsufficientLicensesFault - if there were not enough licenses available to protect the VMs.

- drextapi.InvalidState - if the group we are trying to protect into is not in the 'ProtectedVm.State#active active' state.

- drextapi.fault.ProductionVmDeleted - if the VM to be protected did not exist on VC.

- drextapi.fault.ReplicationProviderFault - if the replication provider rejected the operation.

- drextapi.fault.VmAlreadyProtectedEx - if the VM was already protected in another group.

- vim.fault.ConcurrentAccess - if the group was modified during the operation.

**Faults**

- InvalidArgument - If the list of virtual machines is empty or null.

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Example: Example for ProtectVms**

```java
ManagedObjectReference taskRef = srmPortType.protectVms(
    ManagedObjectReference _this,
    List < SrmProtectionGroupVmProtectionSpec > vms);
```

Where `ManagedObjectReference _this = _protectionGroupRef;`

`where _protectionGroupRef can be taken from:`

```java
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _protectionRef = content.getProtection();
    List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
    _protectionGroupRef = groups.get(0);
```

**UnprotectVms**

This method removes virtual machines from their protection group. With an array-based replication, the protection group is determined by datastore location of the virtual machines. With vSphere Replication, you must also **UnassociateVms** from the protection group. With the vVol replication, the protection group is determined by replication groups of the virtual machines.

**Synopsis**

```java
ProtectionTask unregisterVms(vim.VirtualMachine[] vms)
```

`vms[]` is an array Virtual Machine objects not to protect.

**Protection Task** is the task object to monitor for status of the requested virtual machines.
Faults

- InvalidArgument - If the list of virtual machines is empty or null.
- InvalidState, if a specified VM was not being protected.
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Example for UnprotectVms

```java
ManagedObjectReference taskRef = srmPortType.unprotectVms(
    ManagedObjectReference _this,
    List < ManagedObjectReference > vms);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
- SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
- ManagedObjectReference _protectionRef = content.getProtection();
- List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
- _protectionGroupRef = groups.get(0);

AssociateVms

This method associates one or more virtual machines with a vSphere Replication (VR) protection group. Before you can protect a virtual machine, it must first be associated with a protection group.

Synopsis

```java
void associateVms(vim.VirtualMachine[] vms)
```

vms[] is an array of Virtual Machine objects to associate with.

Faults

- InvalidArgument - If the list of virtual machines is empty or null.
- InvalidState, if a specified VM was already associated with another group.
- RuntimeFault
- vim.fault.ConcurrentAccess, if another operation has modified the object and the change version no longer matches.
- vmodl.fault.NotSupported, if this protection group is not a VR group.

See Faults in Site Recovery Manager API for more details.
Example: Example for AssociateVms

```java
srmPortType.associateVms(
    ManagedObjectReference _this,
    List < ManagedObjectReference > vms);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
  SrmserviceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
  ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);

UnassociateVms

This method removes the association of one or more virtual machines from a specified vSphere Replication (VR) protection group. Once a virtual machine is unassociated, it can no longer be protected.

Synopsis

```java
void unassociateVms(vim.VirtualMachine[] vms)
```

vms[] is an array of Virtual Machine objects to disassociate from.

Faults

- InvalidArgument - If the list of virtual machines is empty or null.
- InvalidState, if a specified VM was already associated with another group.
- RuntimeFault
- vim.fault.ConcurrentAccess, if another operation has modified the object and the change version no longer matches.
- vmodl.fault.NotSupported, if this protection group is not a VR group.

See Faults in Site Recovery Manager API for more details.

Example: Example for UnassociateVms

```java
srmPortType.unassociateVms(
    ManagedObjectReference _this,
    List < ManagedObjectReference > vms);
```

Where ManagedObjectReference _this = _protectionGroupRef;
where _protectionGroupRef can be taken from:
  SrmserviceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
  ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
CheckConfigured

The `checkConfigured` method checks the protection group for not configured VMs, configuration issues detected since the group was created or modified and protected virtual machines that needs configuration (if Placeholder VM needs repair or there are unresolved devices).

**Synopsis**

```java
boolean checkConfigured()
```

The method returns true if the protection group is configured and you can use the group.

**Faults**

- `InvalidState` is thrown if the group is not on the protected site and cannot get information about the remote object.
- `RuntimeFault`
- `vmodl.fault.NotSupported` is thrown if the group is not a VR, SAN or vVol group.

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

ProtectionGroupGetOperationalLocation

The `ProtectionGroupGetOperationalLocation` method returns the effective location of the protection group for the purposes of determining when various operations should be run.

**Synopsis**

```java
String ProtectionGroupGetOperationalLocation()
```

**Faults**

`RuntimeFault`

See [Faults in Site Recovery Manager API](#) for more details.

AddDatastores

Adds datastores to the protection group. Additionally, the virtual machines on these datastores can be protected by the protection group. This can be done by calling `protectVms` method from this interface.

**Synopsis**

```java
void addDatastores(@optional Datastore[] datastores)
```

datastores is the list of datastores that will be added to the protection group.
Faults
- InvalidArgument
- InvalidState
- NotSupported
- ReplicationProviderFault
- RuntimeFault
- vim.fault.ConcurrentAccess

See Faults in Site Recovery Manager API for more details.

RemoveDatastores

Removes datastores from the protection group. Virtual machines on the removed datastores are no longer protected by the protection group.

Synopsis

```java
void removeDatastores(@optional Datastore[] datastores)
```

datastores is the list of datastores that will be removed from the protection group.

Faults
- InvalidArgument
- InvalidState
- NotSupported
- ReplicationProviderFault
- RuntimeFault
- vim.fault.ConcurrentAccess

See Faults in Site Recovery Manager API for more details.

ReconfigureVvolProtectionGroup

Reconfigures settings for this group. For a vVol ProtectionGroup, this method can reconfigure the name, description, and replication groups.

Synopsis

```java
void reconfigureVvolProtectionGroup(
    @optional String name,
    @optional String description,
    @optional ReplicationGroupId[] replicationGroups)
```
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>New name for this protection group.</td>
</tr>
<tr>
<td>description</td>
<td>Sets a new description for this protection group.</td>
</tr>
<tr>
<td>replicationGroups</td>
<td>Replication groups that will be configured for this group.</td>
</tr>
</tbody>
</table>

### Faults

- ConcurrentAccess
- DuplicateName
- InvalidArgument
- InvalidState
- NotSupported
- ReplicationProviderFault
- RuntimeFault
- StringArgumentTooLong

See [Faults in Site Recovery Manager API](#) for more details.

### GetVvolGroupDetails

Gets vVol specific details for this protection group.

### Synopsis

```python
drextapi.vvol.GroupDetails getVvolGroupDetails()
```

GroupDetails is the VvolProvider specific details for a protection group. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>Identifier of the fault domain operated by this protection group. The VVOL protection group is limited to protecting VMs that belong to the same fault domain. The fault domain is determined by the replication groups configured for this protection group.</td>
</tr>
</tbody>
</table>
| ReplicationGroupInfo[]
  replicationGroups            | Source VVOL replication groups for this protection group. Both protection and recovery site report the same replication groups. |
| VmInfo[]
  protectedVms                 | Info about the protected VMs in this protection group including VMs replication groups association. |
| VmInfo[]
  unprotectedVms             | The list of VMs that are not protected and are replicated by a replication group that is part of this protection group. Available only from protection site, i.e. @primary == true |
| primary                       | Flag that is set if the protection group is primary. |
VmInfo is the information about a local vVol replicated vim.VirtualMachine. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The vim.VirtualMachine object at the local site.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the virtual machine.</td>
</tr>
<tr>
<td>ReplicationGroupInfo[] replicationGroups</td>
<td>Replication groups for this VM. Virtual machines sharing the same replication groups belong to the same consistency group. They will be added or removed from a protection group together.</td>
</tr>
</tbody>
</table>

ReplicationGroupInfo[] is the information about a vVol replication group. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicationGroup</td>
<td>Replication group ID.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the replication group. May be unset if not available.</td>
</tr>
</tbody>
</table>

FaultInfo is the warning and error information object. Both VmInfo and ReplicationGroupInfo inherits FaultInfo. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>warnings</td>
<td>Warnings associated with this object.</td>
</tr>
<tr>
<td>name</td>
<td>Errors associated with this object.</td>
</tr>
</tbody>
</table>

**Fault**
- NotSupported
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**MoveGroup**

This method moves specified ProtectionGroup to a different folder.

**Synopsis**

```java
@task void moveGroup(drextapi.Folderdestination);
```

moveGroup returns a task instance to monitor the asynchronous operation of this method. It has the following parameters:
Field | Description
--- | ---
destination | Folder which will become the new parent folder of this group. For more information, see SRM Folder.

If a task fails, its error field may contain one of the following:

- drextapi.fault.DuplicateName - A ProtectionGroup with the same name already exists within the destination folder.
- vim.fault.NotSupported - if the ProtectionGroup is being moved into a folder whose childType() property is not set to the appropriate value. For example, a ProtectionGroup cannot be moved into a folder whose ChildType property value does not contain "ProtectionGroup".
- drextapi.fault.ConnectionDownFault - if the sites are not connected.

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

**GetPlaceholderVmInfo**

This method returns information for the placeholder VM for the specified protected VM.

**Synopsis**

```java
PlaceholderVmInfo getPlaceholderVmInfo(ProtectedVm protectedVm)
```

ProtectedVm is an array of ProtectedVm data objects. For more information, see ListProtectedVms.

PlaceholderVmInfo is a data object. It provides information about the inventory location of the placeholder vim.VirtualMachine. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm</td>
<td>Placeholder VM. This can be unset if the placeholderVm has been deleted or has not been created succesfully.</td>
</tr>
<tr>
<td>folder</td>
<td>Placeholder VM folder</td>
</tr>
<tr>
<td>computeResource</td>
<td>Placeholder VM ComputeResource. Not set if the VM is a template.</td>
</tr>
<tr>
<td>resourcePool</td>
<td>Placeholder VM ResourcePool. Not set if the VM is a template.</td>
</tr>
<tr>
<td>host</td>
<td>Placeholder VM host.</td>
</tr>
<tr>
<td>datacenter</td>
<td>Placeholder VM datacenter</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>placeholderCreationFault</td>
<td>Fault from the most recent placeholder creation operation at the local site, if that operation failed. Otherwise unset.</td>
</tr>
<tr>
<td>repairNeeded</td>
<td>Set to true if the placeholder VM needs to be repaired. false otherwise.</td>
</tr>
</tbody>
</table>

**Faults**

- InvalidArgument
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

**RecreatePlaceholder**

Recreates a placeholder VM.

This method is called when the placeholder needs to be recreated due to one of these reasons:

- Placeholder creation failed.
- Placeholder was deleted.
- Placeholder inventory was lost or needs to be reentered - one use case for this is when production vm was a template but then gets converted to a VM.

This method can be called only on the recovery site. It does not need primary site to be up for succesful completion. This method requires `Resource.com.vmware.vcDr.RecoveryUse` on the host, resource pool, datastore, and folder.

**Synopsis**

```java
@task void recreatePlaceholder(ProtectedVm protectedVm, PlaceholderVmLocation placeholderVmLocation)
```

recreatePlaceholder has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protectedVm</td>
<td>ProtectedVm for which the placeholder VM will be recreated. For more information, see <code>ListProtectedVms</code>.</td>
</tr>
<tr>
<td>placeholderVmLocation</td>
<td>New location for the placeholder VM. For more information, see <code>ProtectVms</code>.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain the following:

- `dextapi.fault.InvalidState` - if repair is called from the wrong site.

**Faults**

- InvalidArgument
- RuntimeFault
GetRecoveryLocationSettings

This method returns the recovery location settings for the specified protected VM.

Synopsis

RecoveryLocationSettings getRecoveryLocationSettings(ProtectedVm protectedVm)

RecoveryLocationSettings is a data object. It has user-editable settings regarding where to find the VM components during recovery time. For more information, see ProtectVms.

ProtectedVm is an array of protected VM data objects. For more information, see ListProtectedVms.

Faults

- InvalidArgument
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

ReconfigureRecoveryLocationSettings

Reconfigures the recovery location settings for the specified protected VM. This method should be invoked on the protection site only.

Synopsis

@task void reconfigureRecoveryLocationSettings(ProtectedVm protectedVm, RecoveryLocationSettings recoveryLocationSettings)

reconfigureRecoveryLocationSettings has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protectedVm</td>
<td>The protected VM which settings will be updated. For more information, see ProtectVms.</td>
</tr>
<tr>
<td>recoveryLocationSettings</td>
<td>The new settings to apply. For more information, see ProtectVms.</td>
</tr>
</tbody>
</table>

ReconfigureRecoveryLocationSettings returns the drextapi.Task. For more information, see SmExtApiTask.

If a task fails, its error field may contain one of the following:

- vim.fault.ConcurrentAccess if another operation has modified the object and the change version no longer matches.
- drextapi.fault.InvalidState if the state of the protected VM is not 'active'.
- drextapi.fault.DeviceBackingConflict if the caller specified a device locator, or explicitly excluded, a device which the provider would like to protect.
Faults

- InvalidArgument
- InvalidState
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Protection Task

A task returned by ProtectVms or UnprotectVms acquires the final status of an operation upon completion. While the task is running, partial results may be returned. Once the task has been completed, the object will be removed from the server after 30 minutes.

GetProtectionStatus

This method gets the virtual machine protection status after completion of ProtectVms or UnprotectVms.

Synopsis

```
ProtectionGroup.VmProtectionInfo[] getProtectionStatus()
```

VmProtectionInfo[] – the completed protection status of VMs that were requested to be protected or unprotected. For more information about VmProtectionInfo, see ProtectionGroupQueryVmProtection.

The VmProtectionInfo.ProtectionStatus has the following fields:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canBeProtected</td>
<td>the VM is able to be protected, but is not currently</td>
</tr>
<tr>
<td>canNotBeProtected</td>
<td>the VM is not able to be protected</td>
</tr>
<tr>
<td>isProtected</td>
<td>the VM is already protected</td>
</tr>
<tr>
<td>needsConfiguration</td>
<td>the VM must be configured or repaired before it may be protected. Please check the faults property for information about any additional prerequisites.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.
Example: Example for GetProtectionStatus

```java
List < SrmProtectionGroupVmProtectionInfo > protectionStatus =
srmPortType.getProtectionStatus(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionTaskRef;
where _protectionTaskRef can be taken from:
```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
ManagedObjectReference _protectionTaskRef = srmPortType.protectVms(
    _protectionGroupRef,
    _vms);
```

GetTasks

This method retrieves task information from the vCenter Server after a ProtectVms or UnprotectVms request, which both take some time to complete.

Synopsis

```java
ProtectionTask.VmTask[] getTasks()
```

VmTask[] is an array of monitorable task information keyed by Virtual Machine, containing:

- task – managed object reference to a task on the Site Recovery Manager server.
- vm – managed object reference to a VirtualMachine.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for GetTasks

```java
List < SrmProtectionTaskVmTask > tasks = srmPortType.getTasks(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionTaskRef;
where _protectionTaskRef can be taken from:
```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
ManagedObjectReference _protectionTaskRef = srmPortType.protectVms(
    _protectionGroupRef,
    _vms);
```
IsComplete
This method checks whether the protection task has completed.

Synopsis

```java
boolean isComplete( )
```

Returns true if the task has completed, false if not.

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for IsComplete

```java
boolean isComplete = srmPortType.isComplete(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionTaskRef;
where _protectionTaskRef can be taken from:
```
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
ManagedObjectReference _protectionTaskRef = srmPortType.protectVms(_protectionGroupRef, _vms);
```

GetResult
This method gets detailed results of the completed protection task.

Synopsis

```java
vim.TaskInfo[] getResult( )
```

TaskInfo is a data object that contains all information about a task. For more information, see TaskInfo in the Site Recovery Manager API Reference Guide.

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.
Example: Example for GetResult

```java
List < TaskInfo > taskInfo = srmPortType.getResult(ManagedObjectReference _this);
```

Where ManagedObjectReference _this = _protectionTaskRef;

where _protectionTaskRef can be taken from:

```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _protectionRef = content.getProtection();
List < ManagedObjectReference > groups = srmPortType.listProtectionGroups(_protectionRef);
_protectionGroupRef = groups.get(0);
ManagedObjectReference _protectionTaskRef = srmPortType.protectVms(_protectionGroupRef, _vms);
```

Recovery

This section covers the external interface to Site Recovery Manager - Recovery. This interface works only on the locally connected site except where specified.

ListPlans

This method retrieves all the recovery plans for this Site Recovery Manager server. Once you have a list of recovery plans, you can retrieve information about each plan.

Synopsis

```java
RecoveryPlan[] listPlans()
```

RecoveryPlan[] is a list of Recovery Plans, including plan information, peer recovery plan, recovery mode, recovery plan location, recovery prompt, and recovery state. For more information, see Recovery Plan.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetHistory

This method retrieves the history of a given recovery plan.

Synopsis

```java
RecoveryPlanHistory getHistory(RecoveryPlan plan)
```

plan is the Recovery Plan of interest.

RecoveryPlanHistory is the history of the given Recovery Plan.
Faults

- RecoveryPlanNotFoundException
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Example for GetHistory

```java
ManagedObjectReference history = srmPortType.getHistory(
    ManagedObjectReference _this,
    ManagedObjectReference plan);
```

Where ManagedObjectReference _this = _recoveryRef;
where _recoveryRef can be taken from:
```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
```

GetRecoveryPlanRootFolder

Gets a reference to the top level container (the root folder) for recovery plans.

Synopsis

```java
RecoveryPlanFolder getRecoveryPlanRootFolder()
```

RecoveryPlanFolder – a Site Recovery Manager folder that holds Recovery Plans and Recovery Plan Folders. For more information, see Recovery Plan Folder.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

CreateRecoveryPlan

The createRecoveryPlan method creates a recovery plan. You call the method by passing the name and folder of the plan, and the protection group(s) that must be included in the plan.

Synopsis

```java
CreateRecoveryPlanTask createRecoveryPlan(
    String name,
    drextapi.Folder folder,
    ProtectionGroup[] groups,
    @optional String description,
    @optional TestNetworkMapping[] mapping)
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the name of the plan. Must be unique in the parent folder, and contain between 1 and 80 characters.</td>
</tr>
<tr>
<td>folder</td>
<td>Specifies the folder where the plan is created.</td>
</tr>
<tr>
<td>groups</td>
<td>Specifies the protection groups that are added to the recovery plan.</td>
</tr>
<tr>
<td>description</td>
<td>Optional parameter. Specifies the recovery plan description. Should not contain more than 4096 characters.</td>
</tr>
<tr>
<td>mapping</td>
<td>Optional parameter. Specifies the test network mappings. For more information, see <code>GetTestNetworkMappings</code>.</td>
</tr>
</tbody>
</table>

CreateRecoveryPlanTask is a task object that contains information about the status of the operation. Site Recovery Manager Server retains the object for 30 minutes after the task finishes.

IsCreateRecoveryPlanComplete returns true if the task for creating a recovery plan is complete.

GetCreateRecoveryPlanFailure returns the failure during the operation for creating a recovery plan. If you pass a name of a plan that exists, the failure is DuplicateNames.

If the operation is successful, the GetNewRecoveryPlan returns the created recovery plan.

**Note** The recovery plan name can not be the same as the folder in which it will be created.

If a task fails, its error field may contain one of the following:

- DuplicateNames if a plan with this name already exists.
- InvalidArgument if the name parameter is empty string
- InvalidType if folder parameter isn't meant to hold a recovery plans.
- InvalidPrimaryNetwork if mapping parameter contains TestNetworkMapping with an invalid network on primary site. For example, uplink DVPortgroup.
- InvalidSecondaryNetwork if mapping parameter contains TestNetworkMapping with an invalid network on secondary site. For example uplink DVPortgroup.
- NoPermission if user doesn't have VcDr.RecoveryProfile.com.vmware.vcDr.Create privilege on the specified folder or VcDr.ProtectionProfile.com.vmware.vcDr.AssignToRecoveryPlan privilege on all protection groups in the plan.
- StringArgumentTooLong if the size of either name or description of the plan is too long.
- UnknownPrimaryNetwork if mapping parameter contains TestNetworkMapping with a network that does not exist on primary site.
- UnknownSecondaryNetwork if mapping contains TestNetworkMapping with a network that does not exist on secondary site.

**Faults**

- DirectionError
- InvalidArgument
- ProtectionGroupNotFound
- RemoteSiteNotEnabled
- RuntimeFault
- StringArgumentTooLong

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

**DeleteRecoveryPlan**

The `deleteRecoveryPlan` method deletes the recovery plan that is passed as parameter.

**Synopsis**

```java
DeleteRecoveryPlanTask deleteRecoveryPlan(RecoveryPlan plan)
```

The `plan` parameter specifies the recovery plan that must be deleted.

`DeleteRecoveryPlanTask` is a `Task` object that contains information about the status of the operation. Site Recovery Manager Server retains the object for 30 minutes after the task finishes.

The `IsDeleteRecoveryPlanComplete` returns true if the task is complete.

The `GetDeleteRecoveryPlanFailure` returns the failure that occurs during the delete operation. If the plan or its peer is not in a valid state, the task fails with `InvalidState`.

**Faults**

RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

**MovePlan**

This method moves the `RecoveryPlan` to a different folder.

**Synopsis**

```java
@task void movePlan(RecoveryPlan recoveryPlan, drextapi.Folder destination)
```

`movePlan` returns a task instance to monitor the asynchronous operation of this method. This method has the following parameters:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recoveryPlan</td>
<td>Recovery plan to move. For more information, see Recovery Plan.</td>
</tr>
<tr>
<td>destination</td>
<td>Folder which will become the new parent folder of this plan. For more information, see SRM Folder.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain one of the following:

- `drextpi.fault.DuplicateName`: A RecoveryPlan with the same name already exists within the destination folder.
- `vim.fault.NotSupported`: If the RecoveryPlan is being moved into a folder whose `childType()` property is not set to the appropriate value. For example, a RecoveryPlan cannot be moved into a folder whose `ChildType` property value does not contain "RecoveryPlan".
- `drextpi.fault.ConnectionDownFault`: If the sites are not connected.

### Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

## Recovery Plan Folder

This section presents methods to traverse the folder hierarchy for Recovery Plans.

### ListChildRecoveryPlanFolders

Returns the child Recovery Plan Folders located in this folder.

#### Synopsis

```java
RecoveryPlanFolder[] listChildRecoveryPlanFolders()
```

`RecoveryPlanFolder[]` is the array of sub-folders within this folder. If the current session does not have the `System.View` privilege for a `RecoveryPlanFolder`, it is removed from the result set.

#### Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

### Example: Example for ListChildRecoveryPlanFolders

```java
List <ManagedObjectReference> childFolders =
    srmPortType.listChildRecoveryPlanFolders(ManagedObjectReference _this);
Where ManagedObjectReference _this = _recoveryPlanRootFolderRef;
```
where \_recoveryPlanRootFolderRef can be taken from:

\[
\text{SrmServiceInstanceContent content} = \_\text{srmPortType.retrieveContent(svcRef);} \\
\text{ManagedObjectReference \_recoveryRef} = \text{content.getRecovery();} \\
\text{ManagedObjectReference \_recoveryPlanRootFolderRef} = \\
\text{srmPortType.getRecoveryPlanRootFolder(\_recoveryRef);} \\
\]

**ListChildRecoveryPlans**

Returns an array of RecoveryPlan objects located within this folder.

**Synopsis**

\[
\text{RecoveryPlan[]} \text{ listChildRecoveryPlans( )} \\
\]

RecoveryPlan[] is the array of Recovery Plans within this folder. If the current session does not have the System.View privilege for a RecoveryPlan, it is removed from the result set. For more information, see Recovery Plan.

**Faults**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

**Example: Example for ListChildRecoveryPlans**

\[
\text{List<ManagedObjectReference> childPlans} = \\
\text{srmPortType.listChildRecoveryPlans(ManagedObjectReference \_this);} \\
\]

Where ManagedObjectReference \_this = \_recoveryPlanRootFolderRef;

where \_recoveryPlanRootFolderRef can be taken from:

\[
\text{SrmServiceInstanceContent content} = \_\text{srmPortType.retrieveContent(svcRef);} \\
\text{ManagedObjectReference \_recoveryRef} = \text{content.getRecovery();} \\
\text{ManagedObjectReference \_recoveryPlanRootFolderRef} = \\
\text{srmPortType.getRecoveryPlanRootFolder(\_recoveryRef);} \\
\]

**GetRecoveryPlan**

Retrieves a specific recovery plan.

**Synopsis**

\[
\text{RecoveryPlan getRecoveryPlan(String name)} \\
\]

name is the name of a Recovery Plan.

RecoveryPlan is a Recovery Plan, which includes plan information, peer recovery plan, recovery mode, recovery plan location, recovery prompt, and recovery state. For more information, see Recovery Plan.
Faults

- RecoveryPlanNotFound
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Example for GetRecoveryPlan

```java
ManagedObjectReference plan = srmPortType.getRecoveryPlan(
    ManagedObjectReference _this,
    String name);

Where ManagedObjectReference _this = _recoveryPlanRootFolderRef;
where _recoveryPlanRootFolderRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _recoveryRef = content.getRecovery();
    ManagedObjectReference _recoveryPlanRootFolderRef =
        srmPortType.getRecoveryPlanRootFolder(_recoveryRef);
```

Recovery Plan

This section covers the interfaces to recovery plans.

RecoveryPlanGetInfo

This method retrieves status information about a given recovery plan, including the name of the recovery plan and its current state.

Synopsis

```java
RecoveryPlan.Info RecoveryPlanGetInfo( )
```

RecoveryPlan.Info is a data object that describes details of this recovery plan, including:

- **name** is the name of this recovery plan.
- **description** is a description of this recovery plan.
- **protectionGroups[]** is an array of protection groups that will be recovered as part of this plan.
- **state** – the state of this recovery plan, enumerated as:
  - **cancelling** – recovery plan is in the process of cancelling
  - **error** – recovery plan has errors
  - **failedOver** – recovery plan has failed over
  - **needsCleanup** – need to cleanup a test run
  - **needsFailover** – need to re-run recovery (failover)
  - **needsReprotect** – need to re-run reprotect
- needsRollback – need to re-run rollback
- prompting – recovery plan is running, but requires user-interaction before it may continue
- protecting – recovery plan is protecting to remote site, run peer recovery plan on remote site
- ready – recovery plan is not in a running state and may be run
- running – recovery plan is currently running

**Faughts**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Example: Examples for RecoveryPlanGetInfo**

```java
SrmRecoveryPlanInfo planInfo = srmPortType.recoveryPlanGetInfo(ManagedObjectReference _this);

// Where ManagedObjectReference _this = _recoveryPlan;
// where _recoveryPlan can be taken from:
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List<ManagedObjectReference> plans = srmPortType.listPlans(_recoveryRef);
ManagedObjectReference _recoveryPlan = plans.get(0);
```

The sample C# and Java code below combines ListPlans with RecoveryPlanGetInfo to retrieve a specified plan.

**C# sample code for recovery plan**

```csharp
ManagedObjectReference[] plans = _service.ListPlans(_sic.recovery); 
if (plans != null && plans.Length > 0) 
{
    for (int i = 0; i < plans.Length; ++i) 
    { 
        SrmRecoveryPlanInfo info = _service.RecoveryPlanGetInfo(plans[i]);
        Console.WriteLine("RecoveryPlan : " + info.name);
        if (info.name.Equals(planName))
        {
            Console.Write(" RecoveryPlan state : ");
            Console.WriteLine(info.state);
        }
    }
}
```

**Java sample code for recovery plan**

```java
private static void listPlans() throws Exception { List<ManagedObjectReference> plans = srmPort.listPlans(serviceContent.getRecovery());
if (plans != null && plans.size() > 0)
```
for (int i = 0; i < plans.size(); ++i) {
    SrmRecoveryPlanInfo info = srmPort.recoveryPlanGetInfo(plans.get(i));
    System.out.println("RecoveryPlan : " + info.getName());
    if (info.getName().equals(planName)) {
        System.out.print(" RecoveryPlan state : ");
        System.out.println(info.getState());
    }
}
}

RecoveryPlanGetPeer

This method retrieves a recovery plan peer, which is the plan at the paired site rather than at the local site.

Synopsis

RecoveryPlan.Peer RecoveryPlanGetPeer()

RecoveryPlan.Peer is the peer recovery plan at the paired site.

- plan references to the SrmRecoveryPlan managed object.
- state is the same enumeration as for RecoveryPlanGetInfo. For more information, see RecoveryPlanGetInfo.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for RecoveryPlanGetPeer

SrmRecoveryPlanPeer peerPlan = srmPortType.recoveryPlanGetPeer(ManagedObjectReference _this);

Where ManagedObjectReference _this = _recoveryPlan;
where _recoveryPlan can be taken from:
SrmServiceInstanceContent content = _srmpType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List<ManagedObjectReference> plans = srmPortType.listPlans(_recoveryRef);
ManagedObjectReference _recoveryPlan = plans.get(0);

Start

This method starts or reconfigures the given recovery plan, or tests and cleans it up, depending on the mode specified. This operation requires one of these privileges depending on recovery
mode, VcDr.RecoveryProfile.com.vmware.vcDr.Failover for a real recovery and VcDr.RecoveryProfile.com.vmware.vcDr.Run for a test recovery.

**Note** You must use the UI, not the API, to initiate forced failover. It requires complicated set-up and validation steps.

**Synopsis**

```java
void start(RecoveryPlan.RecoveryMode mode, @version5 @optional RecoveryOptions options)
```

- **mode** – one of the following recovery modes:
  - **test** – run a test failover to the peer (recovery) site, without halting the local (protected) site.
  - **cleanupTest** – after testing a recovery plan, cleans up all effects of the test operation.
  - **failover** – move to the peer (recovery) site. When all groups are moved the recovery plan is complete.
  - **reprotect** – the peer site becomes the protected site, and the local site becomes the recovery site.
  - **revert** - revert a recovery, abandoning all the VMs on the peer site and powering on the original VMs on the local site. This operation is not allowed unless all the replication groups are in the shadowing, recovered, or partially recovered state. If the sites are not connected, the peer VMs may be left running. In order to correct this situation, you might have to re-run revert after the sites are connected.
  - **migrate** - Migrate the recovery plan to the peer site. Once completed successfully, the plan should be deleted or reprotected. For a successful migration to occur, all the groups in the plan must be in the shadowing or recovered states.

- **options** specifies the recovery options. The `RecoveryOptions.syncData` is a boolean parameter and indicates whether to replicate the recent changes to the recovery site.

**Faults**

- **InvalidArgument**, if the recovery mode is not valid.
- **InvalidState**, if the recovery plan is not in the ready state.
- **RuntimeFault**

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Example: Example for Start**

```java
srmPortType.start(
    ManagedObjectReference _this,
    SrmRecoveryPlanRecoveryMode mode,
    SrmRecoveryOptions options);
```
Cancel

This method cancels this recovery plan. It can take some time to cancel a recovery plan depending on its state. This operation requires one of these privileges depending on recovery mode, VcDr.RecoveryProfile.com.vmware.vcDr.Failover for a real recovery and VcDr.RecoveryProfile.com.vmware.vcDr.Run for a test recovery.

Synopsis

void cancel()

Faults

- InvalidState, if the recovery plan cannot be canceled.
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for Cancel

srmPortType.cancel(ManagedObjectReference _this);

ListPrompts

This method lists the current prompts that are waiting on user input. Prompts appear in the order in which virtual machines are scheduled to power on.

When a prompt step is reached, the recovery plan remains in a waiting state until the user answers the prompt or a program calls AnswerPrompt.
Synopsis

```java
RecoveryPlan.RecoveryPrompt[] listPrompts()
```

RecoveryPrompt[] is an array of data objects containing the prompt and the key for responding to it. It has the following fields:

- **key** - Key for responding to the prompt
- **data** - Data about the prompt

**Faults**

- **InvalidState**, if the recovery plan is not running.
- **RuntimeFault**

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Example: Example for ListPrompts**

```java
List < SrmRecoveryPlanRecoveryPrompt > listPrompts = srmPortType.listPrompts(ManagedObjectReference _this);
```

Where `ManagedObjectReference _this = _recoveryPlan;`
where `_recoveryPlan` can be taken from:

```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(_recoveryRef);
ManagedObjectReference _recoveryPlan = plans.get(0);
```

**AnswerPrompt**

This method answers the current prompt being displayed in a recovery plan. The operation requires one of these privileges depending on recovery mode, `VcDr.RecoveryProfile.com.vmware.vcDr.Failover` for a real recovery and `VcDr.RecoveryProfile.com.vmware.vcDr.Run` for a test recovery.

**Synopsis**

```java
void answerPrompt(String key, boolean cancelVmRecovery, @optional String response)
```

key is a string with the key value from the recovery prompt.

cancelVmRecovery is true if you want to halt further processing on this virtual machine, false otherwise.

response is a response to the prompt that will be recorded.

**Faults**

- **InvalidState**, if the recovery plan is not running.
PromptNotFound, if no prompt with that key exists.

RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for AnswerPrompt

```java
srmPortType.answerPrompt(
    ManagedObjectReference _this,
    String key,
    boolean cancelVmRecovery,
    String response);
```

Where ManagedObjectReference _this = _recoveryPlan;
where _recoveryPlan can be taken from:
```
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(
    _recoveryRef);
ManagedObjectReference _recoveryPlan = plans.get(0);
```

RecoveryPlanGetParentFolder

Gets the parent folder (or root) for a recovery plan.

Synopsis

```
RecoveryPlanFolder RecoveryPlanGetParentFolder()
```

RecoveryPlanFolder is a Site Recovery Manager folder that can hold Recovery Plans and Recovery Plan Folders. For more information, see Recovery Plan Folder.

Faults

RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetRecoverySettings

Gets the recovery settings for the specified virtual machine.

Synopsis

```
RecoverySettings GetRecoverySettings (VirtualMachine vm)
```

vm is the Virtual Machine whose Recovery Settings are to be retrieved.
RecoverySettings – the VM recovery settings for presentation in the user interface, including:

- changeVersion – change version control. When reconfiguring an existing Protected workload this value must be set and must match the most recent value. For settings on newly added VMs, leave this unset.

- status – recovery status. This enumerates the following values:
  - ok - There are no recovery setting conditions to alert to the user.
  - syncConflict - Synchronization error. The settings are in a sync conflict with the remote site. Use the SRM user interface to clear the conflict.

- recoveryPriority – the recovery priority for this VM. This enumerates the following values:
  - highest
  - higher
  - normal
  - lower
  - lowest

- skipGuestShutdown – configure the shutdown behavior for this workload during real failover not to attempt a guest shutdown, even if VMware Tools are enabled.

- powerOffTimeoutSeconds – configure the timeout for guest shutdown operations for this VM.

- finalPowerState – final power state for this VM after recovery.

- localFaultToleranceState – configure FT override setting for this VM when it will be failed back.

- remoteFaultToleranceState – configure an FT override setting for this VM after recovery.

- powerOnTimeoutSeconds – configure the timeout for VMware Tools to respond with a heartbeat.

- powerOnDelaySeconds – configure the fixed time delay after power-on operations for this workload.

- Callout[] prePowerOnCallouts – before power-on Callouts (commands or prompts).

- Callout[] postPowerOnCallouts – after power-on Callouts.

- Callout - Base class for all Callouts (Commands or Prompts). It has the following fields:
  - description - Name/description of the Callout, for display purposes in UI. A short description or name of the Callout for display purposes in the UI.
  - uuid - UUID of the Callout, used by the UI for editing. This string should follow the definition of the string representation of a UUID as per RFC 4122. An example of a UUID is "08e3c162-9213-ff31-681b-ff6e35f2ac1b".
- **Prompt** - Specifies a Callout which pauses the execution of the recovery script and display a message until the user presses continue. It contains the `promptText` field. `promptText` is the text to display while running a recovery. This text will be displayed in the SRM UI and be placed in the SNMP Trap. `promptText` is a non-empty string. The maximum valid length is 4096 characters.

- **Command** - This class specifies a Command which runs during a Recovery, either on the server or on the recovered VM. If the command returns a non-zero value, then recovery indicates that an error has occurred. This Command can be inserted before or after a particular VM is powered on. If inserted after a particular VM is powered on, the Command may be run either on the SRM server machine or inside the recovered VM that was just powered on.

This class has the following fields:

- `command` - Command to run while running a recovery. Non-empty string. The maximum valid length is 4096 characters.

- `long timeout` - Time in seconds to wait until the command is completed. If the command has not completed when the timeout occurs, the child process will be killed.

- `boolean runInRecoveredVm` - Should command be run on SRM server machine or in recovered VM? This only applies to Per-VM Callout Commands set to run post-power-on of a recovered VM. Those Commands may be run either on the SRM machine, or inside the recovered VM.

- **VmIpCustomization** - Contains all data, needed to perform IP customization for a virtual machine, when recovering it to a given SRM site.

- `dependentVmIds` - Dependent VMs. This is a list of VM identities that must be powered on before this VM can be powered on. Dependencies are only valid within VMs of the same recovery priority. If there are dependent VMs that are not in the current plan and same recovery priority, they will be ignored. VM identity is available through `ProtectionGroup#ProtectedVm#drVmIdentity`.

`VmIpCustomization` contains the following data required for performing IP customization for a virtual machine, when recovering it to a given SRM site:

- **IpAddressInfo** - IP address definitions.

- **IpV4AddressInfo** - IPv4 address definitions.

- **IpV6AddressInfo** - IPv6 address definitions.

- **IpV4AddressSpec** - IPv4 address specification. Contains either static or DHCP configuration.

- `@optional IpV4AddressInfo staticAddressInfo` - If this optional field is set, then this spec denotes a static IPv4 address configuration. Otherwise, an unset field denotes a DHCPv4 configuration.

- **IpV6AddressSpec** - IPv6 address specification. Contains either static or DHCP configuration.
- @optional IpV6AddressInfo staticAddressInfo - If this optional field is set, then this spec
denotes a static IPv6 address configuration. Otherwise, an unset field denotes a DHCPv6
configuration.

- enum NetBiosMode - Used to configure NetBIOS on Windows systems. The values correspond
directly to Microsoft constants for NetBIOS mode.

- NicCustomizationSpec - Contains IP customization info for a specific network adapter.

- WindowsNicCustomizationSpec - Contains Windows specific IP customization info for a specific
network adapter for a virtual machine.

- @optional String dependentVmIds - This is a list of VM identities that must be powered-on
before this VM can be powered on. Dependencies are only valid within VMs of the same
recovery priority. If there are dependent VMs that are not in the current plan and same
recovery priority, they will be ignored. VM identity is available through
ProtectionGroup#ProtectedVm#drVmIdentity.

Faults

- RecoveryPlanNotFound
- RuntimeFault
- VmNotFoundInRecoveryPlan

See Faults in Site Recovery Manager API for more details.

SetRecoverySettings

Updates the virtual machines' Recovery Settings. This method updates the specified virtual
machine's Recovery Settings with values contained in the supplied RecoverySettings object. This
class modifies the recovery settings available through the external API. The VmIpCustomization
data object allows user to configure the IP address and corresponding DNS, WINS of the virtual
machine, after the migration is complete. You can disable IP customization by setting
VmIpCustomization to nullptr or by not setting IpCustomizationSpecMapping within
VmIpCustomization.

Synopsis

```c
void setRecoverySettings(vim.VirtualMachine vm, RecoverySettings settings)
```

vm is the Virtual Machine which Recovery Settings are to be updated.

settings is the Recovery Settings to update the VM.

RecoverySettings is the VM recovery settings for presentation in the user interface. For more
information, see GetRecoverySettings.

Faults

- DependencyConflict
AddProtectionGroup

Adds a protection group to the recovery plan.

Synopsis

```java
void addProtectionGroup(ProtectionGroup group)
```

group is the ProtectionGroup to be added. For information, see Protection Group.

Faults

- InvalidArgument
- NoPermission
- ProtectionGroupNotFound
- RecoveryPlanLocked
- RuntimeFault
- VersionConflict

See Faults in Site Recovery Manager API for more details.

AddTestNetworkMappingToRecoveryPlan

The AddTestNetworkMappingToRecoveryPlan method adds or updates a test network mapping to a recovery plan.

Synopsis

```java
void AddTestNetworkMappingToRecoveryPlan(
    vim.Network secondaryNetwork,
    vim.Network testNetwork)
```

secondaryNetwork specifies the network on the remote recovery site.

testNetwork specifies the test network on the remote recovery site.
Faults

- CannotMapDvsUplinkPortgroup
- ConnectionDownFault
- ManagedObjectNotFound
- NoPermission
- RemoteSiteNotAuthenticated
- RemoteSiteNotEnabled
- RecoveryPlanLocked
- RuntimeFault
- VersionConflict

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

RemoveTestNetworkMappingFromRecoveryPlan

The RemoveTestNetworkMappingFromRecoveryPlan method removes a test network mapping from a recovery plan.

Synopsis

```java
void RemoveTestNetworkMappingFromRecoveryPlan(vim.Network secondaryNetwork)
```

The `secondaryNetwork` parameter specifies the secondary site network whose mapping must be removed.

Faults

- ConnectionDownFault
- NetworkNotFound
- NoPermission
- RecoveryPlanLocked
- RemoteSiteNotAuthenticated
- RuntimeFault
- VersionConflict

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.
RemoveProtectionGroupFromRecoveryPlan

The `RemoveProtectionGroupFromRecoveryPlan` method removes a protection group from a recovery plan.

**Synopsis**

```java
void RemoveProtectionGroupFromRecoveryPlan(ProtectionGroup group)
```

The `group` parameter specifies the protection group that must be removed.

**Faults**

- NoPermission
- ProtectionGroupNotFound
- RecoveryPlanLocked
- RuntimeFault
- VersionConflict

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

RecoveryPlanGetLocation

The `RecoveryPlanGetLocation` method checks whether the recovery plan is hosted locally or on the paired site.

**Synopsis**

```java
String RecoveryPlanGetLocation()
```

The method returns the `localToRecoverySite`, `notLocalToRecoverySite`, `unknownLocationNoPgs`, and `unknownLocation` Strings.

<table>
<thead>
<tr>
<th>Returned Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>localToRecoverySite</td>
<td>The recovery plan instance exists locally on the recovery site.</td>
</tr>
<tr>
<td>notLocalToRecoverySite</td>
<td>The recovery plan instance is a peer of the recovery site instance.</td>
</tr>
<tr>
<td>unknownLocationNoPgs</td>
<td>The recovery plan instance has no protection groups. A plan with no protection groups is not local to either site.</td>
</tr>
<tr>
<td>unknownLocation</td>
<td>The location of the recovery plan instance is not known.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).
RecoveryPlanHasRunningTask

The RecoveryPlanHasRunningTask method checks whether there is a task that is associated with the recovery plan.

Synopsis

```java
boolean RecoveryPlanHasRunningTask()
```

The method returns true if there is a task that is associated with the recovery plan.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Recovery History

This section covers the interfaces to recovery history.

GetRecoveryResult

Retrieves recovery results for a given run of this recovery plan. Use this method to get the key so subsequent methods can get recovery results history.

Synopsis

```java
RecoveryResult[] getRecoveryResult(int length)
```

length is the maximum number of results to retrieve.

RecoveryResult[] is an array of recovery results for this recovery plan or its peer plan, including:

- description – summary of the plan at the time of this run
- errorCount – count of error-level faults that were generated by the operation
- executionTimeInSeconds – total execution time in seconds
- key – unique key for this recovery result, useful for subsequent methods
- name – the recovery plan’s name at the time of this run
- plan – recovery plan that this result covers
- resultState – the result state, which is only the final state indicating completion or failure. This enumerates the following values:
  - success - The operation completed with no warnings.
  - warnings - The operation completed with one or more warnings.
  - errors - The operation failed to complete due to one or more errors.
- cancelled - The operation was cancelled.

- runMode – mode of recovery when plan was initiated. This enumerates the following:
  - failover - Failover the recovery plan to the peer site. You can failover multiple times in case problems occurred on previous runs. Once completed successfully, the plan should be deleted or reprotected. For a plan to be successfully failed over, all of the groups in the plan must be in the shadowing, recovered, or partially Recovered states.
  - test - Run a test-failover at this site, leaving the primary state unaffected. For a plan to be successfully tested, all of the protection groups must be in the shadowing state.
  - cleanupTest - Cleanup after a test run.
  - reprotect - Complete an already finished recovery, and start protecting the groups so they may be recovered on the peer site. This will unregister the VMs on the peer site, configuring the storage, and the shadow VMs. This operation may only be performed when the sites are connected, and at least one protection group is in the recovered or partially Recovered state.
  - revert - Revert a recovery, abandoning all the VMs on the peer site and powering on the original VMs on the local site. This operation is not allowed unless all the replication groups are in the shadowing, recovered, or partially recovered state. If the sites are not connected, the peer VMs may be left running. In order to correct this situation, re-run revert after the sites are connected.

  **Note** revert is not supported currently.

- migrate - Migrate the recovery plan to the peer site. Once completed successfully, the plan should be deleted or reprotected. For a successful migration to occur all of the groups in the plan must be in the shadowing or recovered states.

- startTime, stopTime – time when the recovery was started and when it completed or stopped

- totalPausedTimeInSeconds – total time the recovery plan was paused

- warningCount – count of warning-level faults that were generated by the operation

- poweredOnVms - The count of the VM's that are powered on

- errorStateVms - The count of the VM's that are in Error state

- successfullyRecoveredVms - The count of the VM's that are successfully recovered

- ipCustomizedVms - The count of the VM's that are successfully ip customized

- errorCustomizedVms - The count of the VM's that encountered an error during IP customization.

- poweredOffVms - The count of the VM's that are powered off

- warnings - The warnings encountered for this Recovery Plan

- errors - The errors encountered for this Recovery Plan
Faults

- InvalidArgument
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

Example: Example for GetRecoveryResult

```java
List < SrmRecoveryResult > recoveryResult = srmPortType.getRecoveryResult(
    ManagedObjectReference _this,
    int length);

Where ManagedObjectReference _this = _historyRef;
where _historyRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(
    _recoveryRef);
ManagedObjectReference _historyRef = srmPortType.getHistory(
    _recoveryRef,
    plans.get(0));
```

GetResultCount

Retrieves the total number of stored results. This include historical results from both the plan and its peer plan if the sites are connected.

Synopsis

```java
int getResultCount()
```

Returns an integer count with the total number of history entries for this plan, and potentially its peer.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for GetResultCount

```java
int resultCount = srmPortType.getResultCount(
    ManagedObjectReference _this);

Where ManagedObjectReference _this = _historyRef;
where _historyRef can be taken from:
    SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
    ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(
    _recoveryRef);
GetResultLength

Retrieves the length of the XML result document for the requested Recovery Result.

Synopsis

```java
int getResultLength(long key)
```

key is the unique key for the plan history, from return value of the GetRecoveryResult method.

Returns an integer specifying the number of lines in the XML result file.

Faults

- RecoveryResultNotFound, if no result with that key exists.
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for GetResultLength

```java
int length = srmPortType.getResultLength(_this, key);
```

Where ManagedObjectReference _this = _historyRef; where _historyRef can be token from:

```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(_recoveryRef);
ManagedObjectReference _historyRef = srmPortType.getHistory(_recoveryRef,
plans.get(0));
```

RetrieveStatus

Retrieves an XML representation of the specified historical run of the referenced recovery plan. This XML document is transmitted in chunks limited by the maximum length of a string in the transport layer. You specify what line to start at and how many lines to return.

Synopsis

```java
String[] retrieveStatus(long key, int offset, int maxLines)
```
key is the unique key for the plan history, returned in RecoveryResult.key from getGetRecoveryResult.

offset is an integer specifying the starting line number in the XML file, beginning at 0,

maxLines is an integer specifying the maximum number of lines to retrieve.

Returns a string containing an XML representation of all recovery steps and their results.

Only after you have retrieved all the lines and assembled them do you have a valid XML document.

Faults

- RecoveryResultNotFound, if no result with that key exists.
- RuntimeFault
- vmomi.fault.InvalidArgument

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Example: Example for RetrieveStatus

```java
List < String > status = srmPortType.retrieveStatus(
    ManagedObjectReference _this,
    long key,
    int offset,
    int maxlines);
```

Where ManagedObjectReference _this = _historyRef;

where _historyRef can be taken from:

```java
SrmServiceInstanceContent content = _srmPortType.retrieveContent(_svcRef);
ManagedObjectReference _recoveryRef = content.getRecovery();
List < ManagedObjectReference > plans = srmPortType.listPlans(_recoveryRef);
ManagedObjectReference _historyRef = srmPortType.getHistory(_recoveryRef,
    plans.get(0));
```

IP Subnet Mapper

The IpSubnetMapper component resides on the recovery site and manages the IP subnet mapping between Protection and Recovery site networks.

GetIpSubnetMappings

This method returns an array of the IP subnet mappings for this IP Subnet Mapper.

Synopsis

```java
IPSubnetMapping[] getIpSubnetMappings();
```
IPSubnetMapping represents an association between a mapped network on the primary site with an IPMapping describing IP parameter translation during IP Subnet based customization. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primaryObject</td>
<td>Network on the primary (protected) site</td>
</tr>
<tr>
<td>secondaryObject</td>
<td>Network on the secondary (recovery) site</td>
</tr>
<tr>
<td>IPMapping ipMapping</td>
<td>IPMapping to apply to the destination network during IP customization</td>
</tr>
</tbody>
</table>

IPMapping defines the rule(s) used to translate VM's IP settings between protection and recovery sites. IPMapping can be associated with a protected site's vim.Network mapped to a recovery site's vim.Network with InventoryMapping.addNetworkMapping method. This allows IP settings for the recovered VMs be deduced based on the IP subnet parameters without a need to configure IP settings for each protected VM individually. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the IP mapping.</td>
</tr>
<tr>
<td>@optional SubnetRule[] rules</td>
<td>A set of network rules to evaluate/apply during recovery. The rules are evaluated in the order they are specified. SRM applies the first matched rule.</td>
</tr>
</tbody>
</table>

SubnetRule describes the mapping between protection site IP parameters to recovery site ones for a single IP subnet. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the Rule.</td>
</tr>
<tr>
<td>recoverySiteSubnet</td>
<td>CIDR specifying the recovery site subnet.</td>
</tr>
<tr>
<td>protectedSiteSubnet</td>
<td>CIDR specifying the protected site subnet.</td>
</tr>
<tr>
<td>@optional IPSettings recoverySiteIPSettings</td>
<td>IP settings to be applied for all matched network adapters on the recovery site when executing Test and Failover workflows.</td>
</tr>
<tr>
<td>dnsSuffixes[]</td>
<td>DNS Suffixes to be applied to all matched network adapters.</td>
</tr>
</tbody>
</table>

IPSettings is the IP customization info for a specific network adapter. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipV4Gateways</td>
<td>List of IPv4 gateways in the order of preference.</td>
</tr>
<tr>
<td>ipV6Gateways</td>
<td>List of IPv6 gateways in the order of preference.</td>
</tr>
<tr>
<td>dnsServerList</td>
<td>List of server IP addresses to use for DNS lookup.</td>
</tr>
</tbody>
</table>
In Windows, these settings are adapter-specific. In Linux they are used to build a global list of DNS servers for all adapters. Specify these servers in order of preference. If set in case of DHCP, the explicit DNS server list overrides the default DNS configuration acquired through DHCP protocol.

WindowsIPSettings is an extension for IPSettings. It is used for setting Windows specific IP customization information for a specific network adapter.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domainName</td>
<td>A connection-specific DNS domain name.</td>
</tr>
<tr>
<td>wins</td>
<td>The IP addresses of the primary and secondary WINS servers. The first element in this array is always the address of the primary WINS server and if a second element exists, it is the address of the secondary WINS server.</td>
</tr>
<tr>
<td>netBios</td>
<td>NetBIOS setting. Possible values are the constants defined by NetBiosMode enum.</td>
</tr>
</tbody>
</table>

NetBiosMode is used for configuring NetBIOS on Windows systems. The values correspond directly to Microsoft constants for NetBIOS mode. It enumerates the following:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableNetBiosViaDhcp</td>
<td>Enables the DHCP server to decide whether or not to use NetBIOS.</td>
</tr>
<tr>
<td>enableNetBios</td>
<td>Explicitly enables NetBIOS.</td>
</tr>
<tr>
<td>disableNetBios</td>
<td>Explicitly disables NetBIOS.</td>
</tr>
</tbody>
</table>

Fautes

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

AddIpMapping

This method associates an IPMapping object with an inventory-mapped protected site network. This must be called on secondary (recovery) site. The protectedNetwork should be mapped to recoveryNetwork by #InventoryMapper.addNetworkMappings() for the IPMapping to be applied to VMs connected to recoveryNetwork during IP customization. If an IPMapping had been already associated with the protectedNetwork, then it gets replaced by the new IPMapping. Up to 1 IPMapping containing multiple SubnetRule rules can be associated with any given protected site network at any time. If the destination (recovery) network is associated with a Test Network on the recovery site, then the ipMapping will be applied to the test network as well. This method requires Resource.com.vmware.vcDr.RecoveryUse privilege on destinationNetwork.
Synopsis

```java
```

addIpMapping has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protectedNetwork</td>
<td>Primary site network</td>
</tr>
<tr>
<td>recoveryNetwork</td>
<td>Secondary site network</td>
</tr>
<tr>
<td>ipMapping</td>
<td>IPMapping to associate with the existing Network Mapping. For more information, see GetIpSubnetMappings.</td>
</tr>
</tbody>
</table>

Faults

- ConnectionDownFault
- IpMappingFault
- InvalidArgument
- MissingNetworkMapping
- RemoteSiteNotAuthenticated
- RemoteSiteNotEnabled
- RuntimeFault
- StringArgumentTooLong
- UnknownPrimaryNetwork
- UnknownSecondaryNetwork

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

RemoveIpMappings

This method removes IPMappings from mapped primary (protected) site networks. Must be called on secondary (recovery) site.

Synopsis

```java
void removeIpMappings(@optional vim.Network[] protectedNetworks);
```

protectedNetworks primary (protected) site networks whose IPMapping is to be removed.

Faults

- RemoteSiteNotEnabled
- RuntimeFault
For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Storage Adapter

This section describes the methods for getting information about a storage adapter.

FetchInfo

This method fetches basic information about the SRA.

Synopsis

```java
@optional Info fetchInfo();
```

Info class has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>Universally unique identifier of the SRA which is preserved on SRA upgrades.</td>
</tr>
<tr>
<td>installPath</td>
<td>Path to the folder containing SRA installation.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>version</td>
<td>Version of the adapter.</td>
</tr>
<tr>
<td>vendor</td>
<td>Storage Vendor who owns the adapter.</td>
</tr>
<tr>
<td>helpUrl</td>
<td>URL for online documentation for the adapter.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetAdapterConnectionSpec

A partially complete connection spec that will have its address and opaque keys, for the key value pairs, pre-entered. Refer your SRA vendor specific documentation about what each key represents.

Synopsis

```java
AdapterConnectionSpec[] getAdapterConnectionSpec();
```

This method returns array of AdapterConnectionSpec objects for the available SRAs. Part of the data is pre-entered, as shown in the example below:
Example of returned partially-complete spec

```java
class connectionSpec {
    String key = "primary"
    String name = "Primary SAN"
    String hint = "Primary SAN connection parameters"
    String address[0].key = "spA"
    String address[0].name = "IP Address of SP-A"
    String address[0].hint = "Enter IP address of the Storage Processor A"
    String address[0].value = empty (expected user input)
    String opaque[0].key = "volumeNameFilter"
    String opaque[0].name = "Volume name prefix limiting discovery"
    String opaque[0].hint = "Leave empty for full discovery"
    String opaque[0].optional = "true"
    String address[0].value = empty (expected user input)
}
```

AdapterConnectionSpec has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataPrompt key</td>
<td>Identifier of a SRA-defined group of connection parameters. Refer your SRA vendor specific documentation for more information about this value. This will be automatically entered by the StorageAdapter#getAdapterConnectionSpec.</td>
</tr>
<tr>
<td>DataPrompt address[]</td>
<td>List of address-type parameters. Refer your SRA vendor specific documentation about the keys and their corresponding values. Keys will be automatically entered by the StorageAdapter#getAdapterConnectionSpec.</td>
</tr>
<tr>
<td>username</td>
<td>Username.</td>
</tr>
<tr>
<td>password</td>
<td>Password if required.</td>
</tr>
<tr>
<td>OpaquePrompt opaque[]</td>
<td>Opaque parameters if required. Refer your SRA vendor specific documentation about the keys and their corresponding values. Keys will be automatically entered by the StorageAdapter#getAdapterConnectionSpec.</td>
</tr>
</tbody>
</table>

DataPrompt contains key-value pairs with additional information about the key and what value it expects. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>SRA Specific key, this can be either for the address or the opaque.</td>
</tr>
<tr>
<td>name</td>
<td>Prompt string.</td>
</tr>
<tr>
<td>hint</td>
<td>Sample or more verbose description of the requested data for</td>
</tr>
<tr>
<td>value</td>
<td>Value corresponding to the key.</td>
</tr>
</tbody>
</table>

LocalizableString describes localizable string returned from the SRA. Localization support is optional for SRA. If supported, then each string is returned with a key which could be used to lookup a translation in non-default locale.
Example of localizable string returned from SRA: &lt;Xxx stringId="Foo">Foo</Xxx&gt;

Example of non-localizable string returned from SRA: &lt;Xxx&gt;Foo&lt;/Xxx&gt;

LocalizableString has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Key to look up translation for the string. This key is made optional to accommodate SRAs which do not support localization. These SRAs will just return strings in default locale (English).</td>
</tr>
<tr>
<td>text</td>
<td>String text in default locale</td>
</tr>
</tbody>
</table>

OpaquePrompt extends DataPrompt. It contains additional information about the opaque parameters. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>optional</td>
<td>Boolean indicating whether or not this parameter is optional.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

Storage

This section shows a method that you can use to rescan storage.

DiscoverDevices

The discoverDevices method starts a loop through the array managers on the local and remote sites, and for each array pair call discoverDevices. If the user does not have enough privileges, the array managers are skipped.

Synopsis

```java
DiscoverDevicesTask discoverDevices()
```

The method returns a DiscoverDevicesTask object that contains information about the status of the operation. Site Recovery Manager Server retains the object for 30 minutes after the task finishes.

IsDiscoverDevicesTaskComplete returns true if the DiscoverDeviceTask is complete.

GetDiscoverDevicesTaskFailures returns a list of failures that occurred while performing DiscoverDevices on array managers. The entire set of failures can be retrieved when the task is complete. If no failures occurred, the array is empty or null.
DiscoverDevicesFailure.name is a string that contains the array pair name.
DiscoverDevicesFailure.fault is an MethodFault object that indicates the failure.

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

QueryArrayManagers

The queryArrayManagers method returns a list of all the available array managers.

Synopsis

@optional ArrayManager[] queryArrayManagers()

ArrayManager[] is an array of available array managers. For more information, see ArrayManager.

Faults
- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

CreateArrayManager

Creates ArrayManager object. Performs array discovery as part of array manager creation and fails if array discovery fails.

Synopsis

@task ArrayManager createArrayManager(
    String name,
    String uuid,
    AdapterConnectionSpec[] connectionSpec)

createArrayManager returns a task instance to monitor the asynchronous operation of this method. ArrayManager object is returned as task result. The createArrayManager has the following parameters:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the Array Manager</td>
</tr>
<tr>
<td>uuid</td>
<td>Universally unique identifier of the SRA</td>
</tr>
<tr>
<td>connectionSpec</td>
<td>SRA-specific connection parameters for the underlying storage management system. For more information, see GetAdapterConnectionSpec.</td>
</tr>
</tbody>
</table>

**Note** When getting the AdapterConnectionSpec with getAdapterConnectionSpec, all the optional opaques will be returned. However, some vendors do not allow not setting the optional opaques. Ensure that they are excluded from the spec.

If a task fails, its error field may contain one of the following:

- drextapi.fault.DuplicateName - if an array manager with the same name already exists.
- drextapi.fault.InvalidAdapterConnectionSpec - if connectionSpec does not match the internal StorageAdapter connection spec typically, a more specific fault is thrown.
- drextapi.fault.CommandFailed - if the command for creating an array manager fails.
- drextapi.fault.DuplicateArray - if there is another array manager that already discovered a given array.

### Faults

- vim.fault.InvalidArgument
- StorageAdapterNotFound
- StringArgumentTooLong
- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.

## QueryStorageAdapters

List of Storage Replication Adapters (SRAs) information successfully loaded into SRM.

### Synopsis

```java
@optional StorageAdapter[] queryStorageAdapters();
```

StorageAdapter provides information about the Storage Adapter. For more information see [Storage Adapter](#).

### Faults

- RuntimeFault

See [Faults in Site Recovery Manager API](#) for more details.
**RemoveArrayManager**

Deletes `ArrayManager` object. Array managers deletion is deferred until all the tasks currently in progress are complete.

**Synopsis**

```java
@task void removeArrayManager(ArrayManager arrayManager);
```

`removeArrayManager` returns a task instance to monitor the asynchronous operation of this method. `removeArrayManager` has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrayManager</td>
<td>ArrayManager object to delete. For more information, see <code>ArrayManager</code>.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain the following:

- `drestapi.fault.ArrayManagerInUse` - if there are array pairs configured for this array manager.

**Faults**

- `RuntimeFault`
- `TaskInProgress`
- `vim.fault.InvalidArgument`

See [Faults in Site Recovery Manager API](#) for more details.

**ReloadAdapters**

Scans SRA installation directory and reloads SRAs. It returns a task object to monitor the process.

**Synopsis**

```java
@task void reloadAdapters();
```

**Faults**

- `RuntimeFault`
- `TaskInProgress`

See [Faults in Site Recovery Manager API](#) for more details.

**ArrayManager**

This section presents methods to interact with the SRM array managers.
ReadInfo

The ReadInfo method returns information specific to the ArrayManager instance.

Synopsys

ArrayManagerInfo readInfo()

Returns ArrayManagerInfo which contains information for the ArrayManager instance. It has the name field, which is the name of the ArrayManager.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

QueryReplicatedArrayPairs

The queryReplicatedArrayPairs method returns a list of all the replicated array pairs in the ArrayManager.

Synopsis

@optional ReplicatedArrayPair[] queryReplicatedArrayPairs()

ReplicatedArrayPair[] is an array of the replicated array pairs in the ArrayManager. For more information, see ReplicatedArrayPair.

Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetArrayInfo

This method gets the list of discovered storage arrays. This list is populated by discoverArrays API.

Synopsis

@optional ArrayInfo[] getArrayInfo();

ArrayInfo class describes the storage array configured for replication. It has the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Storage array identifier. The 'key' name is used to allow partial property updates for ArrayManager.arrayInfo[]. This is passed to ArrayManager#addArrayPair arrayId.</td>
</tr>
<tr>
<td>name</td>
<td>User-friendly name of the storage.</td>
</tr>
<tr>
<td>@optional PeerArrayInfo peerArray[]</td>
<td>Describes a peer array. It has a property named key. This is passed to AddArrayPair peerArrayId.</td>
</tr>
</tbody>
</table>

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

**GetAdapter**

Returns the corresponding storage adapter to the ArrayManager. If the ArrayManager does not have a storage adapter then, the unset value is returned.

**Synopsis**

```java
@optional StorageAdapter getAdapter();
```

StorageAdapter is a managed object that provides information about the Storage Adapter. For more information, see Storage Adapter.

**Note** In a scenario where the ArrayManager may not have a storage adapter, the unset value is returned.

**Faults**

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

**AddArrayPair**

Creates ReplicatedArrayPair object for a given pair of storage arrays.

**Synopsis**

```java
@task ReplicatedArrayPair addArrayPair(String arrayId, String peerArrayId)
```

addArrayPair returns task instance to monitor the asynchronous operation of this method. ReplicatedArrayPair object is returned as task result. addArrayPair has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrayId</td>
<td>Array identifier of the local storage</td>
</tr>
<tr>
<td>peerArrayId</td>
<td>Identifier of the storage array at the remote site</td>
</tr>
</tbody>
</table>
If a task fails, its error field may contain one of the following:

- `drextapi.fault.ArrayNotFound` - if array with ID `arrayId` cannot be found at local site, or if array with ID `peerArrayId` cannot be found at remote site.
- `drextapi.fault.ReplicatedArrayPairAlreadyExists` - if an array pair involving same `arrayId` and `peerArrayId` already exists.
- `drextapi.fault.PeerArrayNotFound` - if `peerArrayId` is not a valid peer of array pointed by `arrayId` on local site, or if `arrayId` is not a valid peer of array pointed by `peerArrayId` on remote site.
- `drextapi.fault.ConnectionDownFault` - if the other site involved in the operation could not be reached.

Faults

- RemoteSiteNotEnabled
- RuntimeFault
- TaskInProgress
- vim.fault.InvalidArgument

See Faults in Site Recovery Manager API for more details.

**RemoveArrayPair**

Deletes specified ReplicatedArrayPair object. Returns a task instance to monitor the asynchronous operation of this method.

**Synopsis**

```python
@task void removeArrayPair(ReplicatedArrayPair arrayPair)
```

`removeArrayPair` has the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrayPair</td>
<td>ReplicatedArrayPair to be removed</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain one of the following:

- `drextapi.fault.ArrayPairInUse` - if array pair is in use by protection group(s).
- `drextapi.fault.ConnectionDownFault` - if the other site involved in the operation could not be reached.

Faults

- RuntimeFault
- TaskInProgress
- vim.fault.InvalidArgument
See [Faults in Site Recovery Manager API](#) for more details.

**DiscoverArrays**

Discovered storage arrays configured for replication by executing SRA command discoverArrays. Ids of discovered array must be unique across all array managers that use the same SRA.

**Synopsis**

```java
@task ArrayInfo[] discoverArrays()
```

discoverArrays returns a task instance to monitor the asynchronous operation of this method. Array of ArrayInfo objects is returned as task result.

ArrayInfo class describes storage array configured for replication. For more information, see [GetArrayInfo](#).

If a task fails, its error field may contain one of the following:

- `drestapi.fault.StORAGEAdapterNotFound` - if SRA was not found for this array manager.
- `drestapi.fault.DuplicateArray` - if an array, which is already discovered by another array manager with the same SRA, was discovered.
- `drestapi.fault.ArrayPairNotFound` - if arrays are not found for already configured array pair.

**Faults**

- RuntimeFault
- TaskInProgress

See [Faults in Site Recovery Manager API](#) for more details.

**Reconfigure**

Updates array manager name and connection parameters for the SRA. Performs array discovery as part of reconfigure operation. Ensures that all existing configured array pairs are not affected by the connection specs changes and are still discovered. Array discovery is not performed if only name change is requested and connection specs are unchanged.

**Synopsis**

```java
@task void reconfigure(
    String name,
    @optional AdapterConnectionSpec[] connectionSpec)
```

reconfigure returns a task instance to monitor the asynchronous operation of this method. It has the following parameters:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>New name.</td>
</tr>
<tr>
<td>connectionSpec</td>
<td>New connection parameters.</td>
</tr>
<tr>
<td>AdapterConnectionSpec</td>
<td>Connection parameters for the SRA provided by the user. For more information, see GetAdapterConnectionSpec.</td>
</tr>
</tbody>
</table>

If a task fails, its error field may contain one of the following:

- `drextapi.fault.ArrayPairNotFound` - if arrays are not found for already configured array pair.
- `drextapi.fault.DuplicateName` - if an array manager with the same name already exists.
- `drextapi.fault.DuplicateArray` - if array, which is already discovered by another array manager with the same SRA, was discovered.
- `drextapi.fault.InvalidAdapterConnectionSpec` - if `connectionSpec` doesn't match the internal `StorageAdapter` connection spec typically, a more specific fault is thrown.
- `drextapi.fault.StringArgumentTooLong` - if the size of the name parameter is too long.
- `drextapi.fault.StorageAdapterNotFound` - if SRA was not found for this array manager.

### Faults

- `RuntimeFault`
- `TaskInProgress`

See [Faults in Site Recovery Manager API](#) for more details.

### GetArrayDiscoveryStatus

Returns the status and timestamp information of latest array discovery. This can be an unset value if no array discovery is still executed.

### Synopsis

```java
@optional ArrayDiscoveryStatus getArrayDiscoveryStatus()
```

`ArrayDiscoveryStatus` describes the status of the most recent array discovery. Array discovery can be initiated by calling `discoverArrays` API. Additionally, there is a periodic auto-discovery executed in the background. It has the following the fields:
Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

ReplicatedArrayPair

This section presents methods to interact with the replicated array pairs.

QueryReplicatedRdms

The queryReplicatedRdms method returns information about all the replicated RDMs in the ReplicatedArrayPair.

Synopsis

```java
@optional ReplicatedRdmInfo[] queryReplicatedRdms()
```

ReplicatedRdmInfo[] is an array and it contains the following information about all the replicated RDMs in the ReplicatedArrayPair:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Unique key identifying this object. The value is constructed from virtual machine MoID and virtual device key.</td>
</tr>
<tr>
<td>device</td>
<td>Storage device identifier.</td>
</tr>
<tr>
<td>deviceGroup</td>
<td>Consistency group identifier, if any.</td>
</tr>
<tr>
<td>stretchedStorage</td>
<td>Indicates if this is a stretched RDM device.</td>
</tr>
<tr>
<td>sitePreference</td>
<td>For stretched RDM devices, indicates whether this device has the site preference or not.</td>
</tr>
<tr>
<td>vm</td>
<td>Virtual machine to which the RDM is attached.</td>
</tr>
<tr>
<td>deviceKey</td>
<td>Virtual device key of the attached RDM.</td>
</tr>
<tr>
<td>lunUuid</td>
<td>UUID of the RDM LUN.</td>
</tr>
</tbody>
</table>
Faults

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

GetDevices

Returns list of storage devices configured for replication.

Synopsis

```java
@optional StorageDevice[] getDevices()
```

StorageDevice is data object that describes storage device configured for replication. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupId</td>
<td>SRA-specific identifier of a consistency group of the device if any. This property must be set if device is part of a consistency group.</td>
</tr>
<tr>
<td>groupName</td>
<td>User-friendly name of the consistency group. SRA capabilities determine whether this property is set or not. It is either set for all devices or none. This property is set if StorageDevice#groupId is set and StorageDeviceGroupBase#name is set.</td>
</tr>
</tbody>
</table>

DeviceProperty describes the SRA-specific device property. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalizableString name</td>
<td>User-friendly property name.</td>
</tr>
</tbody>
</table>

StorageDevice extends StorageDeviceGroupBase, a base class for storage devices and storage groups. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SRA-specific identifier.</td>
</tr>
<tr>
<td>name</td>
<td>User-friendly name. SRA capabilities determine whether this property is set or not. It is either set for all devices or none.</td>
</tr>
<tr>
<td>role</td>
<td>Role of devices and groups in the replication relationship. See drextapi.StorageDeviceRole.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>targetKey</td>
<td>Key of the promoted replication target if role is promotedTarget. This property is set only for groups and devices with role</td>
</tr>
<tr>
<td></td>
<td>drextapi.StorageDeviceRole#promotedTarget.</td>
</tr>
<tr>
<td>replicationSettings</td>
<td>Replication settings opaque to SRM.</td>
</tr>
<tr>
<td>@optional MethodFault</td>
<td>queryReplicationSettingsFault Error occurred while querying SRA for replication settings, if any.</td>
</tr>
</tbody>
</table>

StorageDeviceGroupBase extends StorageItemBase, a base class for replicated datastores, storage devices and storage groups. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stretchedStorage</td>
<td>Boolean true if stretched storage is enabled.</td>
</tr>
<tr>
<td>staticSitePreference</td>
<td>Boolean. This property is set for stretched devices, device groups and replicated datastores. True if static site preference is supported.</td>
</tr>
</tbody>
</table>

Faults
- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetDeviceGroups

Returns list of consistency groups of storage devices configured for replication.

Synopsis

```java
StorageDeviceGroup[] getDeviceGroups()
```

StorageDeviceGroup describes a consistency group of storage devices. The StorageDeviceGroup data class extends the StorageDeviceGroupBase. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>devices</td>
<td>SRA specific identifiers of the storage devices in the group.</td>
</tr>
</tbody>
</table>

StorageDeviceGroupBase is a base class for storage devices and storage groups. For information, see GetDevices.

Faults
- RuntimeFault

See Faults in Site Recovery Manager API for more details.
GetReplicatedDatastores

Returns list of datastores residing on replicated storage devices. This list is populated by scanning all datastores in local vCenter inventory and matching underlying storage devices to replicated storage devices, discovered by SRM’s discoverDevices operation.

Synopsis

```java
@optional ReplicatedDatastore[] getReplicatedDatastores();
```

ReplicatedDatastore data class describes a datastore residing on replicated storage devices. It extends the StorageItemBase. ReplicatedDatastore has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>Unique key identifying this object.</td>
</tr>
<tr>
<td>datastore</td>
<td>datastore residing on replicated storage devices.</td>
</tr>
<tr>
<td>StorageDevicePartition[] extent</td>
<td>List of device partitions corresponding to datastore extents.</td>
</tr>
<tr>
<td>stretchedStorage</td>
<td>Indicates if this is a stretched datastore.</td>
</tr>
<tr>
<td>staticSitePreference</td>
<td>For stretched datastore indicates whether this datastore has a static site preference or a dynamic site preference.</td>
</tr>
<tr>
<td>sitePreference</td>
<td>For a stretched datastore, indicates whether this datastore has the site preference or not. This attribute is relevant for datastore with a static site preference.</td>
</tr>
<tr>
<td>array</td>
<td>array Reference to the array holding storage devices backing the datastore.</td>
</tr>
</tbody>
</table>

StorageDevicePartition describes a partition of a storage device. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
<td>Storage device identifier.</td>
</tr>
<tr>
<td>deviceGroup</td>
<td>Consistency group identifier, if any.</td>
</tr>
<tr>
<td>partition</td>
<td>Partition number.</td>
</tr>
</tbody>
</table>

StorageItemBase is the base class for replicated datastores, storage devices, and storage groups. For more information, see GetDevices.

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetDeviceDiscoveryStatus

Gets the storage device discovery status. Contains timestamp and faults data for the most recent device discovery.
Synopsis

```java
@optional DeviceDiscoveryStatus getDeviceDiscoveryStatus();
```

DeviceDiscoveryStatus describes status of the most recent storage device discovery. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fault</td>
<td>Fault occurred during the most recent discovery of storage devices if any.</td>
</tr>
<tr>
<td>peerMatchingFault</td>
<td>Local to remote peer device and device group matching faults, if any.</td>
</tr>
<tr>
<td>startTimestamp</td>
<td>Start time of the most recent discovery of storage devices.</td>
</tr>
<tr>
<td>completeTimestamp</td>
<td>Completed time of the most recent discovery of storage devices. This property is not set if discovery task is currently in progress.</td>
</tr>
</tbody>
</table>

Vvol Replication

vVol Replication API provides information about the local vVol topology replicated to the SRM peer site.

GetDomains

Returns a list of local vVol fault domains with their replication groups which target fault domains matching SRM peer site.

Synopsis

```java
@optional DomainInfo[] getDomains()  
```

DomainInfo[] is a list of fault domain information data objects describing the currently available vVol fault domains. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the fault domain.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the fault domain.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the fault domain. The description is expected to be already localized by the VASA provider and the VC.</td>
</tr>
<tr>
<td>vasaProviderUid</td>
<td>Identifier of the vendor of the VASA provider for this fault domain.</td>
</tr>
<tr>
<td>vasaProviderVendor</td>
<td>Name of the vendor of the VASA provider for this fault domain.</td>
</tr>
<tr>
<td>vasaProviderModel</td>
<td>VASA provider model name for this fault domain.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>vasaProviderVersion</td>
<td>VASA provider version string for this fault domain.</td>
</tr>
<tr>
<td>@optional ReplicationGroupInfo[] sourceReplicationGroups</td>
<td>Source replication groups in this fault domain. Empty if no replication groups are in SOURCE state.</td>
</tr>
</tbody>
</table>

ReplicationGroupInfo[] is the information about a vVol replication group. For more information, see GetVvolGroupDetails.

**Note** DomainInfo and ReplicationGroupInfo extends FaultInfo. For more information, see GetVvolGroupDetails.

**Faults**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see Faults in Site Recovery Manager API.

**GetUnprotectedVms**

Returns a list of unprotected vVol replicated virtual machines part of vVol replication groups that target the SRM peer site.

**Synopsis**

```java
@optional UnprotectedVmInfo[] getUnprotectedVms(@optional FaultDomainId[] domains);
```

domains is an optional list of domains to filter the result. If not set the server will return all protectable VMs.

UnprotectedVmInfo[] is a list containing information about unprotected vVol replicated virtual machines part of vVol replication groups that target the SRM peer site. It includes empty Replication Groups that are not part of any SRM protection group and with no virtual machines. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationGroupInfo replicationGroup</td>
<td>Information about a vVol replication group with SRM peer site as target.</td>
</tr>
<tr>
<td>protectionGroup</td>
<td>SRM Protection group which protects this replication group. Unset if the replication group is not protected by any SRM protection group.</td>
</tr>
<tr>
<td>VmInfo[] unprotectedVms</td>
<td>List of vVol replicated virtual machines that are unprotected from SRM point of view and replicated with this replication group. This list might include VMs with vVol configuration errors. Only VMs with no errors are suitable for protection by SRM.</td>
</tr>
</tbody>
</table>

ReplicationGroupInfo is the information about a vVol replication group. For more information, see GetDomains.
VmInfo is the information about a local vVol replicated `vim.VirtualMachine`. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The <code>vim.VirtualMachine</code> object at the local site.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the virtual machine.</td>
</tr>
<tr>
<td><code>@optional ReplicationGroupInfo[]</code> replicationGroups</td>
<td>Replication groups for this VM. Virtual machines sharing the same replication groups belong to the same consistency group. They will be added or removed from a protection group together. For information about <code>ReplicationGroupInfo</code>, see <code>GetDomains</code>.</td>
</tr>
</tbody>
</table>

**Note**  
VmInfo and `ReplicationGroupInfo` extends `FaultInfo`. For more information, see `GetVvolGroupDetails`.

**Faults**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Rescan**

Initiates a rescan of the server's local vVol configuration. The server keeps updating its view of the local vVol configuration periodically. This results in the newly provisioned vVol virtual machines being available for protection only after the passage of update interval. This function is called to force an update.

**Synopsis**

```
Task rescan()
```

Task returns an object to the operation. For more information, see [SrmExtApiTask](#).

**Faults**

- RuntimeFault

For information about the faults that Site Recovery Manager throws, see [Faults in Site Recovery Manager API](#).

**Placeholder Datastore Manager**

This section describes the used method to manage placeholder datastores.
AddDatastore

Adds datastore to the list of placeholder datastores.

Synopsis

```java
@task @optional AddDatastoreResult[] addDatastore(@optional vim.Datastore[] datastore);
```

AddDatastoreResult structure is returned as task result. It has the following properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The datastore that could not be used as a placeholder datastore.</td>
</tr>
<tr>
<td>fault</td>
<td>The list of errors for this datastore.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

RemoveDatastore

Removes datastore(s) from the list of placeholder datastores.

Synopsis

```java
@task RemoveDatastoreResult removeDatastore(@optional vim.Datastore[] datastore);
```

RemoveDatastoreResult structure is returned as task result. It has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notConfigured</td>
<td>The list of datastores that are not configured as placeholder datastores.</td>
</tr>
</tbody>
</table>

Faults

- RuntimeFault

See Faults in Site Recovery Manager API for more details.

GetPlaceholderDatastores

Gets the list of all configured placeholder datastores.

Synopsis

```java
@optional PlaceholderDatastoreInfo[] getPlaceholderDatastores()
```

PlaceholderDatastoreInfo has the following fields:
Field | Description
--- | ---
fault | The list of errors for this datastore. If this property is not empty, this datastore can not be used as a placeholder datastore.

PlaceholderDatastoreInfo extends DatastoreInfo. It has the following fields:

Field | Description
--- | ---
key | reference to the actual vim.Datastore object in VC.
capacity | The total capacity of this datastore in bytes.
freeSpace | The amount of free space in bytes available on this datastore.
reservedSpace | The amount of additional storage space in bytes on this datastore that could potentially be used by all virtual machines on this datastore. A non-zero value in this field is associated with virtual machines having Thin Provisioned disks.
type | The type of the file system on this datastore, such as VMFS, NFS or CIFS. @see vim.host.FileSystemVolume#type

@optional ComputeResourceInfo[] visibleTo | The list of hosts and clusters that can access this datastore. If a cluster is included in this list, it means that this datastore is accessible to all hosts in this cluster. In most cases this list can not be empty. The only reason for this field to be optional is to be able to return placeholder datastores that have become invalid because there are no clusters or hosts that can see this datastore. This could happen, for example, if the user were to add a host to a cluster and this host could not see some of the datastores that were previously visible to all hosts in the cluster.

ComputeResourceInfo has the following properties:

Field | Description
--- | ---
key | The reference to the actual vim.ComputeResource object in VC. Hosts will be represented by vim.ComputeResource with a single vim.HostSystem in it. Clusters will be represented by vim.ClusterComputeResource.

DatastoreInfo and ComputeResourceInfo extend ObjectInfo which has the following properties:

Field | Description
--- | ---
name | The name of the object.
status | The status of the object, such as green, red or yellow. @see vim.ManagedEntity.Status

Faults
- RuntimeFault
See Faults in Site Recovery Manager API for more details.
## Deprecated APIs

### Deprecated Site Recovery Manager APIs

Table 8-1. Replaced APIs

<table>
<thead>
<tr>
<th>SrmApi</th>
<th>Replacement API</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListRecoveryPlans</td>
<td>SrmRecovery.ListPlans</td>
</tr>
<tr>
<td>RecoveryPlanAnswerPrompt</td>
<td>SrmRecoveryPlan.AnswerPrompt</td>
</tr>
<tr>
<td>GetFinalStatus</td>
<td>SrmRecoveryHistory.GetRecoveryResult</td>
</tr>
<tr>
<td>GetApiVersion</td>
<td>no replacement</td>
</tr>
<tr>
<td>SrmApi.SrmLogin</td>
<td>SrmServiceInstance.SrmLoginLocale</td>
</tr>
<tr>
<td>SrmApi.SrmLoginByToken</td>
<td>SrmServiceInstance.SrmLoginByTokenLocale</td>
</tr>
<tr>
<td>SrmApi.SrmLogout</td>
<td>SrmServiceInstance.SrmLogoutLocale</td>
</tr>
<tr>
<td>SrmApi.GetApiVersion</td>
<td>ServiceInstanceContent.apiVersion</td>
</tr>
<tr>
<td>SrmApi.ListRecoveryPlans</td>
<td>SrmRecovery.ListPlans</td>
</tr>
<tr>
<td>SrmApi.RecoveryPlanStart</td>
<td>SrmRecoveryPlan.Start</td>
</tr>
<tr>
<td>SrmApi.RecoveryPlanPause</td>
<td>no replacement</td>
</tr>
<tr>
<td>SrmApi.RecoveryPlanResume</td>
<td>no replacement</td>
</tr>
<tr>
<td>SrmApi.RecoveryPlanAnswerPrompt</td>
<td>SrmRecoveryPlan.AnswerPrompt</td>
</tr>
<tr>
<td>SrmApi.GetFinalStatus</td>
<td>SrmRecoveryHistory.GetRecoveryResult</td>
</tr>
<tr>
<td>RemoteSite.vcHost, RemoteSite.vcPort</td>
<td>RemoteSite.lkpUrl and RemoteSite.vcInstanceUuid</td>
</tr>
<tr>
<td>SrmServiceInstance.getSiteName</td>
<td>SrmServiceInstance.GocalSiteInfo.siteName</td>
</tr>
<tr>
<td>SrmProtection.listReplicatedDatastores</td>
<td>SrmProtection.listUnassignedReplicatedDatastores</td>
</tr>
<tr>
<td>createHbrProtectionGroup</td>
<td>createHbrProtectionGroup2</td>
</tr>
</tbody>
</table>
Site Recovery Manager Faults

This chapter lists the various faults thrown by the Site Recovery Manager Virtual Appliance Management APIs and the Site Recovery Manager APIs.

This chapter includes the following topics:

- Faults in Site Recovery Manager Appliance Management API
- Faults in Site Recovery Manager API

Faults in Site Recovery Manager Appliance Management API

This section lists the various faults thrown by the Site Recovery Manager Appliance Management APIs.

Table 9-1. Faults thrown by Site Recovery Manager Appliance Management functions

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotCreateSraLogDirectory</td>
<td>Thrown when the SRA directory cannot be created.</td>
</tr>
<tr>
<td>CannotExecuteDockerCommand</td>
<td>Thrown when there are issues executing the ‘docker’ CLI inside the SRM VA, that is there is no ‘docker’ command in the '/usr/bin/' directory; there are no permissions to execute the ‘docker’ command, and so on.</td>
</tr>
<tr>
<td>CannotGenerateSystemLogBundle</td>
<td>Thrown when bundle generation fails and we cannot narrow it down to a more precise result.</td>
</tr>
<tr>
<td>CertificateBadKeyPair</td>
<td>Thrown when the private key does not match the certificate’s public key.</td>
</tr>
<tr>
<td>CertificateCaNotAllowed</td>
<td>Thrown when certification authority certificate is provided.</td>
</tr>
<tr>
<td>CertificateCtlInvalidForUsage</td>
<td>Thrown when a certificate trust list used to create this chain is not valid for this usage.</td>
</tr>
<tr>
<td>CertificateCtlSignatureInvalid</td>
<td>Thrown when a certificate trust list used to create this chain did not have a valid signature.</td>
</tr>
<tr>
<td>CertificateCtlTimeInvalid</td>
<td>Thrown when a certificate trust list used to create this chain was not time-valid.</td>
</tr>
<tr>
<td>CertificateCyclicError</td>
<td>Thrown when a cycle in the certificate chain of trust was detected.</td>
</tr>
<tr>
<td>CertificateDnsMismatch</td>
<td>Thrown when the certificate’s Subject Alternative Name does not meet certain conditions.</td>
</tr>
</tbody>
</table>
Table 9-1. Faults thrown by Site Recovery Manager Appliance Management functions (continued)

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CertificateHasExpired</td>
<td>Thrown when the certificate has expired.</td>
</tr>
<tr>
<td>CertificateInvalidKeyLength</td>
<td>Thrown when the certificate does not have the minimum</td>
</tr>
<tr>
<td></td>
<td>required key length.</td>
</tr>
<tr>
<td>CertificateMd5NotAllowed</td>
<td>Thrown when certificate signed with MD5 is provided.</td>
</tr>
<tr>
<td>CertificateNotTimeNested</td>
<td>Thrown when certificates in the host’s certificate chain are</td>
</tr>
<tr>
<td></td>
<td>not properly time-nested.</td>
</tr>
<tr>
<td>CertificateNotTrustedByDrConfig</td>
<td>DrConfig cannot validate SSL certificate.</td>
</tr>
<tr>
<td>CertificateNotValidForUsage</td>
<td>Thrown when a certificate in the host’s chain is not valid</td>
</tr>
<tr>
<td></td>
<td>in its proposed usage.</td>
</tr>
<tr>
<td>CertificateNotYetValid</td>
<td>Thrown when the certificate is not yet valid.</td>
</tr>
<tr>
<td>CertificateParseError</td>
<td>Thrown when certificate can't be parsed due to incorrect</td>
</tr>
<tr>
<td></td>
<td>format or password.</td>
</tr>
<tr>
<td>CertificatePartialChain</td>
<td>Thrown when the host certificate chain is not complete.</td>
</tr>
<tr>
<td>CertificateRevocStatusUnknown</td>
<td>Thrown when a certificate in the host’s chain has an</td>
</tr>
<tr>
<td></td>
<td>unknown revocation status.</td>
</tr>
<tr>
<td>CertificateRevokedError</td>
<td>Thrown when trust for a certificate in the host’s chain has</td>
</tr>
<tr>
<td></td>
<td>been revoked.</td>
</tr>
<tr>
<td>CertificateSha1NotAllowed</td>
<td>Thrown when certificate signed with SHA-1 is not allowed.</td>
</tr>
<tr>
<td>CertificateSignatureNotValid</td>
<td>Thrown when a certificate in the host’s chain does not have</td>
</tr>
<tr>
<td></td>
<td>a valid signature.</td>
</tr>
<tr>
<td>CertificateTimeNotValid</td>
<td>Thrown when a certificate in the host’s chain is not time-</td>
</tr>
<tr>
<td></td>
<td>valid.</td>
</tr>
<tr>
<td>CertificateUnknownError</td>
<td>Thrown when an unknown certificate trust error occurs.</td>
</tr>
<tr>
<td>CertificateUntrustedRoot</td>
<td>Thrown when a certificate in the host’s chain is based on an</td>
</tr>
<tr>
<td></td>
<td>untrusted root.</td>
</tr>
<tr>
<td>ChangePasswordFault</td>
<td>Thrown when change pass operation fails.</td>
</tr>
<tr>
<td>CommandFailed</td>
<td>Thrown when the SRA command execution fails.</td>
</tr>
<tr>
<td>CommandResponseMissing</td>
<td>Thrown when a request is made involving a non-existent</td>
</tr>
<tr>
<td></td>
<td>docker image.</td>
</tr>
<tr>
<td>ConnectionError</td>
<td>Thrown during a failed TCP connection when accessing an</td>
</tr>
<tr>
<td></td>
<td>address using a port number.</td>
</tr>
<tr>
<td>ConnectionRefusedFault</td>
<td>Thrown when the connection was refused by the target.</td>
</tr>
<tr>
<td>CreateDbSchemaFault</td>
<td>Thrown when there is an error during database schema creation.</td>
</tr>
<tr>
<td>DatabaseConnectionFault</td>
<td>Thrown when there’s an error while connecting to</td>
</tr>
<tr>
<td></td>
<td>database.</td>
</tr>
<tr>
<td>DnsLookupFault</td>
<td>Thrown when failed to look up the server in DNS.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DockerCommandFailed</td>
<td>Thrown when the ‘docker’ command execution failed. The contents of the command’s ‘stderr’ can be found in the reason field.</td>
</tr>
<tr>
<td>DockerImageDoesNotExist</td>
<td>Thrown when a request is made involving a non-existent docker image.</td>
</tr>
<tr>
<td>DrConfigMethodFault</td>
<td>Root type for all DrConfig faults</td>
</tr>
<tr>
<td>DrConfigRuntimeFault</td>
<td>Base type for DrConfig faults extending RuntimeFault</td>
</tr>
<tr>
<td>DropDbSchemaFault</td>
<td>Thrown when there is an error during db schema drop.</td>
</tr>
<tr>
<td>ExcessiveTimeSkewFault</td>
<td>Thrown when the time skew between this machine</td>
</tr>
<tr>
<td>FailedToRetrieveUpdateFault</td>
<td>Thrown when check for update fails to retrieve available updates from the repository.</td>
</tr>
<tr>
<td>HostUnreachableFault</td>
<td>Thrown when the host could not be contacted.</td>
</tr>
<tr>
<td>IncompatibleVcFault</td>
<td>Occurs when attempting to connect to a Virtual Center server with incompatible protocol version.</td>
</tr>
<tr>
<td>InstallUpdateFailedFault</td>
<td>Thrown when install update fails.</td>
</tr>
<tr>
<td>InternalError</td>
<td>Occurs when the fault is not better described by a more specific DrConfig fault.</td>
</tr>
<tr>
<td>InvalidLocale</td>
<td>Thrown when a locale name is invalid or unavailable.</td>
</tr>
<tr>
<td>InvalidLogin</td>
<td>Thrown when the login is incomplete due to an incorrect token, user name, or password.</td>
</tr>
<tr>
<td>IncompatibleVmomisetServer</td>
<td>Thrown when a VMOMI server has an incompatible version.</td>
</tr>
<tr>
<td>InvalidCaCertificate</td>
<td>Thrown when provided certificate cannot be decoded from PEM format or when it is not CA certificate.</td>
</tr>
<tr>
<td>InvalidCertificate</td>
<td>Thrown when certificate is not suitable. For example when it is expired, not yet valid, contains a weak key etc.</td>
</tr>
<tr>
<td>InvalidNetworkConfiguration</td>
<td>Thrown when network configuration provided by client is not valid. For example when static IP address is configured but DNS is dynamic.</td>
</tr>
<tr>
<td>NoPermission</td>
<td>Thrown when an operation is denied because of a privilege not held on a managed object.</td>
</tr>
<tr>
<td>NotAuthenticated</td>
<td>Thrown when an operation is denied because the session has not yet successfully logged in.</td>
</tr>
<tr>
<td>NotAuthorized</td>
<td>Failed to authorize account.</td>
</tr>
<tr>
<td>OutOfBounds</td>
<td>Thrown if a parameter exceeds the acceptable range of values.</td>
</tr>
<tr>
<td>PrivateKeyNotFoundException</td>
<td>Thrown when certificate does not match the generated private key.</td>
</tr>
<tr>
<td>ReadDbStatusFault</td>
<td>Thrown when there is an error during db status read.</td>
</tr>
<tr>
<td>RecreateDbSchemaFault</td>
<td>Thrown when there is an error during db.</td>
</tr>
</tbody>
</table>
Table 9-1. Faults thrown by Site Recovery Manager Appliance Management functions (continued)

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceBusy</td>
<td>Thrown if another configuration task is already running.</td>
</tr>
<tr>
<td>ServiceIdle</td>
<td>Thrown when cancel is called and there is no running task.</td>
</tr>
<tr>
<td>ServiceNotFound</td>
<td>Thrown when an attempt is made to retrieve service information for a non-existent service.</td>
</tr>
<tr>
<td>SraOperationsDisabled</td>
<td>Thrown when SRA operation are disabled through the configuration. Used in VMC case.</td>
</tr>
<tr>
<td>SraUuidMismatch</td>
<td>Thrown when an operation involves two SRA images which have different vendor UUIDs. Some operations only make sense between SRA images which represent different version of the same SRA. For example, copying the SRA configuration. By the SRA spec, such images must have the same vendor UUID.</td>
</tr>
<tr>
<td>SrmAlreadyRunning</td>
<td>Thrown when configure is called on an already running SRM server.</td>
</tr>
<tr>
<td>SrmNotConfigured</td>
<td>Thrown when the operation's prerequisite of a configured SRM is not met.</td>
</tr>
<tr>
<td>SsoTokenNotAcquired</td>
<td>Thrown when SSO token could not be acquired.</td>
</tr>
<tr>
<td>SystemLogBundleNotFound</td>
<td>Thrown when an attempt is made to access a system log bundle that does not exist</td>
</tr>
<tr>
<td>UpdateNotAvailableFault</td>
<td>Thrown when install operation is invoked without available update.</td>
</tr>
<tr>
<td>UpgradeDbFault</td>
<td>Thrown when there is an error during db schema upgrade.</td>
</tr>
</tbody>
</table>

Faults in Site Recovery Manager API

This section lists the various faults thrown by the Site Recovery Manager APIs.

Table 9-2. Faults thrown by Site Recovery Manager functions

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentVmNotSupported</td>
<td>Thrown when an agent VM is used in an operation that does not support it.</td>
</tr>
<tr>
<td>AlreadyExists</td>
<td>The name, key, or identifier of the element already exists in the collection.</td>
</tr>
<tr>
<td>AlreadyLoggedInFault</td>
<td>The session is already logged in, and Login was called again.</td>
</tr>
<tr>
<td>AlreadyPairedFault</td>
<td>Thrown if an attempt to pair already paired SRM was made.</td>
</tr>
<tr>
<td>ArrayManagerInUse</td>
<td>Thrown when removing an array manager in use.</td>
</tr>
<tr>
<td>ArrayNotFound</td>
<td>Thrown when storage array with the specified ID cannot be found.</td>
</tr>
<tr>
<td>ArrayPairInUse</td>
<td>Thrown when removing an array pair in use.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ArrayPairNotFound</td>
<td>Thrown when existing array pair is not found in <code>discoverArrays</code> response.</td>
</tr>
<tr>
<td>CannotMapDvsUplinkPortgroup</td>
<td>Invalid DVPortgroup network specified for the mapping. Use this fault when site context is obvious from used mapping type (test mappings) or the operation itself.</td>
</tr>
<tr>
<td>CannotProtectDatastore</td>
<td>Base fault for failure to add a datastore to protection group. To fix remove the datastore from the protection group.</td>
</tr>
<tr>
<td>CannotUnprotectDatastoreInUse</td>
<td>Cannot remove a datastore from protection group because it is used by protected VM(s).</td>
</tr>
<tr>
<td>CannotProtectIFTSecondaryVm</td>
<td>Fault thrown when a user tries to protect a VM which is a fault tolerance secondary VM.</td>
</tr>
<tr>
<td>CannotProtectVm</td>
<td>Cannot add virtual machine to the protection group.</td>
</tr>
<tr>
<td>CertificateCtlSignatureNotValid</td>
<td>Thrown when a certificate trust list used to create this chain did not have a valid signature.</td>
</tr>
<tr>
<td>CommandFailed</td>
<td>Failed to execute SRA command.</td>
</tr>
<tr>
<td>ConfigFileNotReplicated</td>
<td>The VM's config file is located on a datastore which is either not replicated or not protected by the protection group in which the VM is being protected.</td>
</tr>
<tr>
<td>ConnectionDownFault</td>
<td>Thrown if the VMOMI connection to the remote server is down.</td>
</tr>
<tr>
<td>ConnectionLimitReached</td>
<td>Thrown when the configured connection limit has been reached.</td>
</tr>
<tr>
<td>DatastoreAlreadyProtected</td>
<td>Datastore cannot be added to this group because it is already part of another protection group.</td>
</tr>
<tr>
<td>DatastoreMissingProtection</td>
<td>Base fault for datastore missing from the protection group.</td>
</tr>
<tr>
<td>DatastoreNotReplicated</td>
<td>Cannot protect datastore because underlying storage devices are not configured for replication.</td>
</tr>
<tr>
<td>DatastoreProtectionIssue</td>
<td>Base fault for datastore-specific protection issues.</td>
</tr>
<tr>
<td>DependencyConflict</td>
<td>UpdateVmSettings operation was attempted that might cause a dependency cycle.</td>
</tr>
<tr>
<td>DeviceBackingConflict</td>
<td>If the caller specified a device locator, or explicitly excluded, a device which the provider would like to protect.</td>
</tr>
<tr>
<td>DeviceBackingConflict</td>
<td>Fault thrown when a user attempts to protect or reconfigure protection for a VM, specifying one or more backing locators which conflict with those chosen by the provider.</td>
</tr>
<tr>
<td>DeviceGroupMatchingFault</td>
<td>Reported when a device group cannot be matched to a peer device group.</td>
</tr>
<tr>
<td>DeviceMatchingFault</td>
<td>Reported when a device cannot be matched to a peer device.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DevicesNotResolved</td>
<td>Fault thrown when a user attempts to protect or reconfigure protection for a VM without resolving all of its devices</td>
</tr>
<tr>
<td>DirectionError</td>
<td>The direction of the recovery plan cannot be determined.</td>
</tr>
<tr>
<td>DomainDatastoreNotFound</td>
<td>No existing datastore could be matched to a fault domain.</td>
</tr>
<tr>
<td>DomainNotFound</td>
<td>No storage container could be matched to a fault domain. A storage container links the domain to its datastore.</td>
</tr>
<tr>
<td>DomainPeerNotFound</td>
<td>Domains does not have their peer domains reported by the paired SRM. One of the reasons may be no VASA provider is registered at the paired site.</td>
</tr>
<tr>
<td>DomainReplicationGroupNotFound</td>
<td>Replication group is part of a protection group but could not be found in the configuration.</td>
</tr>
<tr>
<td>DomainScNotFound</td>
<td>No storage container could be matched to a fault domain. The storage container links the domain to its datastore.</td>
</tr>
<tr>
<td>DuplicateArray</td>
<td>There is already another array manager that discovered a given array.</td>
</tr>
<tr>
<td>DuplicateName</td>
<td>Call is unable to determine which object to use due to name conflict.</td>
</tr>
<tr>
<td>GroupProtectionOverlapped</td>
<td>Thrown when more than one protection group contain VMs replicated by the same vVol replication groups. If recovery is started in this situation, all colocated VMs might be broken. To resolve this make sure that all VMs located on a certain replication group are protected by the same protection group.</td>
</tr>
<tr>
<td>GroupStateMissmatch</td>
<td>State of the protection group does not match the state of a replication group protected by it. This is possible when the replication group state is changed by an external tool. The error can be corrected by restoring the replication group to the expected state using the storage array instruments and tools.</td>
</tr>
<tr>
<td>GroupTargetsNotFound</td>
<td>Target replication groups cannot be found. This occurs when the group is not in the correct state. Check storage policy assignment.</td>
</tr>
<tr>
<td>IllegalMove</td>
<td>Thrown when a folder is moved to an invalid place in the folder hierarchy. This can be because the move would create a cycle or the destination is the wrong kind of container.</td>
</tr>
<tr>
<td>ImmutableFolder</td>
<td>Thrown when an operation is attempted upon a Folder that cannot be changed. For example, moving or deleting the root folder.</td>
</tr>
<tr>
<td>InsufficientLicensesFault</td>
<td>Thrown by a method that cannot acquire licenses for the object to create.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InternalError</td>
<td>An internal error occurred that cannot be described by a more specific fault or if the hbr provider cannot be found(null reference).</td>
</tr>
<tr>
<td>InvalidAdapterConnectionSpec</td>
<td>An AdapterConnectionSpec doesn't match the corresponding AdapterConnectionPrompt defined by the SRA.</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Base class for invalid argument exceptions. Thrown if the username format invalid, or user or user group does not exist, or if the user is a global vCenter administrator, or if the name of the protection group is empty or if the list of virtual machines is empty or null.</td>
</tr>
<tr>
<td>InvalidFolder</td>
<td>Thrown when a node is moved to an invalid place in the hierarchy. This can be because it is a child of the current node, or a wrong kind of container.</td>
</tr>
<tr>
<td>InvalidLogin</td>
<td>Cannot complete login due to an incorrect user name or password.</td>
</tr>
<tr>
<td>InvalidPrimaryFolder</td>
<td>Thrown for an attempt to create a primary site folder that cannot contain VMs.</td>
</tr>
<tr>
<td>InvalidPrimaryNetwork</td>
<td>Invalid primary network specified for mapping, such as uplink DVPortgroup</td>
</tr>
<tr>
<td>InvalidSecondaryFolder</td>
<td>Thrown for an attempt to create a secondary site folder that cannot contain VMs.</td>
</tr>
<tr>
<td>InvalidSecondaryNetwork</td>
<td>Invalid secondary network specified for mapping (such as uplink DVPortgroup)</td>
</tr>
<tr>
<td>InvalidState</td>
<td>Base class for invalid state exceptions</td>
</tr>
<tr>
<td>InvalidTokenLifetime</td>
<td>SSO token is either expired or not yet valid. This exception is generally thrown if there is a time skew between the local clock and the SSO server. This exception is not supported from methods with version prior to 8. However it is translated to Vim::Fault::InvalidLogin::Exception.</td>
</tr>
<tr>
<td>IpMappingFault</td>
<td>A base class for all IpMapping related faults. Thrown if problems are found while validating IP subnet mapping rules.</td>
</tr>
<tr>
<td>ManagedObjectNotFound</td>
<td>The destination folder does not exist.</td>
</tr>
<tr>
<td>MatchingFault</td>
<td>Reported when a device/device group cannot be matched to a peer device/device group.</td>
</tr>
<tr>
<td>MissingFolderMapping</td>
<td>Thrown when a user attempts to protect a VM, but the folder inventory mapping for the VM is not present in the InventoryMapper.</td>
</tr>
<tr>
<td>MissingFolderAndResourcePoolMapping</td>
<td>Thrown when a user attempts to protect a VM, but both the folder and resourcepool inventory mapping for the VM is not present in the InventoryMapper.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MissingInventoryMapping</td>
<td>Thrown when a user attempts to protect a VM, but the inventory mappings for the VM are not present in the InventoryMapper.</td>
</tr>
<tr>
<td>MissingNetworkMapping</td>
<td>Thrown when user attempts to add an IP mapping, but a network mapping between protectedNetwork and recoveryNetwork is missing.</td>
</tr>
<tr>
<td>MissingNetworkMappingEx</td>
<td>Thrown when a user attempts to protect a VM, but the network inventory mapping for the VM is not present in the InventoryMapper.</td>
</tr>
<tr>
<td>MissingResourcePoolMapping</td>
<td>Thrown when a user attempts to protect a VM, but the resourcePool inventory mapping for the VM is not present in the InventoryMapper.</td>
</tr>
<tr>
<td>MultipleFault</td>
<td>Multiple faults occur.</td>
</tr>
<tr>
<td>NetworkNotFound</td>
<td>Thrown if the test network mapping does not exist in the recovery plan.</td>
</tr>
<tr>
<td>NoPermission</td>
<td>Operation denied because of a privilege not held on a managed object.</td>
</tr>
<tr>
<td>NotEmpty</td>
<td>Thrown when an operation cannot be performed because the object is not empty.</td>
</tr>
<tr>
<td>NotSuitablePlaceholderDatastoreCluster</td>
<td>Thrown if the Placeholder Datastore Manager determines that a datastore should not be used as a placeholder datastore because it is not visible to all hosts in the cluster.</td>
</tr>
<tr>
<td>NotSuitablePlaceholderDatastore</td>
<td>Thrown if the Placeholder Datastore Manager determines that a datastore should not be used as a placeholder datastore for a variety of reasons, for example, it becomes replicated or is not visible to all hosts in the cluster. This fault is a parent for several other faults, which determine the exact reason why this datastore should not be used as a placeholder datastore.</td>
</tr>
<tr>
<td>NotSupported</td>
<td>ProtectionGroup is being moved into a folder whose childType() property is not set to the appropriate value. For example, a ProtectionGroup cannot be moved into a folder whose ChildType property value does not contain &quot;ProtectionGroup&quot;.</td>
</tr>
<tr>
<td>NotAuthenticated</td>
<td>Operation denied because the session has not successfully logged in.</td>
</tr>
<tr>
<td>PairOperationInProgress</td>
<td>Thrown when another pair, repair or break pairing operation is in progress and the request is rejected.</td>
</tr>
<tr>
<td>PeerArrayNotFound</td>
<td>Thrown when peer array with specified ID not found for a storage array.</td>
</tr>
<tr>
<td>PeerDeviceGroupNotMatched</td>
<td>Reported when a local device group cannot be matched to a remote peer device group.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PeerDeviceGroupNotStretched</td>
<td>Reported when a local stretched device group is matched to a remote peer device group but the local device group is not stretched.</td>
</tr>
<tr>
<td>PeerDeviceGroupWithoutStaticSitePreference</td>
<td>Reported when a local stretched device group is matched to a remote stretched peer device group but the local device group has no static site preference.</td>
</tr>
<tr>
<td>PeerDeviceNotMatched</td>
<td>Reported when a local device cannot be matched to a remote peer device.</td>
</tr>
<tr>
<td>PeerDeviceNotStretched</td>
<td>Reported when a local stretched device is matched to a remote peer device but the local device is not stretched.</td>
</tr>
<tr>
<td>PeerDeviceWithoutStaticSitePreference</td>
<td>Reported when a local stretched device is matched to a remote stretched peer device but the local device has no static site preference.</td>
</tr>
<tr>
<td>ProductionVmDeleted</td>
<td>Production VM was deleted.</td>
</tr>
<tr>
<td>PromptNotFound</td>
<td>Thrown when a RecoveryPrompt cannot be found.</td>
</tr>
<tr>
<td>ProtectionGroupNotEmpty</td>
<td>Thrown after attempt to remove a protection group that contains protected VMs.</td>
</tr>
<tr>
<td>ProtectionGroupNotFound</td>
<td>Thrown when an operation on protection group cannot find the protection group.</td>
</tr>
<tr>
<td>ProtectionIssue</td>
<td>Base class for faults describing protection group configuration issues.</td>
</tr>
<tr>
<td>ProtectionStillActive</td>
<td>Thrown when an attempt is made to remove a Primary/SecondarySite that is being replicated from/to.</td>
</tr>
<tr>
<td>ProviderFault</td>
<td>Thrown if either ReplicationProvider rejected the operation. This can occur when the settings are incorrect.</td>
</tr>
<tr>
<td>RecoveredDatastoreNotAvailableForPdm</td>
<td>Thrown if the Placeholder Datastore Manager determines that a datastore should not be used as a placeholder datastore because it was recovered.</td>
</tr>
<tr>
<td>RecoveryPlanLocked</td>
<td>An attempt was made to change a RecoveryPlan that is locked.</td>
</tr>
<tr>
<td>RecoveryPlanNotFound</td>
<td>Thrown when the requested recovery plan was not found.</td>
</tr>
<tr>
<td>RecoveryResultNotFound</td>
<td>Thrown when a RecoveryResult cannot be found.</td>
</tr>
<tr>
<td>RemotePeerDeviceGroupNotMatched</td>
<td>Reported when a remote device group cannot be matched to a local peer device group.</td>
</tr>
<tr>
<td>RemotePeerDeviceGroupNotStretched</td>
<td>Reported when a local stretched device group is matched to a remote peer device group but the remote device group is not stretched.</td>
</tr>
<tr>
<td>RemotePeerDeviceGroupWithoutStaticSitePreference</td>
<td>Reported when a local stretched device group is matched to a remote stretched peer device group but the remote device group has no static site preference.</td>
</tr>
<tr>
<td>RemotePeerDeviceNotMatched</td>
<td>Reported when a remote device cannot be matched to a local peer device.</td>
</tr>
</tbody>
</table>
Table 9-2. Faults thrown by Site Recovery Manager functions (continued)

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemotePeerDeviceNotStretched</td>
<td>Reported when a local stretched device is matched to a remote peer device but the remote device is not stretched.</td>
</tr>
<tr>
<td>RemotePeerDeviceWithoutStaticSitePreference</td>
<td>Reported when a local stretched device is matched to a remote stretched peer device but the remote device has no static site preference.</td>
</tr>
<tr>
<td>RemoteSiteNotAuthenticated</td>
<td>Thrown if the remote site or the session is not authenticated.</td>
</tr>
<tr>
<td>RemoteSiteNotEnabled</td>
<td>An attempt was made to use a remote site that is not enabled.</td>
</tr>
<tr>
<td>RemoteSiteNotInitialized</td>
<td>An attempt is made to use a remote site that is not initialized.</td>
</tr>
<tr>
<td>ReplicatedArrayPairAlreadyExists</td>
<td>Thrown when a replicated array pair already exists for a pair of arrays for a given SRA.</td>
</tr>
<tr>
<td>ReplicatedDatastoreNotAvailableForPdm</td>
<td>Thrown if the Placeholder Datastore Manager determines that a datastore should not be used as a placeholder datastore because it is replicated.</td>
</tr>
<tr>
<td>ReplicationProviderFault</td>
<td>Thrown when an unspecified error was returned from the replication provider.</td>
</tr>
<tr>
<td>SelfPairFault</td>
<td>Thrown when an attempt to pair with ourselves was made.</td>
</tr>
<tr>
<td>SitePairingFault</td>
<td>Thrown when one of the site pairing operations failed.</td>
</tr>
<tr>
<td>SnapshotDirectoryNotReplicated</td>
<td>The VM's snapshot directory is not replicated.</td>
</tr>
<tr>
<td>SourceDeviceGroupsWithStaticSitePreference</td>
<td>Reported when a local stretched device group is matched to a remote stretched peer device group but both the local device group and the remote device group have static site preference.</td>
</tr>
<tr>
<td>SourceDevicesWithStaticSitePreference</td>
<td>Reported when a local stretched device is matched to a remote stretched peer device but both the local device and the remote device have static site preference.</td>
</tr>
<tr>
<td>StorageAdapterNotFound</td>
<td>Thrown when storage adapter is not found.</td>
</tr>
<tr>
<td>StorageProviderFault</td>
<td>Corresponds to dr.vvolProvider.fault.StorageProviderFault.</td>
</tr>
<tr>
<td>StretchedDeviceGroupMatchingFault</td>
<td>Reported when a stretched device group is matched to a peer device group but only one device group is stretched.</td>
</tr>
<tr>
<td>StretchedDeviceMatchingFault</td>
<td>Reported when a stretched device is matched to a peer device but only one device is stretched.</td>
</tr>
<tr>
<td>StringArgumentTooLong</td>
<td>Thrown when a string argument exceeds (maxSize) characters</td>
</tr>
<tr>
<td>SuOperationInProgress</td>
<td>Thrown when another create, update or delete solution user operation is in progress and the request is rejected.</td>
</tr>
<tr>
<td>SuspendDirectoryNotReplicated</td>
<td>The VM's suspend directory is not replicated.</td>
</tr>
<tr>
<td>Fault</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SystemError</td>
<td>Thrown in case of internal SRM error.</td>
</tr>
<tr>
<td>TaskInProgress</td>
<td>Array cannot be found.</td>
</tr>
<tr>
<td>TestDatastoreNotAvailableForPdm</td>
<td>Thrown if the Placeholder Datastore Manager determines that a datastore should not be used as a placeholder datastore because it is created for test recovery.</td>
</tr>
<tr>
<td>UnableToFindPlaceholderDatastore</td>
<td>Thrown if the Placeholder Datastore Manager is unable to find a placeholder datastore for a host or a cluster.</td>
</tr>
<tr>
<td>UnknownPrimaryFolder</td>
<td>Secondary site tried operation on a folder that is nonexistent on primary site (protected site).</td>
</tr>
<tr>
<td>UnknownPrimaryNetwork</td>
<td>Thrown when the secondary site tries to perform an operation involving a network that does not exist on the primary site.</td>
</tr>
<tr>
<td>UnknownSecondaryNetwork</td>
<td>Secondary site tried operation on a network that is nonexistent on secondary site (recovery site).</td>
</tr>
<tr>
<td>UnknownPrimaryResourcePool</td>
<td>Secondary site tried operation on resource pool that is nonexistent on primary site.</td>
</tr>
<tr>
<td>UnknownSecondaryFolder</td>
<td>Primary site tried operation on a folder that is nonexistent on secondary site.</td>
</tr>
<tr>
<td>UnknownSecondaryNetwork</td>
<td>Primary site tried operation on a network that is nonexistent on secondary site.</td>
</tr>
<tr>
<td>UnknownSecondaryResourcePool</td>
<td>Primary site tried operation on resource pool that is nonexistent on secondary site.</td>
</tr>
<tr>
<td>VersionConflict</td>
<td>Attempt to reconfigure with a changeVersion that does not match the current value.</td>
</tr>
<tr>
<td>vim.fault.ConcurrentAccess</td>
<td>Thrown if another operation has modified the object and the change version no longer matches.</td>
</tr>
<tr>
<td>VmAlreadyProtected</td>
<td>Thrown when a user tries to protect a VM which is already protected by SRM.</td>
</tr>
<tr>
<td>VmAlreadyProtectedEx</td>
<td>Thrown when a user tries to protect a VM which is already protected by SRM. Contains extra info about the PG currently holding the VM protection.</td>
</tr>
<tr>
<td>VmFileNotReplicated</td>
<td>A virtual machine file is not replicated.</td>
</tr>
<tr>
<td>VmNotFoundInRecoveryPlan</td>
<td>Attempt to retrieve settings for virtual machine that does not exist in RecoveryPlan.</td>
</tr>
<tr>
<td>VmNotReplicated</td>
<td>No virtual machine files are replicated. Check and fix the storage policy assignment.</td>
</tr>
<tr>
<td>VmNotSupported</td>
<td>Thrown when the type VM used in an operation is not supported by the operation.</td>
</tr>
<tr>
<td>VmPiggybackError</td>
<td>A virtual machine is detected on a replication group that is already protected by one of the protection groups. Either protect the VM or reconfigure it out of the replication group.</td>
</tr>
</tbody>
</table>
Table 9-2. Faults thrown by Site Recovery Manager functions (continued)

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VmReconfigureRequired</td>
<td>The virtual machine configuration has changed and reconfiguration is required. Reconfigure is started automatically without user intervention.</td>
</tr>
<tr>
<td>VmReplicationGroupError</td>
<td>Attempting to protect a virtual machine that belongs to one replication group into a protection group that is configured to use a different set of replication groups.</td>
</tr>
<tr>
<td>VmSplitReplicated</td>
<td>In order to guarantee the consistency, SRM protects only virtual machines replicated by a single replication group. This situation occurs when the configuration file and the virtual disk files do not share the same storage policy assignments.</td>
</tr>
<tr>
<td>VmTemplateFault</td>
<td>vVol provider does not support protection of template VMs. This fault is throw when attempt to protect template VM with vVol provider is made.</td>
</tr>
<tr>
<td>VvolDomainFault</td>
<td>Base class for all fault domain related VvolProvider faults.</td>
</tr>
<tr>
<td>VvolProviderFault</td>
<td>Base class for all VvolProvider faults.</td>
</tr>
<tr>
<td>VvolVmFault</td>
<td>Base class for all virtual machine related VvolProvider faults.</td>
</tr>
<tr>
<td>WrongDrServerFault</td>
<td>Thrown if an attempt to repair connection of a paired SRM with wrong remote SRM.</td>
</tr>
</tbody>
</table>
SSL Certificates and SNMP Traps

This appendix contains information for the requirements and work with SSL certificates and Simple Network Management Protocol (SNMP) Traps.

This chapter includes the following topics:

- SSL Certificates
- SNMP Traps

SSL Certificates

The Site Recovery Manager uses SSL to encrypt communications between a client application and the server. The SSL certificate of the target server must reside on the client machine. To access the Web service programmatically, use its URN from a Web services client application.

The Web service listens to the following ports:

- Site Recovery Manager Appliance Management API
  Port - 5480
- Site Recover Manager APIs for Windows
  Port - 9086
- Site Recover Manager APIs for Photon Virtual Appliance (VA)
  Port - 443

Get vCenter Server Certificate

The Site Recovery Manager API or Site Recovery Manager Virtual Appliance is a secure Web service running on the Site Recovery Manager Server. To develop client applications, you must obtain the VMware vCenter Server certificate, which is used by the Site Recovery Manager Server, and import it into the certificate store of the workstation where you develop client applications.

Procedure

1. From your development workstation, open Internet Explorer (IE).

3 Click View Certificate.

4 Click Install Certificate to launch the Certificate Import wizard. Keep the default settings and click Next.

5 Click Finish. A security warning message displays concerning the certificate's certifying authority.

6 Click Yes.

   A Certificate Import wizard “success” message displays.

7 Click OK to dismiss the success message.

   The Certificate Properties page becomes active again.

8 Click OK in the Certificate dialog box to continue to the server.

   The initial Security Alert message presented in step 2 becomes active again.

9 Click Yes in the Security Alert message to continue with the original HTTPS request.

   The server Welcome page displays. The certificate is now installed in the IE certificate cache.

What to do next

Now that you have the certificate, your next task depends on what programming language you use to develop your client applications.

For C# developers, you can continue setting up your development environment by following the instructions at “Setting Up for Microsoft C# Development” in the Developer’s Setup Guide located at VMware’s Web site developer support page under the vSphere Web Services SDK.

For Java developers, you must export the certificates from the IE cache to a local directory. Minimize the IE browser window, and export the certificates as detailed in the following procedure.

Export Cached Certificates to a Local Directory

For Java development in a Windows environment, you must export the certificate to a local directory.

Procedure

1 Create a directory for the certificate, using the name set in the various batch files for the vSphere Web Services SDK: C:\VMware-Certs.

2 From the IE Tools menu, select Internet Options to open the Internet Options properties page.

3 Click the Content tab to activate the content advisor.

4 Click Certificates to open the Certificate manager.
5 Click the **Trusted Root Certificate Authorities** tab to display the list of trusted certificates.

6 Scroll through the list of certificates to find the certificate. For the vCenter Server, the certificate name is VMware.

7 Click the certificate to select it.

8 Click **Export...** to launch the Certificate Export Wizard.

9 Click **Next** to continue. The Export File Format dialog displays.

10 Keep the defaults (“DER encoded binary X.509 (.CER)”) and click **Next** to continue.

   The File To Export dialog displays, enabling you to enter a unique name for the certificate.

11 Choose a filename and enter it, along with the complete path to the directory: `C:\VMware-Certs\<servername>.cer`

   If you do not enter the complete path, the certificate is stored in your Documents and Settings folder.

12 Click **Next** to continue with the export.

   A Completing the Certificate Export Wizard page displays, summarizing the information about the certificate.

13 Click **Finish** to complete the export.

   A Certificate Export Wizard “success” message displays.

14 Click **OK** to dismiss the success message.

15 Click **Close**.

16 Click **Cancel** to close the Internet Options properties page.

**What to do next**

For more information about setting up your Java development environment, see “Setting Up for Java Development” in the Developer’s Setup Guide located at the VMware Web site developer support page under the vSphere Web Services SDK.

**About the Virtual Machine Keystore**

A Java KeyStore (JKS) is a repository of security certificates – either authorization certificates or public key certificates – used for SSL encryption and related activities. The Java Development Kit (JDK) maintains a keystore in `jre/lib/security/cacerts`, and provides the keytool command to manipulate it.

The `VMKEYSTORE` environment variable specifies the path to the JKS. The `run.sh` and `run.bat` scripts both refer to it. If you use the `--ignorecert` argument to run Java samples, you must still set the `VMKEYSTORE` variable, but you can set it to any location, not the actual JKS location.

Sample paths, Windows and Linux:
VMKEYSTORE=C:\VMware-Certs\vmware.keystore
VMKEYSTORE=/root/vmware-certs/vmware.keystore

SNMP Traps

Site Recovery Manager provides Simple Network Management Protocol (SNMP) traps that collect information sent by the API. All traps are compliant with the SNMPv1 type. Information provided by the traps can be used to initiate actions by client applications. Callers of the Site Recovery Manager API interface should listen for the SNMP traps. You might need to configure the vCenter Server to forward the SNMP traps to the registered SNMP Server. The MIB file is located in the following directory: <installdir>\www\VMWARE-SRM-TRAPS-5_0.MIB

There are two ways to generate SNMP traps from Site Recovery Manager. The first is the method presented here and in other Site Recovery Manager documentation. The second method to generate traps is by configuring SNMP actions on the events and alarms that Site Recovery Manager adds to vCenter Server. Alarms with SNMP traps configured are all raised using the generic alarm definition in VMWARE-VC-EVENT.mib. Consequently alarm-based traps do not have explicit definitions. To manage them, you would need to synthesize the trap, capture its contents, parse the trap, then determine how to filter it.

Look at MIB Names for SNMP Traps for more details.

MIB Names for SNMP Traps

The listed SNMP traps originate from the Site Recovery Manager, not from vCenter Server. Descriptions of SNMP traps are given according to their names in the MIB file. The names in this list can be prefaced by either vmwareSrm (Site Recovery Manager) or oidDr (object ID data recovery).

Table 10-1. SNMP Traps in the MIB

<table>
<thead>
<tr>
<th>SNMP Trap</th>
<th>What Trap Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecoveryPlanExecuteTestBegin</td>
<td>Signaled on the recovery site when a recovery test is initiated.</td>
</tr>
<tr>
<td>RecoveryPlanExecuteTestEnd</td>
<td>Signaled on the recovery site when a recovery test has completed. If an error occurred it is available as [data.Error]</td>
</tr>
<tr>
<td>RecoveryPlanExecuteBegin</td>
<td>Signaled on the recovery site when a recovery is initiated.</td>
</tr>
<tr>
<td>RecoveryPlanExecuteEnd</td>
<td>Signaled on the recovery site when a recovery has completed. If an error occurred it is available as [data.Error]</td>
</tr>
<tr>
<td>RecoveryVmBegin</td>
<td>Signaled when the recovery virtual machine was successfully created. If an error occurs before the virtual machine’s ID is known, the event is not fired.</td>
</tr>
<tr>
<td>RecoveryVmEnd</td>
<td>Signaled after the last post-power on script has completed, or after a recovery-stopping error has occurred for the virtual machine.</td>
</tr>
</tbody>
</table>
## Table 10-1. SNMP Traps in the MIB (continued)

<table>
<thead>
<tr>
<th>SNMP Trap</th>
<th>What Trap Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecoveryPlanPromptDisplay</td>
<td>The recovery plan is displaying prompt [data.PromptKey] and is waiting for user input. PromptKey is a unique identifier.</td>
</tr>
<tr>
<td>RecoveryPlanPromptResponse</td>
<td>The recovery plan received an answer to prompt [data.PromptKey] and is no longer paused waiting for user input.</td>
</tr>
<tr>
<td>RecoveryPlanServerCommandBegin</td>
<td>Signaled on the recovery site when Site Recovery Manager starts to run a Callout command on the Site Recovery Manager server.</td>
</tr>
<tr>
<td>RecoveryPlanServerCommandEnd</td>
<td>Signaled on the recovery site when Site Recovery Manager has finished running a Callout command on the Site Recovery Manager server.</td>
</tr>
<tr>
<td>RecoveryPlanVmCommandBegin</td>
<td>Signaled on the recovery site when Site Recovery Manager has started to run a Callout command on a recovered virtual machine.</td>
</tr>
<tr>
<td>RecoveryPlanVmCommandEnd</td>
<td>Signaled on the recovery site when Site Recovery Manager has finished running a Callout command on a recovered virtual machine.</td>
</tr>
<tr>
<td>RecoveryPlanExecuteReprotectBegin</td>
<td>Signaled on the recovery site when a reprotect is initiated.</td>
</tr>
<tr>
<td>RecoveryPlanExecuteReprotectEnd</td>
<td>Signaled on the recovery site when a reprotect has completed. If an error occurred it is available as [data.Error]</td>
</tr>
<tr>
<td>RecoveryPlanExecuteCleanupBegin</td>
<td>Signaled on the recovery site when a test cleanup is initiated.</td>
</tr>
<tr>
<td>RecoveryPlanExecuteCleanupEnd</td>
<td>Signaled on the recovery site when a test cleanup has completed. If an error occurred it is available as [data.Error]</td>
</tr>
</tbody>
</table>

### Configuring SNMP Receivers in vCenter Server

For a simple procedure to configure SNMP receivers, see the section “Configure SNMP Settings in the vSphere Web Client” in the vSphere vCenter Server and Host Management manual, available in the VMware vSphere 5.5 Documentation Center. For details about configuring the SNMP receiver URL, receiver port, and community, see the section “Configure SNMP Settings for vCenter Server by Using the vSphere Web Client” in the vSphere Monitoring and Performance manual, also in the VMware vSphere 5.5 Documentation Center.

### SNMP Traps and Object IDs

The MIB objects are listed below with IDs, then the SNMP traps themselves with IDs.

#### Table 10-2. MIB objects with IDs

<table>
<thead>
<tr>
<th>MIB_OBJECT</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>oidDrVmName</td>
<td>1.3.6.1.4.1.6876.51.1.1</td>
</tr>
<tr>
<td>oidDrRecoveryName</td>
<td>1.3.6.1.4.1.6876.51.1.2</td>
</tr>
</tbody>
</table>
### Table 10-2. MIB objects with IDs (continued)

<table>
<thead>
<tr>
<th>MIB_OBJECT</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>oidDrPromptString</td>
<td>1.3.6.1.4.1.6876.51.1.3</td>
</tr>
<tr>
<td>oidDrRecoveryType</td>
<td>1.3.6.1.4.1.6876.51.1.4</td>
</tr>
<tr>
<td>oidDrRecoveryState</td>
<td>1.3.6.1.4.1.6876.51.1.5</td>
</tr>
<tr>
<td>oidDrSiteString</td>
<td>1.3.6.1.4.1.6876.51.1.6</td>
</tr>
<tr>
<td>oidDrVmUuid</td>
<td>1.3.6.1.4.1.6876.51.1.7</td>
</tr>
<tr>
<td>oidDrResult</td>
<td>1.3.6.1.4.1.6876.51.1.8</td>
</tr>
<tr>
<td>oidDrCommandName</td>
<td>1.3.6.1.4.1.6876.51.1.9</td>
</tr>
</tbody>
</table>

### Table 10-3. SMNP traps with IDs

<table>
<thead>
<tr>
<th>MIB_TRAP</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecoveryPlanExecuteTestBegin</td>
<td>1.3.6.1.4.1.6876.51.0.1</td>
</tr>
<tr>
<td>RecoveryPlanExecuteTestEnd</td>
<td>1.3.6.1.4.1.6876.51.0.2</td>
</tr>
<tr>
<td>RecoveryPlanExecuteBegin</td>
<td>1.3.6.1.4.1.6876.51.0.3</td>
</tr>
<tr>
<td>RecoveryPlanExecuteEnd</td>
<td>1.3.6.1.4.1.6876.51.0.4</td>
</tr>
<tr>
<td>RecoveryVmBegin</td>
<td>1.3.6.1.4.1.6876.51.0.5</td>
</tr>
<tr>
<td>RecoveryVmEnd</td>
<td>1.3.6.1.4.1.6876.51.0.6</td>
</tr>
<tr>
<td>RecoveryPlanPromptDisplay</td>
<td>1.3.6.1.4.1.6876.51.0.7</td>
</tr>
<tr>
<td>RecoveryPlanPromptResponse</td>
<td>1.3.6.1.4.1.6876.51.0.8</td>
</tr>
<tr>
<td>RecoveryPlanServerCommandBegin</td>
<td>1.3.6.1.4.1.6876.51.0.9</td>
</tr>
<tr>
<td>RecoveryPlanServerCommandEnd</td>
<td>1.3.6.1.4.1.6876.51.0.10</td>
</tr>
<tr>
<td>RecoveryPlanVmCommandBegin</td>
<td>1.3.6.1.4.1.6876.51.0.11</td>
</tr>
<tr>
<td>RecoveryPlanVmCommandEnd</td>
<td>1.3.6.1.4.1.6876.51.0.12</td>
</tr>
<tr>
<td>RecoveryPlanExecuteReprotectBegin</td>
<td>1.3.6.1.4.1.6876.51.0.13</td>
</tr>
<tr>
<td>RecoveryPlanExecuteReprotectEnd</td>
<td>1.3.6.1.4.1.6876.51.0.14</td>
</tr>
<tr>
<td>RecoveryPlanExecuteCleanupBegin</td>
<td>1.3.6.1.4.1.6876.51.0.15</td>
</tr>
<tr>
<td>RecoveryPlanExecuteCleanupEnd</td>
<td>1.3.6.1.4.1.6876.51.0.16</td>
</tr>
</tbody>
</table>