Site Recovery Manager API Developer’s Guide

VMware vCenter Site Recovery Manager 5.8

EN-001401-00

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About This Book

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

This manual provides information about old and new interfaces in the Site Recovery Manager 5.0 and 5.8 releases, for developers who are interested in automating site recovery tasks.

Revision History

This book is revised with each product release or when necessary. Table 1 summarizes changes in each release.

Table 1. Revision History

Revision Date	Description
9 September 2014	Revised for the 5.8 release to include protection group APIs and improved SDK support for Java.
4 October 2013	Added cleanupTest mode on page 8, and more about SNMP traps.
23 May 2013	Specified WSDL port number on page 12.
27 January 2012	Added Associated vCenter Server on page 12, corrected array based replication on pages 22-23.
17 November 2011	Updated *Site Recovery Manager API Developer's Guide* published after the 5.0 release.
16 June 2008	Technical note entitled *Site Recovery Manager API* published with the 1.0 release.

Intended Audience

This book is intended for developers who need 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 replication, recovery, and reprotection 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 as well as other VMware API and SDK public documentation, go to http://www.vmware.com/support/pubs/sdk_pubs.html.

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.

Document Feedback

VMware welcomes your suggestions for improving our documentation. Send your feedback to docfeedback@vmware.com.

The Site Recovery Manager API

1

This manual describes a Web services programming interface to protection groups and recovery plans in VMware vCenter Site Recovery Manager (SRM). This chapter presents the following topics.

- "Introduction to the API" on page 9
- "List of API Operations" on page 10
- "Managed Object Hierarchy" on page 12
- "WSDL Programming Environments" on page 14
- "Accessing vCenter Site Recovery Manager" on page 16

Introduction to the API

The SRM API provides an interface similar to the vSphere API, which is an object-oriented Web service that provides access to vSphere and virtual machine management on vCenter Server and ESXi hosts. You program the vSphere API in Java, C#, or any language that supports Web services definition language (WDSL).

The SRM API offers a way for third party systems to initiate test, recovery, reprotect, or revert operations and collect the results. Protection groups and recovery plans are usually set up from the user interface (UI).

API Releases

The SRM 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 SRM 5.8 release introduces 30 methods and four new managed objects, adding several requested features to the API, including:

- 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
- Ability to modify selected fields in a virtual machine's recovery settings

Terminology

This document uses the following terms as defined below:

- **SOAP** – Client applications invoke operations by sending SOAP formatted messages. When passing data objects between client and server, you need to build or parse a SOAP message that contains the data object properties as XML elements corresponding to the message structures described in 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 Recover Manager API is similar to, and derived from, the vSphere API. For information about the vSphere API, see the *vSphere Web Services SDK Programming Guide* and the *vSphere API Reference*, both available on the VMware Web site.

List of API Operations

Table 1-1 provides a list of Site Recovery Manager APIs organized by approximate order of use.

Table 1-1. List of API operations by function

Method	Description of Operation
Authentication	
SrmLoginLocale	Begin a session with Site Recovery Manager Server.
SrmLoginSites	Log in to both the local and remote vCenter Servers.
SrmLogoutLocale	Log out sites and terminate the current session.
SrmLoginRemoteSite	Log in to remote site when escalated privileges are required on the remote site and the current session has already been authenticated using SrmLoginSites.
Service Instance	
RetrieveContent	Retrieve the properties of a service instance.
Inventory Mapping	
AddFolderMapping	Add a folder mapping between the primary and secondary vCenter Servers.
AddNetworkMapping	Add a network mapping between the primary and secondary vCenter Servers.
AddResourcePoolMapping	Add resource pool mapping between primary and secondary vCenter Servers.
Protection	
ListProtectionGroups	Get a list of the protection groups that are currently configured.
ListInventoryMappings	Get a list of the configured inventory mappings on the protection site.
ListReplicatedDatastores	Get a list of the replicated datastores.
GetProtectionGroupRootFolder	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.
ListUnassignedReplicatedDatastores	Get list of replicated datastores that can be used to create new protection groups.
ProtectionListProtectedDatastores	Get list of replicated datastores that are protected by SRM.
ListUnassignedReplicatedVms	Get list of replicated VMs not currently assigned to an SRM protection group.
ProtectionListProtectedVms	Get list of virtual machines that are protected by SRM.
CreateAbrProtectionGroup	Create an array-based replication (ABR) protection group.
CreateHbrProtectionGroup	Create a host-based replication (HBR) protection group.
Protection Group Folder	Methods to navigate folder hierarchy and retrieve specific protection groups.
ListChildProtectionGroupFolders	Return the child ProtectionGroupFolders located in this folder.
ListChildProtectionGroups	Return the SrmProtectionGroup objects located in this folder.
GetProtectionGroup	Return a reference to the protection group.
GetName	Get name of folder object, given ProtectionGroupFolder or RecoveryPlanFolder.
GetParentFolder	Get parent folder for a ProtectionGroupFolder or RecoveryPlanFolder.

Table 1-1. List of API operations by function (Continued)

Method	Description of Operation
Create Protection Group Task	Methods to track progress and completion of create protection group calls.
IsCreateProtectionGroupComplete	Check completeness of the task to create a new protection group.
GetCreateProtectionGroupResult	Once task is complete, check the result to ensure that it succeeded.
GetNewProtectionGroup	After checking task result, obtain the newly created SrmProtectionGroup.
Protection Group	
GetInfo	Retrieve basic information about this protection group.
ProtectionGroupGetParentFolder	Retrieve the folder that contains this protection group.
GetPeer	Retrieve the peer protection group.
ListProtectedVms	List VMs protected in this group with information about their protection state.
ListProtectedDatastores	Retrieve a list of the Datastores protected by this protection group.
ListAssociatedVms	Retrieve a list of VMs associated with this group. Only for vSphere Replication.
GetProtectionState	Get the current state of the protection group.
ProtectionGroupListRecoveryPlans	Retrieve a list of all Recovery Plans this protection group is a member of.
ProtectionGroupQueryVmProtection	Determine whether the specified VMs can be or currently are protected, which must be mapped to recovery site as per ListInventoryMappings.
ProtectVms	Protect the specified VMs. Each virtual machine's folder, resource pool, and network must be mapped to the recovery site.
UnprotectVms	Unprotect the specified VMs.
AssociateVms	Associate the specified VMs with a group. A prerequisite for protection. Only for vSphere Replication.
UnassociateVms	Unassociate the specified VMs with this group. Only for vSphere Replication.
Protection Task	
GetProtectionStatus	Get the results of ProtectVms or UnprotectVms.
GetTasks	Get Task information from the vCenter Server for each virtual machine that was requested to be protected or unprotected.
IsComplete	Check if this Task has finished.
GetResult	Get the results of this Task.
Recovery	
ListPlans	Retrieve all the Recovery Plans for this Site Recovery Manager Server.
GetHistory	Retrieve the history for a given Recovery Plan.
GetRecoveryPlanRootFolder	Retrieve the root folder for all Recovery Plans.
Recovery Plan Folder	Methods to traverse the folder hierarchy for Recovery Plans.
ListChildRecoveryPlanFolders	Return the child RecoveryPlanFolders located in this folder.
ListChildRecoveryPlans	Return an array of SrmRecoveryPlan objects located in this folder,
GetRecoveryPlan	Get MoRef to Recovery Plan with specified name in this RecoveryPlanFolder.
GetName, GetParentFolder	See ProtectionGroupFolder above.
Recovery Plan	
RecoveryPlanGetInfo	Retrieve basic information about the specified Recovery Plan.
RecoveryPlanGetPeer	Get the peer plan for this Recovery Plan. The returned object refers to a plan at the paired site, not the local site.
Start	Start the Recovery Plan in a selected mode: test, cleanupTest, recovery, reprotect, or revert. Requires Run privilege for tests, Failover privilege for the others.
Cancel	Cancel the specified Recovery Plan.

Table 1-1. List of API operations by function (Continued)

Method	Description of Operation
ListPrompts	List the current prompts that are waiting for input. When a prompt step is reached, the plan goes into waiting state until AnswerPrompt is received. Prompts are given in the same order in which VMs are scheduled to start up.
AnswerPrompt	Answer the current prompt displayed by a Recovery Plan. Requires the Run privilege for test, or the Failover privilege for the other modes.
RecoveryPlanGetParentFolder	Retrieve the root folder for all Recovery Plans.
GetRecoverySettings	Retrieve the per-VM recovery settings for VMs in the Recovery Plan.
SetRecoverySettings	Modify the per-VM recovery settings for VMs in the Recovery Plan.
AddProtectionGroup	Add a protection group to this Recovery Plan.
Recovery History	
GetRecoveryResult	Retrieve the recovery result for a given run of a Recovery Plan.
GetResultCount	Retrieve total number of stored results, including both Recovery and peer plans.
GetResultLength	Get length of XML result document for the requested recovery result.
RetrieveStatus	Retrieve XML document for an historical run of the specified Recovery Plan.
Deprecated 1.0 Methods	
SrmLogin, SrmLogout	Log in to and out of Site Recovery Manager Server.
GetApiVersion	Obtain the API version.
ListRecoveryPlans	Get a list of Recovery Plans at the SRM site.
RecoveryPlanSettings	Get the settings of a specific Recovery Plan at the SRM site.
RecoveryPlanStart	Start a specific Recovery Plan in recovery or test mode.
RecoveryPlanPause	Pause a running Recovery Plan.
RecoveryPlanResume	Restart a paused Recovery Plan.
RecoveryPlanAnswerPrompt	Answer a prompt.
RecoveryPlanCancel	Cancel a Recovery Plan.
GetFinalStatus	Get the final status of a Recovery Plan.

Managed Object Hierarchy

Table 1-2 shows the SRM managed object hierarchy with each managed object's methods, in alphabetic order.

Table 1-2. Managed object hierarchy

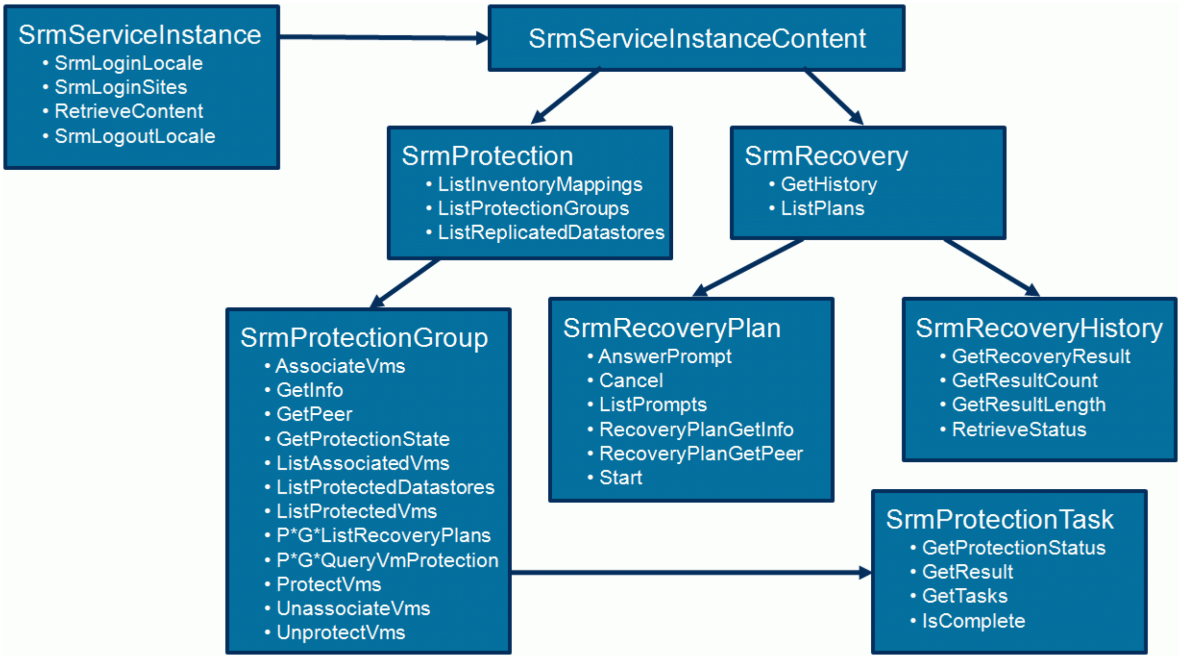
Managed Object	Remarks	Local Methods
SrmApi	Old version 1.0 API, still provided for backward compatibility	GetApiVersion, GetFinalStatus, ListRecoveryPlans, RecoveryPlanAnswerPrompt, RecoveryPlanCancel, RecoveryPlanPause, RecoveryPlanResume, RecoveryPlanSettings, RecoveryPlanStart, SrmLogin, SrmLogout
SrmProtection	List inventory mappings, query protection groups	ListInventoryMappings, ListProtectionGroups, ListReplicatedDatastores New in 5.8: CreateAbrProtectionGroup, CreateHbrProtectionGroup, GetProtectionGroupRootFolder, ListUnassignedReplicatedDatastore, ListUnassignedReplicatedVms, ProtectionListProtectedDatastores, ProtectionListProtectedVms

Table 1-2. Managed object hierarchy

Managed Object	Remarks	Local Methods
SrmProtectionGroup	Add virtual machines to a protection group, get peer, query protected datastores	AssociateVms, GetInfo, GetPeer, GetProtectionState, ListAssociatedVms, ListProtectedDatastores, ListProtectedVms, ProtectionGroupListRecoveryPlans, ProtectionGroupQueryVmProtection, ProtectVms, UnassociateVms, UnprotectVms New in 5.8: ProtectionGroupGetParentFolder
SrmProtectionTask	Get task status for protection groups	GetProtectionStatus, GetResult, GetTasks, IsComplete
SrmRecovery	Query recovery plans	GetHistory, ListPlans New in 5.8: GetRecoveryPlanRootFolder
SrmRecoveryHistory	Recovery plan status	GetRecoveryResult, GetResultCount, GetResultLength, RetrieveStatus
SrmRecoveryPlan	Run a recovery plan	AnswerPrompt, Cancel, ListPrompts, RecoveryPlanGetInfo, RecoveryPlanGetPeer, Start New in 5.8: AddProtectionGroup, GetRecoverySettings, RecoveryPlanGetParentFolder, SetRecoverySettings
SrmServiceInstance	Open and close session	RetrieveContent, SrmLoginLocale, SrmLoginSites, SrmLogoutLocale New in 5.8: GetPairedSite, GetSiteName, SrmLoginRemoteSite
InventoryMapping		New in 5.8: AddFolderMapping, AddNetworkMapping, AddResourcePoolMapping
Folder		New in 5.8: GetName, GetParentFolder
ProtectionGroupFolder		New in 5.8: GetProtectionGroup, ListChildProtectionGroupFolders, ListChildProtectionGroups
RecoveryPlanFolder		New in 5.8: GetRecoveryPlan, ListChildRecoveryPlanFolders, ListChildRecoveryPlans
CreateProtectionGroup Task		New in 5.8: GetCreateProtectionGroupResult, GetNewProtectionGroup, IsCreateProtectionGroupComplete

Figure 1-1, "SRM Object Hierarchy," on page 14 shows the managed object class hierarchy (unfortunately not yet updated for SRM 5.8) with each managed object's methods.

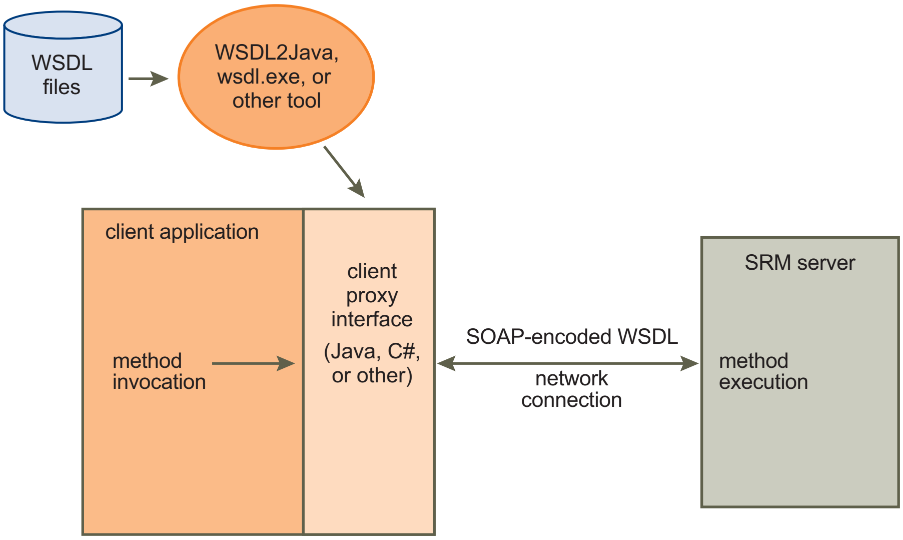
Figure 1-1. SRM Object Hierarchy



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.

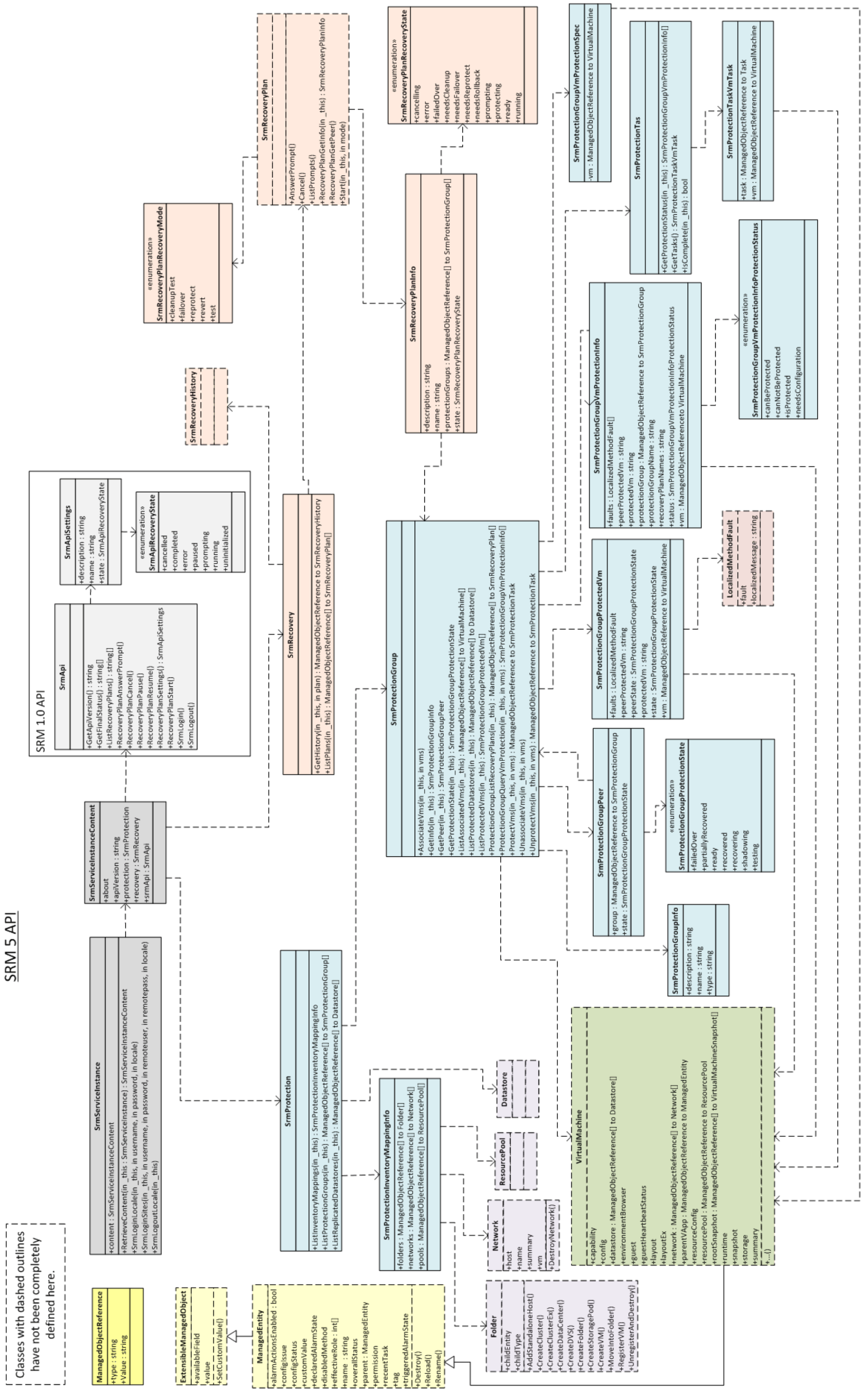
Figure 1-2. WSDL programming components



SRM Objects as WSDL

Figure 1-3, "SRM managed objects," on page 15 shows SRM managed entities and class relationships as they fit into the VMware WSDL scheme. Classes with dashed lines are vSphere managed objects; for information about them, see the *vSphere API Reference* manual.

Figure 1-3. SRM managed objects


SRM 5 API
Classes with dashed outlines have not been completely defined here.
ManagedObjectReference
+type : string
+Value : string
ExtensibleManagedObject
+availableField
+value
+SetCustomValue()
ManagedEntity
+alarmActionsEnabled : bool
+configIssue
+configStatus
+customValue
+declaredAlarmState
+disabledMethod
+effectiveRole : int[]
+name : string
+overallStatus
+parent : ManagedEntity
+permission
+recentTask
+tag
+triggeredAlarmState
+Destroy()
+Reload()
+Rename()
Folder
+childEntity
+childType
+AddStandaloneHost()
+CreateCluster()
+CreateClusterEx()
+CreateDatacenter()
+CreateDVS()
+CreateFolder()
+CreateStoragePod()
+CreateVM()
+MoveIntoFolder()
+RegisterVM()
+UnregisterAndDestroy()
Network
+host
+name
+summary
+vm
+DestroyNetwork()
ResourcePool
Datastore
VirtualMachine
+capability
+config
+datastore : ManagedObjectReference[] to Datastore[]
+environmentBrowser
+guest
+guestHeartbeatStatus
+layout
+layoutEx
+network : ManagedObjectReference[] to Network[]
+parentVApp : ManagedObjectReference to ManagedEntity
+resourceConfig
+resourcePool : ManagedObjectReference to ResourcePool
+rootSnapshot : ManagedObjectReference[] to VirtualMachineSnapshot[]
+runtime
+snapshot
+storage
+summary
+...()
SrmServiceInstance
+content : SrmServiceInstanceContent
+RetrieveContent(in _this : SrmServiceInstance) : SrmServiceInstanceContent
+SrmLoginLocale(in _this, in username, in password, in locale)
+SrmLoginSites(in _this, in username, in password, in remoteuser, in remotepass, in locale)
+SrmLogoutLocale(in _this)
SrmServiceInstanceContent
+about
+apiVersion : string
+protection : SrmProtection
+recovery : SrmRecovery
+srmApi : SrmApi
SRM 1.0 API
SrmApi
+GetApiVersion() : string
+GetFinalStatus() : string[]
+ListRecoveryPlans() : string[]
+RecoveryPlanAnswerPrompt()
+RecoveryPlanCancel()
+RecoveryPlanPause()
+RecoveryPlanResume()
+RecoveryPlanSettings() : SrmApiSettings
+RecoveryPlanStart()
+SrmLogin()
+SrmLogout()
SrmApiSettings
+description : string
+name : string
+state : SrmApiRecoveryState
«enumeration»
SrmApiRecoveryState
+cancelled
+completed
+error
+paused
+prompting
+running
+uninitialized
SrmProtection
+ListInventoryMappings(in _this) : SrmProtectionInventoryMappingInfo
+ListProtectionGroups(in _this) : ManagedObjectReference[] to SrmProtectionGroup[]
+ListReplicatedDatastores(in _this) : ManagedObjectReference[] to Datastore[]
SrmProtectionInventoryMappingInfo
+folders : ManagedObjectReference[] to Folder[]
+networks : ManagedObjectReference[] to Network[]
+pools : ManagedObjectReference[] to ResourcePool[]
SrmRecovery
+GetHistory(in _this, in plan) : ManagedObjectReference to SrmRecoveryHistory
+ListPlans(in _this) : ManagedObjectReference[] to SrmRecoveryPlan[]
SrmRecoveryHistory
«enumeration»
SrmRecoveryPlanRecoveryMode
+cleanupTest
+failover
+reprotect
+revert
+test
SrmRecoveryPlan
+AnswerPrompt()
+Cancel()
+ListPrompts()
+RecoveryPlanGetInfo(in _this) : SrmRecoveryPlanInfo
+RecoveryPlanGetPeer()
+Start(in _this, in mode)
«enumeration»
SrmRecoveryPlanRecoveryState
+cancelling
+error
+failedOver
+needsCleanup
+needsFailover
+needsReprotect
+needsRollback
+prompting
+protecting
+ready
+running
SrmRecoveryPlanInfo
+description : string
+name : string
+protectionGroups : ManagedObjectReference[] to SrmProtectionGroup[]
+state : SrmRecoveryPlanRecoveryState
SrmProtectionGroup
+AssociateVms(in _this, in vms)
+GetInfo(in _this) : SrmProtectionGroupInfo
+GetPeer(in _this) : SrmProtectionGroupPeer
+GetProtectionState(in _this) : SrmProtectionGroupProtectionState
+ListAssociatedVms(in _this) : ManagedObjectReference[] to VirtualMachine[]
+ListProtectedDatastores(in _this) : ManagedObjectReference[] to Datastore[]
+ListProtectedVms(in _this) : SrmProtectionGroupProtectedVm[]
+ProtectionGroupListRecoveryPlans(in _this) : ManagedObjectReference[] to SrmRecoveryPlan[]
+ProtectionGroupQueryVmProtection(in _this, in vms) : SrmProtectionGroupVmProtectionInfo[]
+ProtectVms(in _this, in vms) : ManagedObjectReference to SrmProtectionTask
+UnassociateVms(in _this, in vms)
+UnprotectVms(in _this, in vms) : ManagedObjectReference to SrmProtectionTask
SrmProtectionGroupPeer
+group : ManagedObjectReference to SrmProtectionGroup
+state : SrmProtectionGroupProtectionState
SrmProtectionGroupInfo
+description : string
+name : string
+type : string
«enumeration»
SrmProtectionGroupProtectionState
+failedOver
+partiallyRecovered
+ready
+recovered
+recovering
+shadowing
+testing
SrmProtectionGroupProtectedVm
+faults : LocalizedMethodFault
+peerProtectedVm : string
+peerState : SrmProtectionGroupProtectionState
+protectedVm : string
+state : SrmProtectionGroupProtectionState
+vm : ManagedObjectReference to VirtualMachine
LocalizedMethodFault
+fault
+localizedMessage : string
SrmProtectionGroupVmProtectionInfo
+faults : LocalizedMethodFault[]
+peerProtectedVm : string
+protectedVm : string
+protectionGroup : ManagedObjectReference to SrmProtectionGroup
+protectionGroupName : string
+recoveryPlanNames : string
+status : SrmProtectionGroupVmProtectionInfoProtectionStatus
+vm : ManagedObjectReferenceto VirtualMachine
«enumeration»
SrmProtectionGroupVmProtectionInfoProtectionStatus
+canBeProtected
+canNotBeProtected
+isProtected
+needsConfiguration
SrmProtectionTas
+GetProtectionStatus(in _this) : SrmProtectionGroupVmProtectionInfo[]
+GetTasks() : SrmProtectionTaskVmTask
+IsComplete(in _this) : bool
SrmProtectionTaskVmTask
+task : ManagedObjectReference to Task
+vm : ManagedObjectReference to VirtualMachine
SrmProtectionGroupVmProtectionSpec
+vm : ManagedObjectReference to VirtualMachine

C# and Visual Studio

The SRM 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.

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 SRM specific WSDL.

Java Axis Client

The SRM 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.

Accessing vCenter Site Recovery Manager

The SRM API provides language-neutral interfaces to the VMware vCenter Site Recovery Manager (SRM) server management framework. Interfaces are provided for managing protection groups and recovery plans. Both array based replication and vSphere Replication are supported.

Location of the API

The API is located in the following directory:

- `<installdir>\www`

The API uses the following default listener port:

- SOAP – 9007

You can obtain the WSDL for the SRM API by requesting the file `/srm.wsdl` from the WSDL port 9085. (The WSDL port number changed since the previous release.)

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 the WS-I Basic Profile 1.0, see the http://www.ws-i.org Web site.

Associated vCenter Server

The Site Recovery Manager API currently does not allow you to get the location of its associated vCenter Server, however you can find the SRM location from the vCenter Server using the vim.extensionManager – look for the `com.vmware.vcDr` extension.

In the returned extensionList URL, use the `https` address following the vertical bar, probably on port 8095, ending in `/dr`. Port 8095 is the SOAP port for Site Recovery Manager Server's internal API, while port 9007 is the SOAP port for Site Recovery Manager Server's external API. Port 8095 serves internal APIs used by SRM client, and is proxied from vCenter Server with `http` on port 80 or `https` on port 8095. SRM clients can create a tunneled SSL connection through either port. Port 9007 serves APIs used by external customers so it has no proxy from vCenter Server. External clients must create an SSL connection directly to port 9007.

You log in to SRM using the same credentials as for vCenter Server. Both SOAP ports 9007 and 8095 verify against these credentials.

SDK Installation and Setup 2

This chapter describes how to unpack and use the SDK, and includes the following sections:

Contents of the SDK Package

You can obtain the SDK package by navigating to http://www.vmware.com/support/developer/srm-api and clicking the **Download SDK** link.

You will need to provide an email address or customer number, with valid password, for authentication on the SRM download site.

The SRM SDK is delivered as a ZIP archive (`VMware-srm-sdk-<version>-<build>.zip`) containing the following items:

- The WSDL and XML schema files that define the API for Site Recovery Manager Server management.
- Sample C# code for .NET and Java code for JAX-WS or Axis showing how to display a recovery plan.
- Batch files and shell scripts to automate the process of generating client-side stubs, and for rebuilding the sample applications. For C# developers, Visual Studio project and solution files are included.
- Documentation, including the *VMware Infrastructure SDK Reference Guide* (the previous name for the *vSphere API Reference*) with language-neutral descriptive information about object type definitions, properties, and method signatures for the SRM API.

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 2-1. SDK directory structure

Directory or File	Description
/doc	Contains SDK README files and reference documentation for the SDK.
/doc/ReferenceGuide	API Reference for the VMware vCenter Site Recovery Manager API.
/doc/ReferenceGuide/index.html	To view the API Reference, open `index.html` with a Web browser.
/doc/SDK_Terms_and_Conditions.*	End user license agreement for the Site Recovery Manager SDK.

Table 2-1. SDK directory structure (Continued)

Directory or File	Description
/samples	Top-level directory for language-specific versions of sample client applications.
/samples/DotNet/	Directory containing command scripts to generate the .NET proxy classes and Web service stubs. The `GeneratingStubs.txt` file gives helpful notes about how to generate stubs with your own namespace for Visual Studio 2008.
/samples/DotNet/cs	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.
/samples/JAXWS/	Directory containing Java source code for the JAX-WS framework. 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.
/samples/Axis/	Directory containing Java source code for the Axis framework. 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.
/wsdl/srm/srm.wsdl	The Web services description language (WSDL) file containing definitions for the VMware vCenter Site Recovery Manager API.
/wsdl/srm/srm-Service.wsdl	The WSDL file defining the Web services endpoint at which the API is available. This file references `srm.wsdl` with an import statement, so you should use the appropriate generation tool with `srm–Service.wsdl` (not `srm.wsdl` directly).
/wsdl/srm/*.xsd	XML schema definition files (six).

Download and Setup Procedure

Setting up your environment to develop client applications with the SDK involves the following steps. If you are already developing vSphere applications, some of the steps are unnecessary.

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

2. Identify the target vCenter Site Recovery Manager server (or servers) to use for development.
   A "target server" is a vCenter Site Recovery Manager server that your client application will manage.

3. Install, or verify 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#. VMware recommends using Microsoft Visual Studio 2008 or later, which includes the required .NET Framework. For more information, visit the MSDN Web site.

   - 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 Web site. Open JDK works also.

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 (`wsdl.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

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

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

5. The SDK includes sample code to list and (with additions) prepare to test a recovery plan.

- For C# .NET, see "About the C# .NET Samples" on page 19.
- For JAX-WS, see "About the Java JAX-WS Samples" on page 20.
- For Java Axis, see "About the Java Axis Samples" on page 21.

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).

To build sample code with Visual Studio 2008

1 Open a Visual Studio 2008 command prompt from the Windows Start Menu as follows:

Start > **Programs** > **Visual Studio 2008** > **Visual Studio Tools** > **Command Prompt**

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.

To build sample code with Visual C# 2008 Express

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.

3 Open a Visual Studio 2008 command prompt from the Windows Start Menu as follows:

Start > **Programs** > **Visual Studio 2008** > **Visual Studio Tools** > **Command Prompt**

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.

To run the sample code from Visual Studio

1 Start Visual Studio.

2 Open the `Sample2008.sln` solution 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.

   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.

To run the C# sample code

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 the Java JAX-WS Samples

This section describes how to build and run the sample program that uses JAX-WS bindings for the SRM 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.

To build the JAX-WS sample code

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 SRM 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.

To run the JAX-WS sample code

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. Here are example settings for Linux and Windows:

```
export CLASSPATH=/usr/lib/jvm/java-7-openjdk-i386/lib
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" on page 49.

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 SRM administrator account. You need to pass in additional options:

```
--username srmadmin --password secret --planname myRecoveryPlan
```

To clean up JAX-WS sample code

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 the Java Axis Samples

This section describes how to build and run the sample program that uses Axis Web services for the SRM 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.

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

To build the Java Axis sample code

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 Table 2-2.

Table 2-2. Java and Axis environment variables

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

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.

5 If the build script produces error messages about missing classes (could not find or load class *so-and-so*), 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.

To run the Java Axis sample code

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

```
export CLASSPATH=/usr/lib/jvm/java-7-openjdk-i386/lib
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 an SRM server.

```
export VMKEYSTORE=/usr/share/mime/application/x-java-keystore.xml
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");
```

To clean up Java Axis sample code

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 Order API Usage 3

This chapter contains the following major sections:

- "Connecting to an SRM Server" on page 23
- "Protection Groups and Replication" on page 26
- "Recovery Plans and Reprotect" on page 31
- "vSphere Replication Methods" on page 35

Protection and Recovery Plans

You must use the UI, not the API, to create protection groups and set up replication, or to create recovery plans and reprotection. For details about the UI, see the Site Recovery Manager 5.8 Documentation Center.

Connecting to an SRM Server

Programs connect to an SRM server using the `SrmLoginLocale` API, or to SRM servers at both the protected site and the recovery site using the `SrmLoginSites` API.

Managed Object Reference

SRM 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 3-1. C# code to create SrmServiceInstance

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

The Java code is similar to the C# code.

Example 3-2. Java code to create SrmServiceInstance

```
private static final String SVC_INST_NAME = "SrmServiceInstance";
private static ManagedObjectReference SVC_INST_REF = new ManagedObjectReference();
    ...
    SVC_INST_REF.setType(SVC_INST_NAME);
    SVC_INST_REF.setValue(SVC_INST_NAME);
    srmService = new SrmService();
```

SrmLoginLocale

This method logs in to the SRM server. The Connect public method requires the URL of an SRM 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.

Example 3-3. C# code for SRM login

```
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;
    _service.CookieContainer = new System.Net.CookieContainer();
    _sic = _service.RetrieveContent(_svcRef);
    _service.SrmLoginLocale(_svcRef, username, password, null);
    ...
}
```

The Java code is similar to the C# code but uses a service locator.

Example 3-4. Java code for SRM login

```
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(SVC_INST_REF);
    srmPort.srmLoginLocale(SVC_INST_REF, userName, password, null);
    isConnected = true;
```

Subsequent SRM methods are called as a subclass of `_service`, for example `_service.ListPlans()` in C# or `srmport.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.

Example 3-5. C# and Java for double SRM login

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

Parameters are as follows

- `_svcRef` – managed object reference to SrmServiceInstance
- `username` – user name authorized for access to the local vCenter Server
- `password` – password for that user on the local vCenter Server
- `remoteuser` – user name authorized for access to the remote vCenter Server

- `remotepass` – password for that user on the remote vCenter Server
- `locale` – name of the locale for this session

SrmLogoutLocale

This method logs out of the SRM 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.

Example 3-6. C# code to log out

```
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 3-7. Java code to log out

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

GetApiVersion

This method gets the API version on the SRM Server that the program is logged into. Note that this is an old version 1.0 method, part of managed object SrmApi, and unlike most other methods local to SrmApi, has no direct replacement in the 5.0 or 5.8 API.

Example 3-8. Method to get API version

```
apiversion = _service.GetApiVersion(_svcRef);
```

Parameter and return value:

- `_svcRef` – managed object reference to SrmServiceInstance
- `apiversion` – a string representing the API version, such as “5.0”

Table 3-1 shows the remaining old 1.0 APIs and their replacement 5.0 APIs.

Table 3-1. Replaced APIs

SrmApi	Replacement 5.0 API
ListRecoveryPlans	SrmRecovery.ListPlans
RecoveryPlanAnswerPrompt	SrmRecoveryPlan.AnswerPrompt
RecoveryPlanSettings	SrmRecoveryPlan.RecoveryPlanGetInfo
RecoveryPlanStart, RecoveryPlanCancel	SrmRecoveryPlan.Start, SrmRecoveryPlan.Cancel
GetFinalStatus	SrmRecoveryHistory.GetRecoveryResult

Protection Groups and Replication

This section covers the SRM API methods for protection groups and virtual machine replication.

ListInventoryMappings

You establish placeholder datastores as described in the Site Recovery Manager 5.8 Documentation Center. You establish inventory mappings using the AddFolderMapping, AddNetworkMapping, and AddResourcePoolMapping methods documented in in this manual, or using procedures described in the Documentation Center.

This method returns the configured inventory mappings.

Example 3-9. Method to list inventory mappings

```
inventoryMappingInfo = _service.ListInventoryMappings(_svcRef);
```

Parameter and return value:

- `_svcRef` – managed object reference to an SrmProtection object
- `inventoryMappingInfo` – a list of inventory mappings from the protected site to the recovery site:
  - folders – a list of mapped VirtualMachine Folders
  - networks – a list of mapped virtual machine Networks and dvPortgroups
  - pools – a list of mapped Resource Pools

ListProtectionGroups

This method lists the configured protection groups.

Example 3-10. Method to list protection groups

```
SrmProtectionGroup[] = _service.ListProtectionGroups(_svcRef);
```

Parameter and return value:

- `_svcRef` – managed object reference to an SrmProtection object
- `SrmProtectionGroup[]` – an array of managed object references to all the SrmProtectionGroup managed objects that are currently configured

ListReplicatedDatastores

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

Example 3-11. Method to list replicated datastores

```
Datastore[] = _service.ListReplicatedDatastores(_svcRef);
```

Parameter and return value:

- `_svcRef` – managed object reference to an SrmProtection object
- `Datastore[]` – an array of managed object references to the replicated Datastore managed objects

You can also query protected virtual machines with "ProtectionGroupQueryVmProtection" on page 30.

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 vSphere Replication, you can use the AssociateVms method to place virtual machines into a protection group.

Before you can protect a virtual machine, the VirtualMachine's folder, resource pool, and network must be mapped from the protected site to the recovery site. To verify, you can use the ListInventoryMappings method to get a list of currently configured mappings.

Example 3-12. Method to protect virtual machines

```
SrmProtectionTaskRef = _service.ProtectVms(_svcRef, vm[] );
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionGroup object
- `vm[]` – an array of managed object references to the VirtualMachine objects for protection
- `SrmProtectionTaskRef` – the task object to monitor status of the request

UnprotectVms

This method removes virtual machines from their protection group. With 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.

Example 3-13. Method to unprotect virtual machines

```
SrmProtectionTaskRef = _service.UnprotectVms(_svcRef, vm[] );
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionGroup object
- `vm[]` – an array of managed object references to the VirtualMachine objects for unprotection
- `SrmProtectionTaskRef` – the task object to monitor status of the request

GetTasks

This method retrieves the task information from vCenter Server after a ProtectVms or UnprotectVms request, which take some time to complete.

Example 3-14. Method to get protection tasks

```
SrmProtectionTaskVmTask[] = _service.GetTasks(_svcRef);
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionTask object
- `SrmProtectionTaskVmTask[]` – Array of monitorable task information objects, each containing:
  - `task` – managed object reference to a task on the SRM server
  - `vm` – managed object reference to a VirtualMachine

IsComplete

This method checks whether the protection task has completed.

Example 3-15. Method to check protection task

```
isDone = _service.IsComplete(_svcRef);
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionTask object
- `isDone` – true if the task has completed, false if not

GetResult

This method gets detailed results of a completed protection task.

Example 3-16. Method to get protection task results

```
TaskInfoObj[] = _service.GetResult(_svcRef);
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionTask object
- `TaskInfoObj[]` – data object with at least 20 task properties set during task execution; for details see TaskInfo in the *API Reference Guide*

GetProtectionStatus

This method gets the virtual machine protection status after completion of ProtectVms or UnprotectVms.

Example 3-17. Method to get protection task results

```
protectionInfo[] = _service.GetProtectionStatus(_svcRef);
```

Parameters and return value:

- `_svcRef` – managed object reference to an SrmProtectionTask object
- `protectionInfo[]` – data object with 9 properties indicating the protection status of a virtual machine; for details see SrmProtectionGroupVmProtectionInfo in the *API Reference Guide*

GetInfo

This method retrieves basic information about the specified protection group. To get an SrmProtectionGroup managed object reference, see "ListProtectionGroups" on page 26.

Example 3-18. Method to get protection group information

```
ptGrpInfo = _service.GetInfo(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `ptGrpInfo` – information about the protection group
  - description – protection group description
  - name – protection group name
  - type – either `san` for array based replication, or `vr` for vSphere replication

GetPeer

Given an SrmProtectionGroup on the local site, this method retrieves the SrmProtectionGroup at the peer site.

Example 3-19. Method to get peer protection group

```
peerPtGrp = _service.GetPeer(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to the local SrmProtectionGroup object
- `peerPtGrp` – managed object reference to the SrmProtectionGroupPeer object

GetProtectionState

Get current state of the specified protection group. Not to be confused with "GetProtectionStatus" on page 28, which returns a virtual machine's (un)protect status, not the state of an entire protection group.

Example 3-20. Method to get protection state

```
ptState = _service.GetProtectionState(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to the local SrmProtectionGroup object
- `ptState` – enumeration for SrmProtectionGroupProtectionState:
  - failedOver – the protection group has been failed-over to the remote site
  - partiallyRecovered – the protection group is partially recovered
  - ready – the protection group is in a ready state
  - recovered – the protection group has been recovered
  - recovering – the protection group is in the process of being recovered
  - shadowing – this protection group is shadowing the remote-site group that is in ready state
  - testing – the protection group is currently being tested

ListProtectedDatastores

This method retrieves a 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.

Example 3-21. Method to list protected datastores

```
datastore[] = _service.ListProtectedDatastores(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to the local SrmProtectionGroup object
- `datastore[]` – array of managed object references to Datastore objects

ListProtectedVms

This method retrieves a list of virtual machines that are protected by the specified protection group.

Example 3-22. Method to list protected virtual machines

```
protectedVm[] = _service.ListProtectedVms(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to the local SrmProtectionGroup object

- `protectedVm[]` – array of SrmProtectionGroupProtectedVm data objects with the following fields:
  - faults – any faults associated with this protected virtual machine
  - needsConfiguration – the protected virtual machine needs to be configured or repaired
  - peerProtectedVm – the protected virtual machine identifier on the remote site
  - peerState – the protection state on the remote site
  - protectedVm – the protected virtual machine identifier on the local site
  - state – the protection state of this particular virtual machine
  - vm – the locally protected virtual machine (this reference is valid after reprotect or revert operations)

ProtectionGroupQueryVmProtection

Determine whether the specified VMs are currently protected, or can be protected. The virtual machine and its folder, resource pool, and network must be inventory mapped to the recovery site. To get a list of currently configured mappings, see "ListInventoryMappings" on page 26.

Example 3-23. Method to query virtual machine protection

```
protectInfo[] = _service.ProtectionGroupQueryVmProtection(_svcPtGrp, vms[]);
```

Parameters and return value:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `vms[]` – array of managed object references to VirtualMachine objects
- `protectInfo[]` – array of SrmProtectionGroupVmProtectionInfo data objects with the following fields:
  - faults – any faults encountered while processing queryVmProtection for this virtual machine
  - peerProtectedVm – the protected virtual machine identifier on the remote site
  - protectedVm – the protected virtual machine identifier on the local site
  - protectionGroup – the group this virtual machine is a member of, if it is protected
  - protectionGroupName – the name of this virtual machine's protection group, if it is protected
  - recoveryPlanNames – the name(s) of any recovery plans the virtual machine will be recovered in
  - recoveryPlans – any recovery plans the virtual machine will be recovered in
  - status – the current protection status of the virtual machine
  - vm – the virtual machine for which protection status is being returned

You can also query replicated datastores with "ListReplicatedDatastores" on page 26.

ProtectionGroupListRecoveryPlans

This method retrieves a list of all the recovery plans that this protection group is a member of.

Example 3-24. Method to list a protection group's recovery plans

```
plans[] = _service.ProtectionGroupListRecoveryPlans(_svcPtGrp);
```

Parameters and return value:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `plans[]` – array of managed object references to SrmRecoveryPlan objects

Recovery Plans and Reprotect

This section covers the Site Recovery Manager API methods for recovery plans and reprotection.

You can set the recovery point objective (RPO), a desired time period for rerunning replication to avoid data loss, using a slider in the vSphere Client. This is also where you can configure guest OS quiescing.

You can also set the recovery priority of virtual machines as part of running a recovery plan.

ListPlans

This method queries and retrieves all the recovery plans for the logged-into SRM server.

Example 3-25. Method to list recovery plans

```
plans = _service.ListPlans(_srm.recovery);
```

Parameters and return value:

- `_srm.recovery` – managed object reference to an SrmRecovery object
- `plans` – managed object reference to an SrmRecoveryPlan object

Once you have a list of recovery plans, you can retrieve information about each plan.

GetHistory

This method retrieves the history of a given recovery plan.

Example 3-26. Method for recovery plan history

```
history = _service.GetHistory(_srm.recovery, plan);
```

Parameters and return value:

- `_srm.recovery` – managed object reference to an SrmRecovery object
- `plan` – managed object reference to an SrmRecoveryPlan object
- `history` – managed object reference to an SrmRecoveryHistory object

RecoveryPlanGetInfo

This method retrieves status information about a given recovery plan, including the name of the recovery plan and its current state.

Example 3-27. Method to get recovery plan status

```
status = _service.RecoveryPlanGetInfo(_srm.plan);
```

Parameters and return value:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object
- `status` – managed object reference to an SrmRecoveryPlanInfo object, which includes possible states:
  - 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 the remote site, run peer recovery plan on the remote site
- ready – recovery plan is not in a running state and may be run
- running – recovery plan is currently running

The sample C# and Java code combines ListPlans with optional RecoveryPlanGetInfo for a specified plan, as shown in Example 3-28 and Example 3-29.

Example 3-28. C# sample code for recovery plan

```
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);
        }
    }
}
```

Example 3-29. Java sample code for recovery plan

```
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.

Example 3-30. Method to get recovery plan peer

```
peer = _service.RecoveryPlanGetPeer(_srm.plan);
```

Parameters and return value:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object
- `peer` – managed object reference to an SrmRecoveryPlanPeer object

Start

This method starts or reconfigures the given recovery plan, or tests and cleans it up, depending on the mode specified (see below).

Example 3-31. Method to start recovery plan

```
void _service.Start(_srm.plan, mode);
```

There is no return value. Parameters are as follows:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object
- `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 – reverse a failover, powering on virtual machines at the local site and abandoning the peer site

Cancel

This method cancels the specified recovery plan peer, in the specified mode. It can take some time to cancel a recovery plan depending on its operation state. In addition to cancel, old SRM versions had pause and resume APIs, but in SRM 5.0 and 5.8, these old APIs are no-ops.

Example 3-32. Method to cancel recovery plan

```
void _service.Cancel(_srm.plan);
```

There is no return value. Parameter is as follows:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object

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.

Example 3-33. Method to list prompts

```
prompts[] = _service.ListPrompts(_srm.plan);
```

Parameters and return value:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object
- `prompts[]` – array of data objects containing the prompt, and the key for responding to the prompt

AnswerPrompt

This method answers the current prompt step in a recovery plan, and requires test or Failover privilege depending on the mode of recovery.

Example 3-34. Method to answer prompt

```
void _service.AnswerPrompt(_srm.plan, key, cancel, response);
```

There is no return value. Parameters are as follows:

- `_srm.plan` – managed object reference to an SrmRecoveryPlan object
- `key` – string with the key value for responding to the prompt
- `cancel` – true is you want to halt further processing on this virtual machine, false otherwise
- `response` – a text string that will be recorded as the answer to this prompt

GetHistory

This method retrieves the history of a given recovery plan.

Example 3-35. Method to get recovery plan history

```
history = _service.GetHistory(_srm.recovery, plan);
```

Parameters and return value:

- `_srm.recovery` – managed object reference to an SrmRecovery object
- `plan` – managed object reference to an SrmRecoveryPlan object
- `history` – managed object reference to an SrmRecoveryHistory managed object to access recovery result

GetRecoveryResult

This method retrieves recovery results for a given recovery plan. Use this method to get the key so subsequent methods can get recovery results history.

Example 3-36. Method to get recovery plan history

```
result[] = _service.GetRecoveryResult(_srm.history, length);
```

Parameters and return value:

- `_srm.history` – managed object reference to an SrmRecoveryHistory object
- `length` – integer specifying the maximum number of results to retrieve
- `result[]`– 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
  - runMode – mode of recovery when plan was initiated (test, recovery, reprotect, revert)
  - 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

GetResultCount

This method retrieves the number of stored (XML) results for the specified recovery plan and peer plan history.

Example 3-37. Method to get recovery result history count

```
entries = _service.GetResultCount(_srm.history);
```

Parameters and return value:

- `_srm.history` – managed object reference to an SrmRecoveryHistory object
- `entries` – integer specifying the number of stored (XML) history results for this plan

GetResultLength

This method retrieves the size of the XML results for the specified recovery plan or peer plan history.

Example 3-38. Method to get size of recovery result

```
length = _service.GetResultLength(_srm.history, key);
```

Parameters and return value:

- `_srm.history` – managed object reference to an SrmRecoveryHistory object
- `key` – unique key for the plan history, from return value of the GetRecoveryResults method
- `length` – integer specifying the number of lines in the XML file

RetrieveStatus

This method gets the XML representation for the specified historical run of the referenced recovery plan. The XML document is returned 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. Only after you have retrieved all the lines and assembled them do you have a valid XML document.

Example 3-39. Method to get recovery history XML

```
*recoveryhistory = _service.RetrieveStatus(_srm.history, key, offset, lines);
```

Parameters and return value:

- `_srm.history` – managed object reference to an SrmRecoveryHistory object
- `key` – unique key for the plan history, from return value of the GetRecoveryResults method
- `offset` – integer specifying the starting line number in the XML file, starting at 0
- `lines` – integer specifying the maximum number of lines to retrieve
- `recoveryhistory` – string containing an XML representation of all recovery steps and their results

vSphere Replication Methods

You can use these methods with vSphere Replication but not with array based replication.

For array based replication, SRM organizes datastore groups to collect all files associated with protected virtual machines. You associate datastore groups with protection groups. All virtual machines in a datastore group replicate files together, and all virtual machines recover together.

You can configure 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.

ListAssociatedVms

This method lists the virtual machines currently associated with a specified vSphere Replication protection group. For the method to get a list of protection groups, see "ListProtectionGroups" on page 26.

Example 3-40. Method to list associated virtual machines

```
VirtualMachine[] = _service.ListAssociatedVms(_svcPtGrp);
```

Parameter and return value:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `VirtualMachine[]` – an array of managed object references to VirtualMachine objects

AssociateVms

This method associates one or more virtual machines with a specified vSphere Replication protection group. Before you can protect a virtual machine, it must first be associated with a protection group.

Example 3-41. Method to associate virtual machines with a protection group

```
void _service.AssociateVms(_svcPtGrp, VirtualMachine[] );
```

Parameters:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `VirtualMachine[]` – an array of managed object references to VirtualMachine object(s)

UnassociateVms

This method removes the association of one or more virtual machines with a specified vSphere Replication protection group. Once a virtual machine is unassociated, it can no longer be protected.

Example 3-42. Method to unassociate virtual machines with protection group

```
void _service.UnssociateVms(_svcPtGrp, VirtualMachine[] );
```

Parameters:

- `_svcPtGrp` – managed object reference to an SrmProtectionGroup object
- `VirtualMachine[]` – an array of managed object references to VirtualMachine object(s)

Protection Group APIs 4

This chapter contains the following major sections:

Protection Groups for Virtual Machines

In the 5.8 release of Site Recovery Manager, many new APIs were implemented 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 vCenter 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

The new APIs cannot configure host-based replication (HBR) or perform test-recover operations, however.

Resource Mappings

Resource mappings are the pairings of resources between the protected and recovery sites. That is, 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.

MoRef – Managed Object Reference

In the following sections, MoRef indicates a String that references a managed object.

Protection Group API for Virtual Machines

The new APIs that became available in SRM 5.8 are documented in alphabetic order below.

AddFolderMapping

Description

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

Example

```
void
AddFolderMapping(MoRef primaryFolder, MoRef secondaryFolder)
```

Faults

- RuntimeFault
- SrmFaultInvalidPrimaryNetworkFault
- SrmFaultInvalidSecondaryFolderFault
- SrmFaultUnknownPrimaryNetworkFault
- SrmFaultUnknownSecondaryNetworkFault

AddNetworkMapping

Description

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

Example

```
void
AddNetworkMapping(MoRef primaryNetwork, MoRef secondaryNetwork)
```

Faults

- RuntimeFault
- SrmFaultInvalidPrimaryNetworkFault
- SrmFaultInvalidSecondaryFolderFault
- SrmFaultUnknownPrimaryNetworkFault
- SrmFaultUnknownSecondaryNetworkFault

AddProtectionGroup

Description

Adds a protection group to the specified recovery plan.

Example

```
void
AddProtectionGroup(MoRef newProtectionGroup)
```

Faults

- RuntimeFault
- SrmFaultVersionConflictFault
- SrmFaultRecoveryPlanLockedFault
- SrmFaultRecoveryPlanNotFoundFault
- SrmFaultProtectionGroupNotFoundFault

- SrmFaultInvalidArgumentFault

AddResourcePoolMapping

Description

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

Example

```
void
AddResourcePoolMapping(_this, MoRef primaryResourcePool, MoRef secondaryResourcePool)
```

`_this` is a managed object reference to an `InventoryService` object.

Faults

- RuntimeFault
- SrmFaultUnknownPrimaryResourcePoolFault
- SrmFaultUnknownSecondaryResourcePoolFault

CreateAbrProtectionGroup

Description

Create an array-based replication (ABR) protection group.

Example

```
MoRef ProtectionGroup
CreateAbrProtectionGroup(_this, MoRef protectionFolder, String nameAbrProtectionGroup,
               @optional String descriptionAbrProtectionGroup, MoRef datastoreRef)
```

`_this` is a managed object reference to an `SrmProtection` object.

Faults

- InvalidTypeFault
- RuntimeFault
- SrmFaultInvalidArgumentFault
- SrmFaultReplicationProviderFaultFault

CreateHbrProtectionGroup

Description

Create a host-based replication (HBR) protection group.

Example

```
MoRef ProtectionGroup
CreateHbrProtectionGroup(_this, MoRef protectionFolder, String nameHbrProtectionGroup,
               @optional String descriptionHbrProtectionGroup, MoRef datastoreRef)
```

`_this` is a managed object reference to an `SrmProtection` object.

Faults

- InvalidTypeFault
- RuntimeFault
- SrmFaultReplicationProviderFaultFault

GetCreateProtectionGroupResult

Description

A task returned by GetNewProtectionGroup fills in the protection group result with the final status of the operation. This function gets the TaskInfo object containing the detailed results. To check completeness, see IsCreateProtectionGroupComplete.

Example

```
TaskInfo taskResult
GetCreateProtectionGroupResult(MoRef protectionGroup)
```

Faults

- RuntimeFault

GetName

Description

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

Example

```
String entityName
GetName(MoRef _folderRef)
```

Faults

- RuntimeFault

GetNewProtectionGroup

Description

After calling CreateAbrProtectionGroup or CreateHbrProtectionGroup 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.

Example

```
MoRef protectionGroup
GetNewProtectionGroup(_this)
```

_this is a managed object reference to an CreateProtectionGroupTask object.

Faults

- RuntimeFault
- SrmFaultInvalidStateFault

GetPairedSite

Description

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.

Example

```
String remoteSite
GetPairedSite(_this)
```

_this is a managed object reference such as _svcRef.

Faults

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

GetParentFolder

Description

The API interacts with `ProtectionGroupFolder` or `RecoveryPlanFolder` instances; the former holds only protection group and child folders, the latter holds only recovery plan and child folders. This function gets the parent folder for either of the aforementioned folder types.

Example

```
MoRef parentFolder
GetParentFolder(MoRef thisFolder)
```

Faults

- RuntimeFault

GetProtectionGroupRootFolder

Description

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

Example

```
MoRef ProtectionGroupFolder
getProtectionGroupRootFolder(MoRef SrmProtection)
```

Faults

- RuntimeFault

GetProtectionGroup

Description

Returns a reference to the protection group with the specified name, if any.

Example

```
MoRef protectionGroup
GetProtectionGroup(MoRef _pgFolder, String protectionGroupName)
```

Faults

- RuntimeFault

GetRecoveryPlanRootFolder

Description

Gets a reference to the top-level container (the root folder) for recovery plans.

Example

```
MoRef RecoveryPlanFolder
GetRecoveryPlanRootFolder(MoRef SrmReovery)
```

Faults

- RuntimeFault

GetRecoveryPlan

Description

Gets a managed object reference to the recovery plan with the specified name in this Recovery Plan Folder.

Example

```
MoRef RecoveryPlan
GetRecoveryPlans(MoRef RecoveryPlanFolder, String planName);
```

Faults

- RuntimeFault
- SrmRecoveryPlanNotFoundFault

GetRecoverySettings

Description

Gets the recovery settings for the specified virtual machine.

Example

```
RecoverySettings recoverySettings
GetRecoverySettings(MoRef SrmRecoveryPlan, MoRef virtualMachine)
```

Data Type

The `RecoverySettings` class represents VM recovery settings for presentation in the user interface. The external object is a flat structure with these fields:

```
– changeVersion
– status
– recoveryPriority
– skipGuestShutdown
– powerOffTimeoutSeconds
– finalPowerState
– localFaultToleranceState
– remoteFaultToleranceState
– powerOnTimeoutSeconds
– powerOnDelaySeconds
– prePowerOnCallouts
– postPowerOnCallouts
```

Faults

- RuntimeFault
- SrmFaultRecoveryPlanNotFoundFault
- SrmFaultVmNotFoundInRecoveryPlanFault

GetSiteName

Description

Get the name of the current site.

Example

```
_service.GetSiteName(_siteRef)
```

`_siteRef` is a managed object reference to the `SrmServiceInstance` object.

Faults

- RuntimeFault

IsCreateProtectionGroupComplete

Description

A task returned by GetNewProtectionGroup fills in the protection group result with the final status. This function checks completeness of the operation. To get the result, see GetCreateProtectionGroupResult.

Example

```
Boolean isComplet
IsCreateProtectionGroupComplete(MoRef CreateProtectionGroupTask);
```

Faults

- RuntimeFault

ListChildProtectionGroupFolders

Description

Return the child Protection Group Folders located in this folder.

Example

```
MoRef ProtectionGroupFolder[]
ListChildProtectionGroupFolders(MoRef ProtectionGroupFolder)
```

Faults

- RuntimeFault

ListChildProtectionGroups

Description

Return the `SrmProtectionGroup` objects located in this folder.

Example

```
MoRef SrmProtectionGroup[]
ListChildProtectionGroups(MoRef ProtectionGroupFolder)
```

Faults

- RuntimeFault

ListChildRecoveryPlanFolders

Description

Return the child Recovery Plan Folders located in this folder.

Example

```
MoRef RecoveryPlanFolder[]
ListChildRecoveryPlanFolders(MoRef RecoveryPlanFolder)
```

Faults

- RuntimeFault

ListChildRecoveryPlans

Description

Returns an array of `SrmRecoveryPlan` objects located in this folder.

Example

```
MoRef SrmRecoveryPlan[]
ListChildRecoveryPlans(MoRef RecoveryPlanFolder)
```

Faults

- RuntimeFault

ListUnassignedReplicatedDatastores

Description

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

Example

```
MoRef Datastore[]
ListUnassignedReplicatedDatastores(MoRef SrmProtection)
```

Faults

- RuntimeFault

ListUnassignedReplicatedVms

Description

Get a list of replicated VMs which are currently not assigned to an SRM protection group.

Example

```
MoRef Vm[]
ListUnassignedReplicatedVms(MoRef SrmProtection, String replicationType)
```

String `replicationType` is an enumeration defined in `SrmProtectionGroup`. Valid values are `san` or `vr`. The former is for ABR replicated VMs, while the latter is for HBR replicated VMs.

Faults

- RuntimeFault
- SrmFaultInvalidArgumentFault

ProtectionGroupGetParentFolder

Description

Given a protection group, get the parent folder.

Example

```
MoRef protectionGroupFolder
getParentFolder(MoRef protectionGroup)
```

Faults

- RuntimeFault

ProtectionListProtectedDatastores

Description

Get a list of the replicated datastores that are protected by SRM.

Example

```
MoRef datastore[]
ProtectionListProtectedDatastores(MoRef protectionGroup)
```

Faults

- RuntimeFault

ProtectionListProtectedVms

Description

Get a list of virtual machines that are protected by SRM.

Example

```
MoRef virtualMachine[]
ProtectionListProtectedDatastores(MoRef protectionGroup)
```

Faults

- RuntimeFault

RecoveryPlanGetParentFolder

Description

Get the parent folder (root) for a recovery plan.

Example

```
MoRef recoveryPlanFolder
RecoveryPlanGetParentFolder(MoRef recoveryPlan)
```

Faults

- RuntimeFault

SetRecoverySettings

Description

Set the recovery settings for this recovery plan.

Example

```
void
SetRecoverySettings(MoRef SrmRecoveryPlan, String vmId, RecoverySettings settings)
```

Faults

- RuntimeFault
- SrmFaultDependencyConflictFault
- SrmFaultInvalidArgumentFault
- SrmFaultRecoveryPlanLockedFault
- SrmFaultRecoveryPlanNotFoundFault
- SrmFaultVersionConflictFault
- SrmFaultVmNotFoundInRecoveryPlanFault

SrmLoginRemoteSite

Description

Log in to the remote SRM 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.

Example

```
void
SrmLoginRemoteSite(_this, String remoteUsername, String remotePassword, @optional String locale)
```

`_this` is a managed object reference to `ServiceInstance`.

Faults

- InvalidLocaleFault
- InvalidLoginFault
- RuntimeFault
- SrmFaultAlreadyLoggedInFault
- SrmFaultConnectionLimitReachedFault
- SrmFaultRemoteSiteNotEnabledFault

SRM Faults

Table 4-1 shows various faults thrown by SRM external APIs.

Table 4-1. Faults thrown by SRM functions

Fault	Description
AlreadyLoggedInFault	The session is already logged in, and Login was called again
ConnectionDownFault	The VMOMI connection to the remote server is down
ConnectionLimitReached	Thrown when the configured connection limit has been reached
DependencyConflict	UpdateVmSettings operation was attempted that might cause a dependency cycle
DuplicateName	Call is unable to determine which object to use due to name conflict
InsufficientLicensesFault	Thrown by a method that cannot acquire licenses for the object to create
InternalError	An internal error occurred that is not better described by a more specific fault
InvalidArgument	Base class for invalid argument exceptions
InvalidLogin	Cannot complete login due to an incorrect user name or password
InvalidPrimaryFolder	Thrown for an attempt to create a primary site folder that cannot contain VMs
InvalidPrimaryNetwork	Invalid primary network specified for mapping (such as uplink DVPortgroup)
InvalidSecondaryFolder	Thrown for an attempt to create a secondary site folder that cannot contain VMs
InvalidSecondaryNetwork	Invalid secondary network specified for mapping (such as uplink DVPortgroup)
InvalidState	Base class for invalid state exceptions
NoPermission	Operation denied because of a privilege not held on a managed object
NotAuthenticated	Operation denied because the session has not successfully logged in
PromptNotFound	Thrown when a RecoveryPrompt could not be found
ProtectionGroupNotEmpty	Thrown after attempt to remove a protection group that contains protected VMs
ProtectionGroupNotFound	Thrown when an operation on protection group could not find the protection group
RecoveryPlanLocked	An attempt was made to change a RecoveryPlan that is locked
RecoveryPlanNotFound	Thrown when the requested recovery plan was not found
RecoveryResultNotFound	Thrown when a RecoveryResult could not be found
RemoteSiteNotEnabled	An attempt was made to use a remote site that has not been enabled
ReplicationProviderFault	Thrown when an unspecified error was returned from the replication provider
StringArgumentTooLong	Thrown when a string argument exceeds {maxSize} characters
UnknownPrimaryFolder	Secondary site tried operation on a folder that is nonexistent on primary site
UnknownPrimaryNetwork	Secondary site tried operation on a network that is nonexistent on primary site
UnknownPrimaryResourcePool	Secondary site tried operation on resource pool that is nonexistent on primary site
UnknownSecondaryFolder	Primary site tried operation on a folder that is nonexistent on secondary site
UnknownSecondaryNetwork	Primary site tried operation on a network that is nonexistent on secondary site
UnknownSecondaryResourcePool	Primary site tried operation on resource pool that is nonexistent on secondary site
VersionConflict	Attempt to reconfigure with a changeVersion that does not match the current value
VmNotFoundInRecoveryPlan	Attempt to retrieve settings for virtual machine that does not exist in RecoveryPlan

SSL Certificates and SNMP Traps

A

This appendix contains the following main sections:

- "SSL Certificates" on page 49
- "SNMP Traps" on page 51

SSL Certificates

The SRM Web service listens by default on port 9007. It 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, for example:

```
https://<FQDN.hostname.or.IP.Address>:9007
```

Server Certificate Requirements

The SRM API is a secure Web service running on the SRM Server. To develop client applications, you must obtain the vCenter Server certificate, which is used by the SRM Server, and import it into the certificate store of the workstation where you develop client applications.

To obtain a vCenter Server certificate:

1. From your development workstation, open Internet Explorer (IE).
2. Navigate to the vCenter Server using HTTPS protocol – `https://<servername>`.

   A Security Alert message displays a warning regarding the certificate's certifying authority.
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.

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.

Exporting Cached Certificates to a Local Directory

For Java development in a Windows environment, you must export the certificate to a local directory:

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.

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

SRM 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 SRM API interface should listen for the SNMP traps listed below. 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 actually two ways to generate SNMP traps from SRM. The first is the method presented here and in other SRM documentation. The second method to generate traps is by configuring SNMP actions on the events and alarms that SRM 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.

The SNMP traps listed in Table A-1 originate from SRM, not from vCenter Server.

MIB Names for SNMP Traps

Table A-1 provides a description of SNMP traps 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 A-1. SNMP Traps in the MIB

SNMP Trap	What Trap Indicates
RecoveryPlanExecuteTestBegin	Signaled on the recovery site when a recovery test is initiated.
RecoveryPlanExecuteTestEnd	Signaled on the recovery site when a recovery test has completed. If an error occurred it is available as [data.Error]
RecoveryPlanExecuteBegin	Signaled on the recovery site when a recovery is initiated.
RecoveryPlanExecuteEnd	Signaled on the recovery site when a recovery has completed. If an error occurred it is available as [data.Error]
RecoveryVmBegin	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.
RecoveryVmEnd	Signaled after the last post-power on script has completed, or after a recovery-stopping error has occurred for the virtual machine.
RecoveryPlanPromptDisplay	The recovery plan is displaying prompt [data.PromptKey] and is waiting for user input. PromptKey is a unique identifier.
RecoveryPlanPromptResponse	The recovery plan received an answer to prompt [data.PromptKey] and is no longer paused waiting for user input.
RecoveryPlanServerCommandBegin	Signaled on the recovery site when SRM starts to run a Callout command on the SRM server.
RecoveryPlanServerCommandEnd	Signaled on the recovery site when SRM has finished running a Callout command on the SRM server.
RecoveryPlanVmCommandBegin	Signaled on the recovery site when SRM has started to run a Callout command on a recovered virtual machine.
RecoveryPlanVmCommandEnd	Signaled on the recovery site when SRM has finished running a Callout command on a recovered virtual machine.
RecoveryPlanExecuteReprotectBegin	Signaled on the recovery site when a reprotect is initiated.
RecoveryPlanExecuteReprotectEnd	Signaled on the recovery site when a reprotect has completed. If an error occurred it is available as [data.Error]
RecoveryPlanExecuteCleanupBegin	Signaled on the recovery site when a test cleanup is initiated.
RecoveryPlanExecuteCleanupEnd	Signaled on the recovery site when a test cleanup has completed. If an error occurred it is available as [data.Error]

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 SMNP traps themselves with IDs.

```
MIB_OBJECT                   ID
oidDrVmName                  1.3.6.1.4.1.6876.51.1.1
oidDrRecoveryName            1.3.6.1.4.1.6876.51.1.2
oidDrPromptString            1.3.6.1.4.1.6876.51.1.3
oidDrRecoveryType            1.3.6.1.4.1.6876.51.1.4
oidDrRecoveryState           1.3.6.1.4.1.6876.51.1.5
oidDrSiteString              1.3.6.1.4.1.6876.51.1.6
oidDrVmUuid                  1.3.6.1.4.1.6876.51.1.7
oidDrResult                  1.3.6.1.4.1.6876.51.1.8
oidDrCommandName             1.3.6.1.4.1.6876.51.1.9     <-- New MIB object
MIB_TRAP                              ID
RecoveryPlanExecuteTestBegin          1.3.6.1.4.1.6876.51.0.1
RecoveryPlanExecuteTestEnd            1.3.6.1.4.1.6876.51.0.2
RecoveryPlanExecuteBegin              1.3.6.1.4.1.6876.51.0.3
RecoveryPlanExecuteEnd                1.3.6.1.4.1.6876.51.0.4
RecoveryVmBegin                       1.3.6.1.4.1.6876.51.0.5
RecoveryVmEnd                         1.3.6.1.4.1.6876.51.0.6
RecoveryPlanPromptDisplay             1.3.6.1.4.1.6876.51.0.7
RecoveryPlanPromptResponse            1.3.6.1.4.1.6876.51.0.8
RecoveryPlanServerCommandBegin        1.3.6.1.4.1.6876.51.0.9      <-- New MIB trap
RecoveryPlanServerCommandEnd          1.3.6.1.4.1.6876.51.0.10     <-- New MIB trap
RecoveryPlanVmCommandBegin            1.3.6.1.4.1.6876.51.0.11     <-- New MIB trap
RecoveryPlanVmCommandEnd              1.3.6.1.4.1.6876.51.0.12     <-- New MIB trap
RecoveryPlanExecuteReprotectBegin     1.3.6.1.4.1.6876.51.0.13     <-- New MIB trap
RecoveryPlanExecuteReprotectEnd       1.3.6.1.4.1.6876.51.0.14     <-- New MIB trap
RecoveryPlanExecuteCleanupBegin       1.3.6.1.4.1.6876.51.0.15     <-- New MIB trap
RecoveryPlanExecuteCleanupEnd         1.3.6.1.4.1.6876.51.0.16     <-- New MIB trap
```

Index