

CA ARCserve® Replication 和 High Availability

SOAP Line 参考指南

r16.5



本文档仅供参考，其中包括内嵌帮助系统和以电子形式分发的材料（以下简称“文档”），Arcserve 随时可对其进行更改或撤销。

未经 Arcserve 事先书面同意，不得擅自复制、转让、翻印、透露、修改或转录本文档的全部或部分内容。本文档属于 Arcserve 机密和专有信息，不得擅自透露，或除以下协议中所允许的用途，不得用于其他任何用途：(i) 您与 Arcserve 之间关于使用与本文档相关的 Arcserve 软件的单独协议；或者 (ii) 您与 Arcserve 之间单独的保密协议。

尽管有上述规定，但如果您为本文档中所指的软件产品的授权用户，则您可打印或提供合理数量的本文档副本，供您及您的雇员内部用于与该软件相关的用途，前提是所有 Arcserve 版权声明和标识必须附在每一份副本上。

打印或提供本文档副本的权利仅限于此类软件所适用的许可协议的有效期内。如果该许可因任何原因而终止，您应负责向 Arcserve 书面证明已将本文档的所有副本和部分副本已退还给 Arcserve 或被销毁。

在所适用的法律允许的范围内，ARCSERVE 按照“现状”提供本文档，不附带任何保证，包括但不限于商品适销性、适用于特定目的或不侵权的默示保证。在任何情况下，ARCSERVE 对您或其他第三方由于使用本文档所造成的直接或间接损失或损害都不负任何责任，包括但不限于利润损失、投资损失、业务中断、信誉损失或数据丢失，即使 ARCSERVE 已经被提前明确告知这种损失或损害的可能性。

本文档中涉及的任何软件产品的使用均应遵照有关许可协议的规定且根据本声明中的条款不得以任何方式修改此许可协议。

本文档由 Arcserve 制作。

仅提供“有限权利”。美国政府使用、复制或透露本系统受 FAR Sections 12.212、52.227-14 和 52.227-19(c)(1) - (2) 以及 DFARS Section 252.227-7014(b)(3) 的相关条款或其后续条款的限制。

版权所有 © 2015 Arcserve (USA), LLC 及其子公司和分支机构。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

Arcserve 产品引用

本文档引用以下 Arcserve 产品：

- Arcserve® Replication
- Arcserve® High Availability (HA)
- Arcserve® Assured Recovery®
- Arcserve® Content Distribution

联系 Arcserve

Arcserve 支持团队提供了丰富的资源集，用于解决您的技术性问题，并允许轻松访问重要的产品信息。

<https://www.arcserve.com/support>

借助 Arcserve 支持：

- 您可以与由 Arcserve 支持专家内部共享的相同信息库直接接触。此站点为您提供我们知识库 (KB) 文档的访问权限。从这里您可以轻松搜索并找到产品相关的 KB 文章，包含针对许多顶层问题和共同问题的实地解决方案。
- 您可以使用我们的 Live Chat 链接，立即启动与 Arcserve 支持团队之间的实时对话。使用 Live Chat，您可以获得您所关注问题的答复，同时仍可访问该产品。
- 您可以参加 Arcserve 全球用户社区以便提问和回答问题、共享建议和技巧、讨论最佳实践并与同行对话。
- 您可以开出支持故障单。通过在线开出支持故障单，您可以从您正在咨询的产品领域的专家那里得到回复。

您可以访问适于您 Arcserve 产品的其他有用资源。

[就产品文档提供反馈](#)

如果您对 Arcserve 产品文档有任何意见或疑问。请联系[我们](#)。

文档更改

自此文档的上一版本以来已做出以下文档更新：

- 这是此参考指南的第一版。

目录

第 1 章： 简介	7
使用 SOAP API 编程 - 流	7
SOAP API 的概述	7
本指南中使用的约定	7
第 2 章： SOAP API	9
方案管理 API	9
开始工作: <code>create_session()</code>	9
创建方案: <code>create_scenario_ex()</code>	11
设置根目录: <code>set_root_dir()</code>	13
添加根目录: <code>add_root_dir()</code>	14
添加副本主机: <code>add_replica()</code>	15
添加多个副本主机: <code>add_replica_ex()</code>	16
设置方案属性: <code>set_scenario_data()</code>	17
设置主机属性: <code>set_host_data()</code>	18
启动方案: <code>run()</code>	20
停止方案: <code>stop()</code>	21
暂停数据复制: <code>suspend_replication()</code>	22
完成工作: <code>close_session()</code>	23
管理凭据: <code>add_credentials_ex()</code>	24
与 Arcserve Backup 集成: <code>add_bab_credentials()</code>	25
更新方案属性: <code>update_scenario()</code>	27
删除方案: <code>remove_scenario()</code>	28
导入方案: <code>import_scenario()</code>	29
同步运行方案: <code>synchronize()</code>	30
恢复复制: <code>resume_replication()</code>	31
添加回滚书签: <code>set_rewind_bookmark()</code>	32
高可用性方案管理 API	33
触发保障恢复: <code>start_ar()</code>	33
禁用高可用性方案中的心跳监视: <code>stop_is_alive()</code>	35
切换高可用性方案: <code>switchover()</code>	36
启用高可用性方案中的心跳监视: <code>start_is_alive()</code>	39
VSS 快照管理 API	40
在副本服务器上安装 VSS 快照: <code>mount_snapshot()</code>	40
从副本服务器卸载 VSS 快照: <code>unmount_snapshot()</code>	42
从副本服务器中删除 VSS 快照: <code>delete_snapshot()</code>	43
从副本服务器获取 VSS 快照列表: <code>get_snapshot_list()</code>	44

收集统计 API	45
获取扩展方案统计信息: <code>get_data_ex()</code>	45
获取方案统计信息: <code>get_scenario_data()</code>	52

第 1 章：简介

此部分包含以下主题：

[使用 SOAP API 编程 - 流 \(p. 7\)](#)

[SOAP API 的概述 \(p. 7\)](#)

[本指南中使用的约定 \(p. 7\)](#)

使用 SOAP API 编程 - 流

要使用 SOAP API 控制 Arcserve Replication and High Availability，客户端应针对 RHA CS 验证，然后创建会话。此操作通过调用 `create_session` API 函数完成。客户端创建会话之后，客户端可以调用在此参考指南中描述的任何 API 函数。完成该任务时，客户端调用 `close_session` API 以便作废该会话并释放资源。

SOAP API 的概述

此指南描述 Arcserve Replication and High Availability 导出的 SOAP API。通过 API，您可以管理各个复制和高可用性。例如，您可以控制 RHA 方案，并可以执行各种操作，如运行或停止方案并执行方案故障切换操作。要使用 Arcserve Replication and High Availability SOAP API，您可以使用包含集成 Web 服务支持的任何编程语言，如 Visual Basic 或 C#。

注意：在此参考指南中提供的示例代码使用 C# 语言编写。

用于示例代码的 SOAP API URL 是

http://127.0.0.1:8088/ws_man/xosoapapi.asmx。将 URL 的地址部分替换为 RHA CS IP 地址或调用 API 时的主机名。

本指南中使用的约定

《Arcserve Replication and High Availability SOAP API 参考指南》使用下表中所述的参数类型：

类型	说明
<code>bool</code>	布尔值
<code>out uint</code>	输出整数

[本指南中使用的约定](#)

类型	说明
<code>out ulong</code>	输出无正负号的长整数
<code>out string</code>	输出字符串
<code>uint</code>	无正负号的整数
<code>ulong</code>	无正负号的长整数
<code>ushort</code>	无正负号的短整数

第 2 章： SOAP API

此部分包含以下主题：

- [方案管理 API \(p. 9\)](#)
- [高可用性方案管理 API \(p. 33\)](#)
- [VSS 快照管理 API \(p. 40\)](#)
- [收集统计 API \(p. 45\)](#)

方案管理 API

以下部分描述执行基本方案管理任务的 API。

本节包括以下主题：

- [开始工作: `create_session\(\)` \(p. 9\)](#)
- [创建方案: `create_scenario_ex\(\)` \(p. 11\)](#)
- [设置根目录: `set_root_dir\(\)` \(p. 13\)](#)
- [添加根目录: `add_root_dir\(\)` \(p. 14\)](#)
- [添加副本主机: `add_replica\(\)` \(p. 15\)](#)
- [添加多个副本主机: `add_replica_ex\(\)` \(p. 16\)](#)
- [设置方案属性: `set_scenario_data\(\)` \(p. 17\)](#)
- [设置主机属性: `set_host_data\(\)` \(p. 18\)](#)
- [启动方案: `run\(\)` \(p. 20\)](#)
- [停止方案: `stop\(\)` \(p. 21\)](#)
- [暂停数据复制: `suspend_replication\(\)` \(p. 22\)](#)
- [完成工作: `close_session\(\)` \(p. 23\)](#)
- [管理凭据: `add_credentials_ex\(\)` \(p. 24\)](#)
- [与 Arcserve Backup 集成: `add_bab_credentials\(\)` \(p. 25\)](#)
- [更新方案属性: `update_scenario\(\)` \(p. 27\)](#)
- [删除方案: `remove_scenario\(\)` \(p. 28\)](#)
- [导入方案: `import_scenario\(\)` \(p. 29\)](#)
- [同步运行方案: `synchronize\(\)` \(p. 30\)](#)
- [恢复复制: `resume_replication\(\)` \(p. 31\)](#)
- [添加回滚书签: `set_rewind_bookmark\(\)` \(p. 32\)](#)

开始工作: `create_session()`

`create_session` API 函数允许您根据 Control Service 进行身份验证并返回会话 ID。在调用任何其他 API 函数时，通过会话 ID 作为参数。通过使用 [`close_session API` \(p. 23\)](#) 使打开的会话无效。

参数

create_session API 函数包括下表所述参数：

名称	类型	说明
user_name	string	用户名。 示例： MyDomain\Administrator
password	string	用户名的密码。 示例： Arcserve
error_code	out uint	零值表示 API 函数成功执行。非零值表示失败 API。

返回值

使用 create_session 的身份验证成功时，此函数使用会话 ID 且将 error_code 参数设置为 0 的情况下返回 uint 值。否则，将返回最大 uint 值（0xFFFFFFFF），且 error_code 参数包含详细的错误代码。

注意：您应当以格式 <DOMAIN_NAME>\<USER_NAME> 来指定 user_name 参数。例如，test_domain\Administrator。

示例

示例 1：创建 Web 服务对象。

```
xosoapapi_c get_mng()
{
    xosoapapi_c mng = new xosoapapi_c();
    return mng;
}
```

示例 2: 创建会话。

```
public bool create_session_example()
{
    try
    {
        uint err = 0;
        string user_name =
global::api_examples.Properties.Settings.Default.user_name;
        string password =
global::api_examples.Properties.Settings.Default.password;
        _session_id = get_mng().create_session(user_name, password, out
err);
        if (_session_id == 0xffffffff)
        {
            return false;
        }
        return true;
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

创建方案: [create_scenario_ex\(\)](#)

`create_scenario_ex` API 函数允许您创建方案。您创建方案后，您可以执行诸如运行方案、更改方案属性、添加或删除主机等操作。

参数

`create_scenario_ex` API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
product	uint	指定对应于以下产品的整数值： 0 - DR 方案 1 - HA 方案 2 - CD 方案

名称	类型	说明
application	uint	<p>指定对应于以下应用程序的整数值:</p> <ul style="list-style-type: none"> 0 - 文件服务器方案。 1 - Exchange 方案 2 - Oracle 方案 3 - 未使用 4 - SQL Server 方案。 5 - IIS 方案 6 - Control Service 方案 7 - Hyper-v 方案 8 - Sharepoint 方案 9 - vCenter 方案 10 - CRM 方案 11 - 完整系统方案 12 - 自定义方案
is_ar	bool	<p>指定方案是否支持 Assured Recovery (AR):</p> <ul style="list-style-type: none"> ■ true: 支持 AR ■ false: 不支持 AR
is_cdp	bool	始终 false, 未使用。
integrate_opt	uint	<p>指定对应于以下应用程序的整数值:</p> <ul style="list-style-type: none"> 0 - 无集成 1 - Backup 2 - Arcserve D2D 3 - Arcserve Central Applications
group_id	uint	组 ID。group_id 是 0xFFFFFFFF 时, 方案属于默认方案组, 其通常被命名为“方案”。

名称	类型	说明
scenario_data	out string	方案的数据。调用此 API 之后，参数包含更新的方案数据。 注意： 有关详细信息，请参阅本主题中的“示例”。
group_data	out string	组数据。

返回值

此 API 返回创建的方案 ID。其他 API（如运行和停止）需要方案 ID。方案 ID 为零时，API 无效。另外，非零值表示，API 成功完成。

注意：在此 API 调用成功完成时，方案的框架即被创建。方案的所有属性都有默认值。重要属性（如根目录、主机 IP 等等）是空的。您可以调用其他 API 来满足这些属性。此参考指南中后面的 API 描述如何满足这些属性。

示例

```
uint product = 0;
        uint app = 0;
        bool is_ass_rec = false;
        bool is_cdp = false;
        uint si_opt = 0;
        uint group_id = 0xFFFFFFFF;
        string scenario_data = "";
        string group_data = "";

        uint scenario_id = get_mng().create_scenario_ex(_session_id,
product, app, is_ass_rec, is_cdp, si_opt, group_id, out scenario_data, out
group_data);
```

设置根目录： `set_root_dir()`

`set_root_dir()` API 函数允许您编辑根目录的路径。

参数

`set_root_dir` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	方案的 ID。
<code>host_index</code>	<code>uint</code>	想要设置的主机的索引。
<code>root_dir_index</code>	<code>uint</code>	根目录的索引，从零开始，如果您有两个根目录，索引为 0 和 1。
<code>root_dir</code>	<code>string</code>	文件夹名称。 注意： 仅完整系统方案需要此参数。然而，此 API 函数当前不支持为完整系统方案设置根目录。

返回值

API 调用成功完成时，此函数返回 `true` 值。否则，此函数返回 `false` 值。

注意：[create_scenario_ex](#) (p. 11) API 仅创建方案的框架。您可以调用此 API 以设置根目录。

示例

```
uint master_host_index = 1;

get_mng().set_root_dir(_session_id, scenario_id, master_host_index, 0,
"E:/test");
```

添加根目录： `add_root_dir()`

`add_root_dir` API 函数允许您为方案添加新的根目录。

参数

`add_root_dir` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
<code>scenario_data</code>	<code>ref string</code>	方案的数据。调用此 API 之后，参数包含更新的方案数据。
<code>root_directories</code>	<code>string</code>	文件夹名称。
<code>new_root_dir_index</code>	<code>out uint</code>	新创建的根目录的索引。

返回值

API 调用成功完成时，此函数返回 `true` 值。否则，此函数返回 `false` 值。

示例：添加根目录

```
String scenario_data = get_mng().get_scenario_data(_session_id, scenario_id);

//添加根目录
get_mng().add_root_dir(_session_id, ref scenario_data, "c:/test", out
new_root_dir_id);
```

添加副本主机： `add_replica()`

`add_replica` API 函数允许您为方案添加新的副本主机。如果想将主主机数据复制到多个副本主机，您可以调用此 API 来添加一些副本主机。

参数

`add_replica` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。

名称	类型	说明
scenario_data	ref string	方案的数据。调用此 API 之后，参数包含更新的方案数据。
host_index	uint	父主机索引；主主机索引始终为 1；第一个副本主机通常为 2。
new_replica_index	out uint	新添加的副本主机的索引

返回值

API 调用成功完成时，此函数返回 `true` 值。否则，此函数返回 `false` 值。

示例：添加副本主机

```
//在主主机下添加一个副本主机
get_mng().add_replica(_session_id, ref scenario_data, master_host_index, out
new_replica_index);
```

添加多个副本主机： `add_replica_ex()`

`add_replica_ex` API 函数允许您为方案一次添加一个或多个新的副本主机。

参数

`add_replica_ex` API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
scenario_data	ref string	方案的数据。调用此 API 之后，参数包含更新的方案数据。
host_index	uint	父主机索引；主主机索引始终为 1；第一个副本主机通常为 2。

名称	类型	说明
host_list	ref string	<p>主机列表以 XML 格式提供主机方面的信息。xml 包含您想要添加到方案的主机的有关信息。</p> <p>注意：请参阅此表之后的屏幕。</p>

例如，下列的 xml 代码包含有关两个主机的信息。

```
<?xml version="1.0"?>
<object>
- <object>
  <data type="String" val="155.35.76.155" label="Host"/>
  <data type="String" val="155.35.76.155" label="IP"/>
</object>
- <object>
  <data type="String" val="155.35.76.156" label="Host"/>
  <data type="String" val="155.35.76.156" label="IP"/>
</object>
</object>
```

返回值

API 调用成功完成时，此函数返回 true 值。否则，此函数返回 false 值。

示例：添加多个副本主机

```
string host_list = "<?xml version=\"1.0\"?>登录信息 登录信息<data label=\"Host\" val=\"155.35.76.155\" type=\"String\" /><data label=\"IP\" val=\"155.35.76.155\" type=\"String\" /> </object>登录信息<data label=\"Host\" val=\"155.35.76.156\" type=\"String\" /><data label=\"IP\" val=\"155.35.76.156\" type=\"String\" /> </object></object>";

//在主主机下添加两个副本主机
get_mng().add_replica_ex(_session_id, ref scenario_data, master_host_index, ref host_list);
```

设置方案属性：[set_scenario_data\(\)](#)

[set_scenario_data](#) API 功能允许您编辑方案属性。方案的多数属性可以由此 API 更新。

参数

`set_scenairo_data` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	方案的 ID。
<code>property_name</code>	<code>string</code>	主机的属性名称。每个属性有唯一的名称。
<code>property_value</code>	<code>string</code>	属性的值。

返回值

API 调用成功完成时，此函数返回 `true` 值。否则，此函数返回 `false` 值。

示例：设置方案属性

```
//设置方案数据属性
get_mng().set_scenario_data(_session_id, scenario_id,
"Scenario.ReplicateCompressAttr", "True");
```

下表列出常用方案属性名称。

属性名称	说明
<code>Scenario.ScenarioName</code>	方案名称
<code>Scenario.BuildShares</code>	同步 windows 共享
<code>Scenario.SyncADS</code>	复制 NTFS ADS

设置主机属性：[set_host_data\(\)](#)

`set_host_data()` API 函数允许您编辑主机属性。主机的多数属性可以由此 API 更新。

参数

`set_host_data` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
<code>host_index</code>	<code>uint</code>	主机的索引。
<code>scenario_id</code>	<code>uint</code>	方案的 ID。
<code>property_name</code>	<code>string</code>	主机的属性名称。每个属性有唯一的名称。
<code>property_value</code>	<code>string</code>	属性的值。

返回值

API 调用成功完成时，此函数返回 `true` 值。否则，此函数返回 `false` 值。

示例：设置主机属性

```
//设置主机的主机名和 IP 地址
get_mng().set_host_data(_session_id, scenario_id, master_host_index,
"Scenario.ReplicationTree.ReplNode.CommonHostProps.Host", "master");

get_mng().set_host_data(_session_id, scenario_id, master_host_index,
"Scenario.ReplicationTree.ReplNode.CommonHostProps.IP", "155.35.78.187");
```

下表列出常用主主机属性名称。

属性名称	说明
<code>Scenario.ReplicationTree.ReplNode.CommonHostProps.Host</code>	主主机名称
<code>Scenario.ReplicationTree.ReplNode.CommonHostProps.IP</code>	主主机 IP
<code>Scenario.ReplicationTree.ReplNode.CommonHostProps.Data_IP</code>	主主机复制 IP 地址
<code>Scenario.ReplicationTree.ReplNode.CommonHostProps.Port</code>	主主机连接端口号
<code>Scenario.ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore</code>	同步之前运行脚本

属性名称	说明
Scenario.ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore.Path	脚本路径
ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore.Args	脚本参数

启动方案: `run()`

`run` API 函数允许您运行方案。作为最佳实践，在调用 `run` API 之前，请调用 [add_credentials_ex](#) (p. 24) API 验证主主机。

参数

`run` API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
scenario_id	uint	要启动的方案 ID。
sync_method	uint	可以是下列选项之一的同步方式： 0 - 文件同步 1 - 块同步 2 - 卷同步（仅用于完整系统方案。）
ignore_same_files	bool	忽略相同大小/时间的文件。
arc_up	bool	此参数仅适用于 Arcserve Backup 集成方案。方案不是 Arcserve Backup 集成参数时，您应当将此参数设为 <code>false</code> 。 方案与 Arcserve Backup 集成时，调用 <code>run</code> API 之前，请调用 add_bab_credentials (p. 24) API。
verification_and_run	uint	保留。必须始终定义为 1。
message	out string	包含此 API 失败时错误的详细说明。

返回值

方案成功启动时，此函数返回 `true`。否则，此函数返回 `false`，并在 `message` 参数中提供错误的详细说明。

示例

```
public bool run_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        //0: 文件同步; 1:阻止同步; 2: 卷同步
        uint sync_method = 0;
        bool ignore_same_files = true;
        bool arc_integrated = false;
        string message = "";
        return get_mng().run(_session_id, scenario_id, sync_method,
ignore_same_files, arc_integrated, 1, out message);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

停止方案： `stop()`

`stop` API 函数允许您停止运行方案。

参数

`stop` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	要停止的方案 ID。
<code>execute_sync</code>	<code>bool</code>	控制同步还是异步调用此 API。将此参数设置为 <code>true</code> 时，API 函数才将不返回值，直到方案停止。否则，函数将立即返回值。
<code>why_not_reason</code>	<code>out string</code>	包含此 API 失败时错误的详细说明。

返回值

返回类型是布尔值。返回值为 true 时，API 成功完成。返回值为 false 时，API 未成功完成。如果返回值是 false，复查 why_not_reason 参数中返回的消息，以确定 API 为何失败。

示例

```
public bool stop_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string why_not_reason = "";
        bool execute_sync = true;
        return get_mng().stop(_session_id, scenario_id, execute_sync,
out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

暂停数据复制: suspend_replication()

suspend_replication API 允许您挂起主机的复制操作。

参数

suspend_replication API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
scenario_id	uint	要挂起的方案 ID。
replica_index	uint	方案中的副本主机索引。通常，副本主机索引的值为 2。
execute_sync	bool	控制同步还是异步调用此 API。将此参数设置为 true 时，API 函数才将不返回值，直到操作完成。否则，函数将立即返回值。
message	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 true 时，API 成功完成。返回值为 false 时，API 未成功完成。如果返回值为 false，查看消息以确定 API 失败的原因。

示例

```
public bool suspend_replication_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string message = "";
        bool execute_sync = true;
        uint replica_index = 2;
        return get_mng().suspend_replication(_session_id, scenario_id,
replica_index, execute_sync, out message);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

完成工作：close_session()

通过 close_session API 函数，您可以从 Control Service 注销。登录 Control Service 后，您调用 close_session 从 Control Service 注销。

参数

close_session API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 true 时，该会话已关闭。返回值为 false 时，该会话未关闭。

示例

```
public bool close_session_example()
{
    try
    {
        string why_not_reason = "";
        return get_mng().close_session(_session_id, out
why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

管理凭据: [add_credentials_ex\(\)](#)

`add_credentials_ex` API 函数允许您添加主机的凭据。

参数

`add_credentials_ex` API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) 函数 API 返回的会话 ID。
scenario_id	uint	要增加凭据的方案 ID。
host_name	string	主机 IP 地址。 示例: 155.35.76.44。
port	uint	引擎端口号。通常，端口号为 25000。
user_name	string	用户名。 示例: Administrator
password	string	用户名的密码。 示例: Arcserve
domain_name	string	域名。 示例: arcserve.com

返回值

返回类型是 `ulong`。返回值为零表示 API 成功完成。返回值不等于零时，API 失败。

示例

```
public bool add_credentials_ex_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        //添加主服务器的凭据
        string host_ip = "155.35.66.138";
        uint port = 25000;
        string user_name = "administrator";
        string password = "caworld";
        string domain_name = "155.35.66.138";
        ulong res = get_mng().add_credentials_ex(_session_id,
scenario_id, host_ip, port, user_name, password, domain_name);
        //添加副本服务器的凭据
        host_ip = "155.35.66.142";
        domain_name = "155.35.66.142";
        res = get_mng().add_credentials_ex(_session_id, scenario_id,
host_ip, port, user_name, password, domain_name);
        return (res == 0);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

与 Arcserve Backup 集成: `add_bab_credentials()`

`add_bab_credentials` API 函数允许您添加凭据，以便您可以访问 Arcserve Backup。

参数

`add_bab_credentials` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	要增加 Arcserve 凭据的方案 ID。

名称	类型	说明
username	string	用户名。 示例: Administrator
password	string	用户名的密码。 示例: Arcserve
async_id	out ulong	异步调用此 API 函数时, <code>async_id</code> 参数将包含非零值, 这允许函数等待操作完成并且检索结果。

返回值

返回类型是布尔值。返回值为 `true` 时, API 成功完成。返回值为 `false` 时, API 未成功完成。

示例

```
public bool add_bab_credentials_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string username = "admin";
        string password = "caworld";
        ulong async_id = 0;
        bool res = get_mng().add_bab_credentials(_session_id,
scenario_id,username,password,out async_id);
        return res;
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

更新方案属性: `update_scenario()`

`update_scenario` API 函数允许您在运行时更新方案数据。

参数

`update_scenario` API 函数包括下表所述参数:

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 <code>create_session</code> API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	要更新的方案组 ID。
<code>scenario_data_str</code>	<code>ref string</code>	关于方案的数据。在操作成功完成时，即提供新的方案数据。
<code>why_not_reason</code>	<code>ref string</code>	此 API 失败时，包含失败原因。

返回值

返回值以 XML 格式为方案提供数据。

示例

```
public bool update_scenario_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string why_not_reason = "";
        string scenario_data_str =
get_mng().get_scenario_data(scenario_id);
        //为方案做一些更改。
        return get_mng().update_scenario(_session_id, scenario_id, ref
scenario_data_str, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

删除方案: `remove_scenario()`

`remove_scenario` API 函数允许您从 Control Service 中删除方案。

参数

`remove_scenario` API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
scenario_id	uint	将运行的方案 ID。
arc_upd	bool	此参数仅适用于 Arcserve Backup 集成方案。
why_not-reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool remove_scenario_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        bool arc_integrated = false;
        string why_not_reason = "";
        return get_mng().remove_scenario(_session_id, scenario_id,
arc_integrated, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

导入方案: import_scenario()

import_scenario API 函数允许您将方案导入到 Control Service。

参数

import_scenario API 函数包括下表所述参数:

类型	名称	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
group_id	uint	要导入的方案组 ID。如果不確定组 ID, 将此设置为 0xFFFFFFFF。
scenario_id	out uint	API 成功完成时, 检索方案 id。
scenario_data	string	关于方案的数据。通常地, 您从方案文件检索方案数据字符串。数据格式为 XML。
why_not_reason	out string	此 API 失败时, 包含失败原因。

返回值

返回值以 XML 格式为方案提供数据。

示例

```
public bool import_scenario_example()
{
    try
    {
        uint scenario_id = 0;
        string why_not_reason = "";
        string scenario_data = "load the data from a scenario file.";
        uint group_id = 0xFFFFFFFF;
        return get_mng().import_scenario(_session_id,
group_id,scenario_data,out scenario_id,out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

同步运行方案: `synchronize()`

`synchronize` API 函数允许您为方案同步数据。

参数

`synchronize` API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
scenario_id	uint	要同步的方案组 ID。
sync_method	uint	可以是下列选项之一的同步方式: 0 - 文件同步 1 - 块同步 2 - 卷同步(仅用于完整系统方案。)
ignore_same_files	bool	忽略相同大小/时间的文件。
execute_sync	bool	控制同步还是异步调用此 API。
message	out string	此 API 失败时, 包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时, API 成功完成。返回值为 `false` 时, API 未成功完成。如果返回值为 `false`, 查看消息以确定 API 失败的原因。

示例

```
public bool synchronize_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string message = "";
        bool execute_sync = true;
        uint sync_method = 1;
        bool ignore_same_files = false;
        return get_mng().synchronize(_session_id, scenario_id,
sync_method, ignore_same_files, execute_sync, out message);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

恢复复制: resume_replication()

resume_replication API 函数允许您恢复主机的复制操作。

参数

suspend_replication API 函数包括下表所述参数:

Name	Type	说明
session_id	uint	通过调用 create_session API 返回的会话 ID。
scenario_id	uint	要恢复复制的方案 ID。
replica_index	uint	方案中的副本主机索引。通常，副本主机索引的值为 2。这是要挂起复制的主机。例如，数据更改将汇集于缓冲池中，而非将数据复制到磁盘，直到复制操作恢复。
execute_sync	bool	控制同步还是异步调用此 API。
message	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool resume_replication_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string message = "";
        bool execute_sync = true;
        uint replica_index = 2;
        return get_mng().resume_replication(_session_id, scenario_id,
replica_index, execute_sync, out message);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

添加回滚书签: `set_rewind_bookmark()`

`set_rewind_bookmark` API 函数允许您为方案设置书签。

参数

`set_rewind_bookmark` API 函数包括下表所述参数:

名称	类型	说明
scenario_id	string	要设置书签的方案 ID。
host_index	uint	始终 = 1。
bookmark_msg	string	书签名称。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool set_rewind_bookmark_example()
{
    try
    {
        string scenario_id = _scenario_id.ToString();
        uint host_index = 1;
        string why_not_reason = "";
        string bookmark_msg = "test bookmark";
        return get_mng().set_rewind_bookmark(scenario_id, host_index,
bookmark_msg, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

高可用性方案管理 API

以下部分描述管理高可用性方案的 API。

本节包括以下主题：

[触发保障恢复: `start_ar\(\)` \(p. 33\)](#)

[禁用高可用性方案中的心跳监视: `stop_is_alive\(\)` \(p. 35\)](#)

[切换高可用性方案: `switchover\(\)` \(p. 36\)](#)

[启用高可用性方案中的心跳监视: `start_is_alive\(\)` \(p. 39\)](#)

触发保障恢复: `start_ar()`

`start_ar` API 函数允许您为方案执行保障恢复操作 (AR)。在执行自动 AR 时，您不需要调用其他 API 来停止 AR。在 AR 操作完成后，AR 将停止。在您执行手动 AR 时，请调用 API [resume_application](#) (p. 31) 来停止 AR 操作。

参数

`start_ar` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) API 返回的会话 ID。

名称	类型	说明
scenario_id	uint	要执行保障恢复的方案 ID。
replica_index	uint	方案中的副本主机索引。通常，副本主机索引的值为 2。这是要挂起复制的主机。
auto_ar	bool	自动或手动运行 AR。 <ul style="list-style-type: none"> ■ True - 自动 ■ False - 手动
execute_sync	bool	控制同步还是异步调用此 API。
message	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool start_ar_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string message = "";
        bool execute_sync = true;
        uint replica_index = 2;
        bool auto_ar = true;
        return get_mng().start_ar(_session_id, scenario_id,
replica_index, auto_ar, execute_sync, out message);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

禁用高可用性方案中的心跳监视: `stop_is_alive()`

`Is-alive` 是副本服务器发送给识别节点状态的主服务器的电子信号。高可用性方案运行的同时，副本服务器周期性将电子信号 (`ping`) 发送到主服务器。默认情况下，`ping` 的频率是 30 秒。副本服务器在预定时段经过（默认是 300 秒）之后无法 `ping` 到主机时，您可以触发切换事件。

`stop_is_alive` API 函数允许您挂起仍在运行检查。

参数

`stop_is_alive` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	要挂起仍在运行检查的方案 ID。
<code>execute_sync</code>	<code>bool</code>	控制同步还是异步调用此 API。
<code>err_message</code>	<code>out string</code>	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool stop_is_alive_example()
{
    try
    {
        uint scenario_id = _ha_scenario_id;
        string err_messages = "";
        bool execute_sync = true;
        return get_mng().stop_is_alive(session_id, scenario_id,
execute_sync, out err_messages);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

切换高可用性方案: `switchover()`

`switchover` API 函数允许您执行切换操作。

使用完整系统、高可用性方案，您可以执行切换操作到任何副本服务器。要切换到非故障切换的副本服务器时，在调用切换 API 之前调用 `execute_action` API。

注意：`execute_action` API 在示例中进行说明。

参数

`switchover` API 函数包括下表所述参数：

名称	类型	说明
<code>session_id</code>	<code>uint</code>	通过调用 create_session (p. 9) API 返回的会话 ID。
<code>scenario_id</code>	<code>uint</code>	要执行切换操作的方案 ID。
<code>execute_sync</code>	<code>bool</code>	控制同步还是异步调用此 API。
<code>run_reverse_scenario</code>	<code>bool</code>	切换操作发生后，运行或不运行向后方案。
<code>err_message</code>	<code>out string</code>	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

示例 1

```
public bool switchover_example()
{
    try
    {
        uint scenario_id = _ha_scenario_id;
        string err_messages = "";
        bool execute_sync = true;
        bool run_reverse_scenario = false;
        return get_mng().switchover(_session_id, scenario_id,
execute_sync, run_reverse_scenario, out err_messages);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

示例 2

```
public bool switchover_2nd_example()
{
    try
    {
set_xcmd_data("switchover", "switchover_index","3" );
        uint scenario_id = _ha_scenario_id;
        string err_messages = "";
        bool execute_sync = true;
        bool run_reverse_scenario = false;
        return get_mng().switchover(_session_id, scenario_id,
execute_sync, run_reverse_scenario, out err_messages);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

示例 3

默认情况下，复制和高可用性执行切换操作到预定义的故障转移、副本服务器。使用完整系统、高可用性方案，您可以切换到非故障切换的副本服务器。不过，要使用切换 API 切换到非故障切换服务器时，在调用切换 API 之前调用 execute_action API，以下示例进行说明：

```
set_xcmd_data("switchover", "switchover_index", "3" );

public bool set_xcmd_data(string cmd_name_str, string cmd_data_str, string cmd_value_str)
{
    try
    {
        string result_data = "";
        string action_data;

        XmlDocument doc = new XmlDocument();
        XmlNode actions = doc.CreateNode(XmlNodeType.Element,
xomngapi.WANSync_c.xo_actions, "");

        XmlNode commonNode = doc.CreateNode(XmlNodeType.Element,
xomngapi.WANSync_c.action_common_lab, "");
        XmlAttribute attrSession =
doc.CreateAttribute(xomngapi.WANSync_c.action_com_session_id);
        XmlAttribute attrScenario =
doc.CreateAttribute(xomngapi.WANSync_c.action_com_scenario_id);
        XmlAttribute attrHostindex =
doc.CreateAttribute(xomngapi.WANSync_c.action_com_host_index);
        XmlAttribute attrUsedfor =
doc.CreateAttribute(xomngapi.WANSync_c.action_used_for);

        attrSession.Value =
xomngapi.WANSync_c.WANSync.session_id.ToString();
        attrScenario.Value = this.id.ToString();
        attrUsedfor.Value = xomngapi.WANSync_c.action_x_command_data;

        commonNode.Attributes.Append(attrSession);
        commonNode.Attributes.Append(attrScenario);
        commonNode.Attributes.Append(attrHostindex);
        commonNode.Attributes.Append(attrUsedfor);

        XmlNode xo_cmd = doc.CreateNode(XmlNodeType.Element,
xomngapi.WANSync_c.xo_cmd, "");
        XmlAttribute cmd_name =
doc.CreateAttribute(xomngapi.WANSync_c.action_cmd_name);
        XmlAttribute cmd_data =
doc.CreateAttribute(xomngapi.WANSync_c.action_cmd_data);
        XmlAttribute cmd_value =
doc.CreateAttribute(xomngapi.WANSync_c.action_cmd_value);

        cmd_name.Value = cmd_name_str;
        cmd_data.Value = cmd_data_str;
```

```

cmd_value.Value = cmd_value_str;

xo_cmd.Attributes.Append(cmd_name);
xo_cmd.Attributes.Append(cmd_data);
xo_cmd.Attributes.Append(cmd_value);

actions.AppendChild(commonNode);
commonNode.AppendChild(xo_cmd);
doc.AppendChild(actions);
action_data = doc.OuterXml;
string error;
return get_mng().execute_action(action_data, true, out
result_data, out error);
}
catch (System.Exception)
{
    return false;
}

```

启用高可用性方案中的心跳监视: `start_is_alive()`

`Is-alive` 是副本服务器发送给识别节点状态的主服务器的电子信号。高可用性方案运行的同时，副本服务器周期性将电子信号 (`ping`) 发送到主服务器。默认情况下，`ping` 的频率是 30 秒。副本服务器在预定时段经过（默认是 300 秒）之后无法 `ping` 到主机时，您可以触发切换事件。

`start_is_alive` API 函数允许您恢复仍在运行检查。

参数

`start_is_alive` API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
scenario_id	uint	要启动仍在运行检查的方案 ID。
execute_sync	bool	控制同步还是异步调用此 API。
err_message	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool start_is_alive_example()
{
    try
    {
        uint scenario_id = _ha_scenario_id;
        string err_messages = "";
        bool execute_sync = true;
        return get_mng().start_is_alive(session_id, scenario_id,
execute_sync, out err_messages);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

VSS 快照管理 API

以下部分描述管理 VSS 快照的 API。

本节包括以下主题：

[在副本服务器上安装 VSS 快照: mount_snapshot\(\) \(p. 40\)](#)

[从副本服务器卸载 VSS 快照: unmount_snapshot\(\) \(p. 42\)](#)

[从副本服务器中删除 VSS 快照: delete_snapshot\(\) \(p. 43\)](#)

[从副本服务器获取 VSS 快照列表: get_snapshot_list\(\) \(p. 44\)](#)

在副本服务器上安装 VSS 快照: mount_snapshot()

mount_snapshot API 函数允许您将 VSS 快照安装到副本服务器上的特定文件夹。

参数

mount_snapshot API 函数包括下表所述参数：

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
host_name	string	引擎主机名
ip_string	string	host_name 的 IP 地址

名称	类型	说明
host_port	ushort	引擎端口号。通常，端口号为 25000。
mount_path	string	要安装快照的文件夹。
snapshot_id	string	VSS 快照 ID。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool mount_snapshot_example()
{
    try
    {
        string host_name = "155.35.66.142";
        string ip_string = "155.35.66.142";
        ushort host_port = 25000;
        string mount_path = "c:/mount";
        string snapshot_id = "{9CFDE664-62D5-4fd8-A304-2B664900B98F}";
        string why_not_reason = "";
        return get_mng().mount_snapshot(session_id, host_name,
ip_string, host_port, snapshot_id, mount_path, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

从副本服务器卸载 VSS 快照:unmount_snapshot()

unmount_snapshot API 函数允许您从文件夹卸载 VSS 快照。

参数

unmount_snapshot API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
host_name	string	引擎主机名
ip_string	string	host_name 的 IP 地址
host_port	ushort	引擎端口号。通常，端口号为 25000。
snapshot_id	string	VSS 快照 ID。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool unmount_snapshot_example()
{
    try
    {
        string host_name = "155.35.66.142";
        string ip_string = "155.35.66.142";
        ushort host_port = 25000;
        string snapshot_id = "{9CFDE664-62D5-4fd8-A304-2B664900B98F}";
        string why_not_reason = "";
        return get_mng().unmount_snapshot(session_id, host_name,
ip_string, host_port, snapshot_id, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

从副本服务器中删除 VSS 快照: `delete_snapshot()`

`delete_snapshot` API 函数允许您从副本服务器中删除 VSS 快照。

参数

`delete_snapshot` API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
host_name	string	引擎主机名
ip_string	string	host_name 的 IP 地址
host_port	ushort	引擎端口号。通常，端口号为 25000。
snapshot_id	string	VSS 快照 ID。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool delete_snapshot_example()
{
    try
    {
        string host_name = "155.35.66.142";
        string ip_string = "155.35.66.142";
        ushort host_port = 25000;
        string snapshot_id = "{9CFDE664-62D5-4fd8-A304-2B664900B98F}";
        string why_not_reason = "";
        return get_mng().delete_snapshot(session_id, host_name,
ip_string, host_port, snapshot_id, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

从副本服务器获取 VSS 快照列表: `get_snapshot_list()`

`get_snapshot_list` API 函数允许您从主机获取 VSS 快照列表。

参数

`get_snapshot_list` API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
host_name	string	引擎主机名
ip_string	string	host_name 的 IP 地址
host_port	ushort	引擎端口号。通常，端口号为 25000。
snapshot_list	out string	卷快照列表。
why_not_reason	out string	此 API 失败时，包含失败原因。

返回值

返回类型是布尔值。返回值为 `true` 时，API 成功完成。返回值为 `false` 时，API 未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

```
public bool get_snapshot_list_example()
{
    try
    {
        string host_name = "155.35.66.142";
        string ip_string = "155.35.66.142";
        ushort host_port = 25000;
        string snapshot_list = "";
        string why_not_reason = "";
        return get_mng().get_snapshot_list(session_id, host_name,
ip_string, host_port, out snapshot_list, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

收集统计 API

以下部分描述管理收集方案统计信息过程的 API。

本节包括以下主题：

[获取扩展方案统计信息: get_data_ex\(\) \(p. 45\)](#)

[获取方案统计信息: get_scenario_data\(\) \(p. 52\)](#)

获取扩展方案统计信息: `get_data_ex()`

`get_data_ex` API 函数允许您获取所有方案信息，如下：

- 方案事件
- 方案状态
- 同步和复制统计信息

参数 `manager_data` 是 XML 格式字符串。数据包含所有方案信息。例如，方案状态（运行、停止等等）、事件、方案统计信息等。XML 格式字符串类似以下内容：

```
<?xml version="1.0"?>
<manager_data>
- <scenarios>
+ <scenario signature="4636778060728034734" ha_type="Forward" is_arcservc_integrated="False" is_cdp="Fa
id="1094498606">
+ <scenario signature="4334615870148788711" is_arcservc_integrated="False" is_cdp="False" is_ass_rec="Fa
+ <scenario signature="14684688067413199200" ha_type="Forward" is_arcservc_integrated="False" is_cdp="F
+ <scenario signature="15270013466011305310" ha_type="Forward" is_arcservc_integrated="False" is_cdp="F
id="43557253">
+ <scenario signature="5773759741404806146" is_arcservc_integrated="False" is_cdp="False" is_ass_rec="Fa
+ <scenario signature="7020398949829650879" is_arcservc_integrated="False" is_cdp="False" is_ass_rec="Fa
+ <scenario signature="7044671085026122361" is_arcservc_integrated="False" is_cdp="False" is_ass_rec="Tru
+ <scenario signature="7920379657132428156" ha_type="Forward" is_arcservc_integrated="False" is_cdp="Fa
id="3423940998">
</scenarios>
+ <scenario_groups>
</manager_data>
```

以下部分演示如何使用 XML 字符串。

参数

`get_data_ex` API 包括下表所述参数：

名称	类型	说明
<code>session_id</code>	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
<code>scenarios_with_statistics</code>	uint	方案 ID 的数组。检索方案的统计。
<code>last_update_time</code>	ulong	上次更新的时间戳。

名称	类型	说明
request_flag	uint	请求数据类型。值可以是: 1 - 方案数据 2 - cdp 数据[不使用] 4 - 主机管理数据 8 - 快照数据 15 - 上述全部数据
manager_data	out string	以 XML 格式返回方案的数据。

返回值

返回类型是布尔值。返回值为 `true` 时，该命令成功完成。返回值为 `false` 时，该命令未成功完成。如果返回值为 `false`，查看消息以确定 API 失败的原因。

示例

示例 1:

```
public bool get_data_ex_example()

{
    try
    {
        uint[] scenarios_with_statistics = new uint[] { _scenario_id };

        uint request_flag = 1;

        ulong last_update_time = 0;

        string manager_data = "";

        bool res = get_mng().get_data_ex(_session_id,
scenarios_with_statistics, request_flag, ref last_update_time, out
manager_data);

        return res;
    }

    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }

    return false;
}
```

示例 2:

此函数解析 `get_data_ex()` 函数返回的 XML 缓冲区数据 (`manager_data`)。下列示例描述在方案正在运行或停止时如何获得其状态：

```
string get_scenario_state (string manager_data, string scenario_name_or_id)
{
    string scenario_state = "unknown";
    XmlDocument manager_data_doc = new XmlDocument();
    manager_data_doc.LoadXml(manager_data);
    XmlNode root_node =
    manager_data_doc.SelectSingleNode("./manager_data");
    if (root_node == null)
    {
        return scenario_state;
    }
    //get all the scenario data information
    XmlNode scenario_nodes =
    root_node.SelectSingleNode("./scenarios");
    if (scenario_nodes == null)
    {
        return scenario_state;
    }
    foreach (XmlNode scenario_node in scenario_nodes.ChildNodes)
    {
        uint scenario_id = 0;
        string scenario_name = "";
        XmlAttribute id_attr = scenario_node.Attributes["id"];
        if (id_attr != null)
            scenario_id = Convert.ToInt32(id_attr.Value);
        XmlAttribute name_attr = scenario_node.Attributes["name"];
        if (name_attr != null)
            scenario_name = name_attr.Value;

        if (scenario_name_or_id != scenario_id.ToString() &&
scenario_name_or_id.ToLower() != scenario_name.ToLower())
            continue;

        //get the scenario status, running or stopped
        foreach (XmlNode node in scenario_node.ChildNodes)
        {
            //get the scenario state, running or stop
            if (0 == string.Compare("state", node.Name, true))
            {
                if (node.Attributes["val"] != null)
                {
                    scenario_state =
node.Attributes["val"].Value.ToLower();
                }
            }
        }
    }
    return scenario_state;
}
```

```
}
```

示例 3:

此函数解析 `get_data_ex()` 函数返回的 XML 缓冲区数据 (`manager_data`)。下列示例描述如何获得方案的所有事件:

```
void get_scenario_events(string manager_data, string scenario_name_or_id, ref
ArrayList events)
{
    XmlDocument manager_data_doc = new XmlDocument();
    manager_data_doc.LoadXml(manager_data);
    XmlNode root_node =
    manager_data_doc.SelectSingleNode("./manager_data");
    if (root_node == null)
    {
        return ;
    }
    //get all the scenario data information
    XmlNode scenario_nodes =
    root_node.SelectSingleNode("./scenarios");
    if (scenario_nodes == null)
    {
        return ;
    }
    foreach (XmlNode scenario_node in scenario_nodes.ChildNodes)
    {
        uint scenario_id = 0;
        string scenario_name = "";
        XmlAttribute id_attr = scenario_node.Attributes["id"];
        if (id_attr != null)
            scenario_id = Convert.ToInt32(id_attr.Value);
        XmlAttribute name_attr = scenario_node.Attributes["name"];
        if (name_attr != null)
            scenario_name = name_attr.Value;

        if (scenario_name_or_id != scenario_id.ToString() &&
scenario_name_or_id.ToLower() != scenario_name.ToLower())
            continue;

        //get the scenario status, running or stopped
        foreach (XmlNode node in scenario_node.ChildNodes)
        {
            //get the scenario state, running or stop
            if (0 == string.Compare("gen", node.Name, true))
            {
                events.Add(new event_data_c(node));
            }
        }
    }
}
```

示例 4:

您定义 `scenarios_with_statistics` 参数，以便获得方案的同步和复制统计信息。参数是数组。要获得多个方案的统计信息，将方案的 ID 添加到数组中。

`get_data_ex` 获得以下统计信息：

注意：下列屏幕说明传输和同步统计信息的过程。

已传输到副本主机的字节数：					
主机	已发送数据总量	当前文件名	要发送的数据	传输速度	当前进度
155.35.75.216	47.77 KB	0	0 字节	32 位/秒	

最后 同步 统计信息：块级数据同步

同步进度：							
• 155.35.75.96 -> 155.35.75.216							
C:/	状态	文件数	总计大小	比较进度	要发送的数据	发送进度	开始时间
已初始化	80024	16.05 GB	0.0 %	0 字节			2013/1/4 14:20:52

代码

此函数解析 `get_data_ex()` 函数返回的 XML 缓冲区数据 (`manager_data`)。以下代码演示如何获取传输和同步的统计。（请参阅前面的屏幕。）

```
void get_scenario_sync_statistics(string manager_data, string
scenario_name_or_id, ref ArrayList sync_statistics)

{
    XmlDocument manager_data_doc = new XmlDocument();
    manager_data_doc.LoadXml(manager_data);
    XmlNode root_node =
manager_data_doc.SelectSingleNode("./manager_data");
    if (root_node == null)
    {
        return;
    }
    //get all the scenario data information
    XmlNode scenario_nodes =
root_node.SelectSingleNode("./scenarios");
    if (scenario_nodes == null)
    {
        return;
    }
    foreach (XmlNode scenario_node in scenario_nodes.ChildNodes)
    {
        uint scenario_id = 0;
        string scenario_name = "";
        XmlAttribute id_attr = scenario_node.Attributes["id"];
        if (id_attr != null)
            scenario_id = Convert.ToInt32(id_attr.Value);
        XmlAttribute name_attr = scenario_node.Attributes["name"];
        if (name_attr != null)
            scenario_name = name_attr.Value;

        if (scenario_name_or_id != scenario_id.ToString() &&
scenario_name_or_id.ToLower() != scenario_name.ToLower())
            continue;

        //get the scenario status, running or stopped
        foreach (XmlNode node in scenario_node.ChildNodes)
        {
            //get the scenario state, running or stop
            if (0 == string.Compare("statistics", node.Name, true))
            {
                sync_statistics.Add(new host_statistics_c(node));
            }
        }
    }
}
```

示例 5:

以下代码演示如何检索方案的同步统计。

```
ArrayList sync_statistics = new ArrayList();
    get_scenario_sync_statistics(manager_data, "FileServer 1", ref
sync_statistics);
    //show the statistics
    foreach (host_statistics_c stat in sync_statistics)
    {
        //host name
        string host_name = stat.host_name;
        //transmission statistics
        foreach (transfer_to_replica_c trans in stat.trans_to_reps)
        {
            //handle the transfer data such as speed.
            //handle the transfer data such as speed.
        }
        //sync statistics
        foreach (sync_statistics_host_c sync_host in
stat.children_hosts)
        {
            //root directory
            foreach(sync_statistics_root_dir_c root_dir in
sync_host.sync_root_dirs)
            {
                //root_dir.total_size
            }
        }
    }
```

获取方案统计信息: [get_scenario_data\(\)](#)

`get_scenario_data` API 函数允许您获取方案 ID。

参数

`get_scenario_data` API 函数包括下表所述参数:

名称	类型	说明
session_id	uint	通过调用 create_session (p. 9) API 返回的会话 ID。
scenario_id	uint	方案 ID。

返回值

返回值为方案提供数据。

示例

```
public bool get_scenario_data_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        string scenario_data_str =
get_mng().get_scenario_data(session_id.scenario_id);
        return true;
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}
```

