

Arcserve® Replication and High Availability

SOAP API Reference Guide

r16.5

arcserve®

Pre-release Document, only for reference

Esta documentación, que incluye sistemas incrustados de ayuda y materiales distribuidos por medios electrónicos (en adelante, referidos como la "Documentación") se proporciona con el único propósito de informar al usuario final, pudiendo Arcserve proceder a su modificación o retirada en cualquier momento.

Queda prohibida la copia, transferencia, reproducción, divulgación, modificación o duplicación de la totalidad o parte de esta Documentación sin el consentimiento previo y por escrito de Arcserve. Esta Documentación es información confidencial, propiedad de Arcserve, y no puede ser divulgada por Vd. ni puede ser utilizada para ningún otro propósito distinto, a menos que haya sido autorizado en virtud de (i) un acuerdo suscrito aparte entre Vd. y Arcserve que rijan su uso del software de Arcserve al que se refiere la Documentación; o (ii) un acuerdo de confidencialidad suscrito aparte entre Vd. y Arcserve.

No obstante lo anterior, si dispone de licencias de los productos informáticos a los que se hace referencia en la Documentación, Vd. puede imprimir, o procurar de alguna otra forma, un número razonable de copias de la Documentación, que serán exclusivamente para uso interno de Vd. y de sus empleados, y cuyo uso deberá guardar relación con dichos productos. En cualquier caso, en dichas copias deberán figurar los avisos e inscripciones relativas a los derechos de autor de Arcserve.

Este derecho a realizar copias de la Documentación sólo tendrá validez durante el período en que la licencia aplicable para el software en cuestión esté en vigor. En caso de terminarse la licencia por cualquier razón, Vd. es el responsable de certificar por escrito a Arcserve que todas las copias, totales o parciales, de la Documentación, han sido devueltas a Arcserve o, en su caso, destruidas.

EN LA MEDIDA EN QUE LA LEY APLICABLE LO PERMITA, ARCSERVE PROPORCIONA ESTA DOCUMENTACIÓN "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO INCLUIDAS, ENTRE OTRAS PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, ADECUACIÓN A UN FIN CONCRETO Y NO INCUMPLIMIENTO. ARCSERVE NO RESPONDERÁ EN NINGÚN CASO, ANTE VD. NI ANTE TERCEROS, EN LOS SUPUESTOS DE DEMANDAS POR PÉRDIDAS O DAÑOS, DIRECTOS O INDIRECTOS, QUE SE DERIVEN DEL USO DE ESTA DOCUMENTACIÓN INCLUYENDO A TÍTULO ENUNCIATIVO PERO SIN LIMITARSE A ELLO, LA PÉRDIDA DE BENEFICIOS Y DE INVERSIONES, LA INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL, LA PÉRDIDA DEL FONDO DE COMERCIO O LA PÉRDIDA DE DATOS, INCLUSO CUANDO ARCSERVE HUBIERA PODIDO SER ADVERTIDA CON ANTELACIÓN Y EXPRESAMENTE DE LA POSIBILIDAD DE DICHAS PÉRDIDAS O DAÑOS.

El uso de cualquier producto informático al que se haga referencia en la Documentación se regirá por el acuerdo de licencia aplicable. Los términos de este aviso no modifican, en modo alguno, dicho acuerdo de licencia.

Arcserve es el fabricante de esta Documentación.

Esta Documentación presenta Derechos restringidos. El uso, la duplicación o la divulgación por parte del gobierno de los Estados Unidos está sujeta a las restricciones establecidas en las secciones 12.212, 52.227-14 y 52.227-19(c)(1) - (2) de FAR y en la sección 252.227-7014(b)(3) de DFARS, según corresponda, o en posteriores.

© 2015 Arcserve y sus empresas subsidiarias o afiliadas. Todos los derechos reservados. Las marcas registradas o de copyright de terceros son propiedad de sus respectivos propietarios.

Referencias a productos de Arcserve

Este documento hace referencia a los siguientes Arcserve productos :

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

Contacto con Arcserve

El equipo de Soporte de Arcserve ofrece un conjunto importante de recursos para resolver las incidencias técnicas y proporciona un fácil acceso a la información relevante del producto.

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

Con el Soporte de Arcserve:

- Se puede poner en contacto directo con la misma biblioteca de información compartida internamente por nuestros expertos de Soporte de Arcserve. Este sitio le proporciona el acceso a los documentos de nuestra base de conocimiento (KB). Desde aquí se pueden buscar fácilmente los artículos de la KB relacionados con el producto que contienen soluciones probadas para muchas de las principales incidencias y problemas comunes.
- Se puede utilizar nuestro vínculo Conversación en vivo para iniciar instantáneamente una conversación en tiempo real con el equipo de Soporte de Arcserve. Con la Conversación en vivo, se pueden obtener respuestas inmediatas a sus asuntos y preguntas, mientras todavía se mantiene acceso al producto.
- Se puede participar en la Comunidad global de usuarios de Arcserve para preguntar y responder a preguntas, compartir sugerencias y trucos, discutir prácticas recomendadas y participar en conversaciones con sus pares.
- Se puede abrir un ticket de soporte. Al abrir un ticket de soporte en línea se puede esperar una devolución de llamada de uno de nuestros expertos en el área del producto por el que está preguntando.

Se puede acceder a otros recursos útiles adecuados para su producto de Arcserve.

Documentation Changes

Desde la última versión de esta documentación, se han realizado estos cambios y actualizaciones:

- Ésta es la primera versión de esta guía de referencia.

Contenido

Capítulo 1: Introducción	7
Programación con la API de SOAP: el flujo.....	7
Descripción general de la API de SOAP	7
Convenciones utilizadas en esta Guía de referencia	8
Capítulo 2: API de SOAP	9
API de gestión de escenarios.....	9
Inicio del trabajo: create_session()	9
Creación de escenarios: create_scenari_ex()	11
Establecimiento de directorios raíz: set_root_dir()	13
Adición de directorios raíz: add_root_dir()	14
Adición de réplicas: add_replica()	15
Adición de varias réplicas: add_replica_ex()	16
Establecimiento de propiedades del escenario: set_scenari_data()	17
Establecimiento de propiedades del host: set_host_data()	18
Inicio de escenarios: run()	20
Detención de escenarios: stop()	21
Pausa de la replicación de datos: suspend_replication()	23
Finalización del trabajo: close_session()	24
Gestión de credenciales: add_credentials_ex()	25
Integración con Arcserve Backup: add_bab_credentials().....	27
Actualización de las propiedades del escenario: update_scenari()	28
Eliminación de escenarios: remove_scenari().....	29
Importación de escenarios: import_scenari()	30
Sincronización en la ejecución de escenarios: synchronize().....	31
Reanudación de la replicación: resume_replication().....	33
Adición de los marcadores de rebobinado: set_rewind_bookmark().....	34
API de gestión de escenarios de alta disponibilidad	35
Activación de la recuperación asegurada: start_ar()	36
Desactivación de la pulsación en los escenarios de alta disponibilidad: stop_is_alive()	37
Conmutación de los escenarios de alta disponibilidad: switchover()	38
Activación de las pulsaciones en los escenarios de alta disponibilidad: start_is_alive()	42
API de gestión de instantáneas de VSS	43
Montaje de la instantánea VSS en los servidores réplica: mount_snapshot()	44
Desmontaje de instantáneas VSS desde los servidores réplica: unmount_snapshot()	45
Eliminación de instantáneas VSS desde los servidores réplica: delete_snapshot()	46

Obtención de listas de instantáneas de VSS desde los servidores réplica: <code>get_snapshot_list()</code>	48
Recopilación de las API de estadísticas	49
Obtención de la estadística del escenario extendido: <code>get_data_ex()</code>	49
Obtención de la estadística de escenario: <code>get_scenario_data()</code>	57

Capítulo 3: Index

59

Capítulo 1: Introducción

Esta sección contiene los siguientes temas:

[Programación con la API de SOAP: el flujo](#) (en la página 7)

[Descripción general de la API de SOAP](#) (en la página 7)

[Convenciones utilizadas en esta Guía de referencia](#) (en la página 8)

Programación con la API de SOAP: el flujo

Para controlar Arcserve Replication and High Availability mediante la API de SOAP, el cliente debería autenticarse en el servicio de control de RHA y, a continuación, crear una sesión. Esta operación se logra invocando la función de API `create_session`. Después de que el cliente cree una sesión, podrá invocar cualquier función de API que se describe en esta guía de referencia. Cuando la tarea finaliza, el cliente llama a la API `close_session` para invalidar la sesión y liberar los recursos.

Descripción general de la API de SOAP

Esta guía describe la API de SOAP que exporta Arcserve Replication and High Availability. Las API permiten gestionar diversos aspectos de replicación y alta disponibilidad. Por ejemplo, se pueden controlar escenarios de RHA y se pueden realizar diversas operaciones como la ejecución o detención de escenarios y la realización de una operación de conmutación por error del escenario. Para usar la API de SOAP de Arcserve Replication and High Availability, se puede utilizar cualquier lenguaje de programación que contenga soporte de servicios Web integrado, como Visual Basic o C#.

Nota: El código de muestra proporcionado en esta guía de referencia se escribe en el lenguaje C#.

La dirección URL de la API de SOAP que se utiliza en el código de muestra es `http://127.0.0.1:8088/ws_man/xosoapapi.asmx`. Parte de la dirección de sustitución de la URL con la dirección IP de servicio de control de RHA o el nombre de host cuando se llama a la API.

Convenciones utilizadas en esta Guía de referencia

La Guía de referencia de la API de SOAP de Arcserve Replication and High Availability utiliza los tipos de argumento que se describen en la tabla siguiente:

Tipo	Descripción
bool	valor booleano
out uint	entero de salida
out ulong	entero largo de salida sin firmar
out string	cadena de salida
uint	entero sin firmar
ulong	entero largo sin firmar
ushort	entero corto sin firmar

Capítulo 2: API de SOAP

Esta sección contiene los siguientes temas:

- [API de gestión de escenarios](#) (en la página 9)
- [API de gestión de escenarios de alta disponibilidad](#) (en la página 35)
- [API de gestión de instantáneas de VSS](#) (en la página 43)
- [Recopilación de las API de estadísticas](#) (en la página 49)

API de gestión de escenarios

En las secciones siguientes se describen las API que permiten ejecutar tareas básicas de gestión de escenarios.

Esta sección incluye los siguientes temas:

- [Inicio del trabajo: `create_session\(\)`](#) (en la página 9)
- [Creación de escenarios: `create_scenario_ex\(\)`](#) (en la página 11)
- [Establecimiento de directorios raíz: `set_root_dir\(\)`](#) (en la página 13)
- [Adición de directorios raíz: `add_root_dir\(\)`](#) (en la página 14)
- [Adición de réplicas: `add_replica\(\)`](#) (en la página 15)
- [Adición de varias réplicas: `add_replica_ex\(\)`](#) (en la página 16)
- [Establecimiento de propiedades del escenario: `set_scenario_data\(\)`](#) (en la página 17)
- [Establecimiento de propiedades del host: `set_host_data\(\)`](#) (en la página 18)
- [Inicio de escenarios: `run\(\)`](#) (en la página 20)
- [Detención de escenarios: `stop\(\)`](#) (en la página 21)
- [Pausa de la replicación de datos: `suspend_replication\(\)`](#) (en la página 23)
- [Finalización del trabajo: `close_session\(\)`](#) (en la página 24)
- [Gestión de credenciales: `add_credentials_ex\(\)`](#) (en la página 25)
- [Integración con Arcserve Backup: `add_bab_credentials\(\)`](#) (en la página 27)
- [Actualización de las propiedades del escenario: `update_scenario\(\)`](#) (en la página 28)
- [Eliminación de escenarios: `remove_scenario\(\)`](#) (en la página 29)
- [Importación de escenarios: `import_scenario\(\)`](#) (en la página 30)
- [Sincronización en la ejecución de escenarios: `synchronize\(\)`](#) (en la página 31)
- [Reanudación de la replicación: `resume_replication\(\)`](#) (en la página 33)
- [Adición de los marcadores de rebobinado: `set_rewind_bookmark\(\)`](#) (en la página 34)

Inicio del trabajo: `create_session()`

La función de la API `create_session` permite la autenticación frente al servicio de control y devuelve el ID de la sesión. Transfiere el ID de sesión como un argumento cuando se llama a otras funciones de la API. La sesión abierta se puede invalidar mediante la [API `close_session`](#) (en la página 24).

Argumentos

La función de API create_session incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
user_name	cadena	El nombre de usuario. Ejemplo: MyDomain\Administrator
password	cadena	La contraseña para el nombre de usuario. Ejemplo: Arcserve
error_code	out uint	El valor cero indica que la función de API se ha ejecutado correctamente. Un valor no nulo indica un error de API.

Valores de retorno

Cuando la autenticación mediante create_session sea correcta, esta función devuelve un valor uint con el ID de sesión y un argumento de error_code establecido como 0. De lo contrario, se devuelve el valor uint MAX (0xFFFFFFFF) y el argumento error_code contiene el código de error detallado.

Nota: Debería especificar el argumento nombre_usuario en el formulario <NOMBRE_DOMINIO>\<NOMBRE_USUARIO>. Por ejemplo, test_dominio\Administrador.

Ejemplos

Ejemplo 1: Creación de un objeto de servicios web.

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

Ejemplo 2: Creación de una sesión.

```

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;
}

```

Creación de escenarios: create_scenari_ex()

La función de API create_scenari_ex permite crear escenarios. Después de crear los escenarios se pueden realizar operaciones como ejecutar el escenario, cambiar las propiedades del escenario, agregar o eliminar host, entre otros.

Argumentos

La función de API create_scenari_ex incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
producto	uint	Especifique un valor entero para los productos siguientes: 0 - Escenario de recuperación de desastres (DR) 1 - Escenario de alta disponibilidad (HA) 2 - Escenario de CD

Name	Tipo	Descripción
aplicación	uint	Especifique un valor entero para las aplicaciones siguientes: 0 - Escenario del servidor de archivos. 1- Escenario de Exchange 2 - Escenario de Oracle 3 - No utilizado 4 - Escenario de SQL Server 5 - Escenario de IIS 6 - Escenario del servicio de control 7 - Escenario de Hyper-V 8 - Escenario de SharePoint 9 - Escenario de vCenter 10 - Escenario de CRM 11- Escenario de todo el sistema 12 - Personalizar el escenario
is_ar	bool	Especifica si el escenario es compatible con la recuperación asegurada (AR): <ul style="list-style-type: none">■ verdadero: admite la recuperación asegurada■ falso: no admite la recuperación asegurada
is_cdp	bool	Siempre falso, no se utiliza.
integrate_opt	uint	Especifique un valor entero real para las aplicaciones siguientes: 0 - Sin integración 1 - Backup 2 - Arcserve D2D 3 - Arcserve Central Applications
group_id	uint	ID de grupo. Cuando group_id es 0xFFFFFFFF, el escenario pertenece al grupo de escenarios predeterminado, que se suele llamar escenario.

Name	Tipo	Descripción
scenario_data	out string	Los datos para el escenario. Después de haber llamado a esta API, el argumento contiene el datos del escenario actualizados. Nota: Para obtener más información, consulte el Ejemplo de este tema.
group_data	out string	Los datos del grupo.

Valores de retorno

Esta API devuelve el ID del escenario que se ha creado. Las otras API, como la de ejecución y detención, requieren el ID de escenario. Cuando el ID de escenario es cero, la API genera un fallo. Además, los valores que no son cero indican que la API se ha completado correctamente.

Nota: Cuando esta llamada API se completa correctamente, se crea el esqueleto del escenario. Todas las propiedades del escenario tienen los valores predeterminados. Las propiedades importantes como el directorio raíz, la IP del host, etc., están vacías. Se puede llamar a otras API para cumplimentar las propiedades. Las API subsiguientes de esta guía de referencia describen cómo cumplimentar las propiedades.

Ejemplo

```
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);
```

Establecimiento de directorios raíz: set_root_dir()

La función de API set_root_dir() permite editar la ruta del directorio raíz.

Argumentos

La función de API `set_root_dir` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	<code>uint</code>	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
<code>scenario_id</code>	<code>uint</code>	El ID del escenario.
<code>host_index</code>	<code>uint</code>	El índice del host que desea establecer.
<code>root_dir_index</code>	<code>uint</code>	El índice del directorio raíz, empezando desde cero; si se tienen dos directorios raíz, los índices son 0 y 1.
<code>root_dir</code>	<code>cadena</code>	El nombre de la carpeta. Nota: Este argumento es necesario solamente para los escenarios de todo el sistema. Sin embargo, esta función de API no admite actualmente el establecimiento de directorios raíz para escenarios de todo el sistema.

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Nota: La API [create_scenario_ex](#) (en la página 11) crea solamente un esqueleto del escenario. Se puede llamar esta API para establecer el directorio raíz.

Ejemplo

```
uint master_host_index = 1;  
get_mng().set_root_dir(_session_id, scenario_id, master_host_index, 0, "E:/test");
```

Adición de directorios raíz: `add_root_dir()`

La función de API `add_root_dir` permite agregar un directorio raíz nuevo para el escenario.

Argumentos

La función de API `add_root_dir` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	<code>uint</code>	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
<code>scenario_data</code>	<code>ref string</code>	Los datos para el escenario. Después de haber llamado a esta API, el argumento contiene el datos del escenario actualizados.
<code>root_directories</code>	<code>cadena</code>	El nombre de la carpeta.
<code>new_root_dir_index</code>	<code>out uint</code>	El índice del directorio raíz creado recientemente.

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Ejemplo: Cómo agregar un directorio raíz

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

```
//agregar directorio raíz
get_mng().add_root_dir(_session_id, ref scenario_data, "c:/test", out new_root_dir_id);
```

Adición de réplicas: `add_replica()`

La función de API `add_replica` permite agregar un host réplica nuevo para el escenario. Si desea replicar los datos master a más de un host réplica, se puede llamar a esta API para agregar algunos host réplica.

Argumentos

La función de API `add_replica` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	<code>uint</code>	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).

Name	Tipo	Descripción
scenario_data	ref string	Los datos para el escenario. Después de haber llamado a esta API, el argumento contiene el datos del escenario actualizados.
host_index	uint	El índice del host principal; el índice del host master es siempre 1; la primera réplica normalmente es 2.
new_replica_index	out uint	El índice del host réplica agregado recientemente.

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Ejemplo: Cómo agregar una réplica

```
//agregar una réplica en el host master
get_mng().add_replica(_session_id, ref scenario_data, master_host_index, out new_replica_index);
```

Adición de varias réplicas: add_replica_ex()

La función de API add_replica_ex permite agregar uno o más host de réplica nuevos al escenario de una sola vez.

Argumentos

La función de API add_replica_ex incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
scenario_data	ref string	Los datos para el escenario. Después de haber llamado a esta API, el argumento contiene el datos del escenario actualizados.
host_index	uint	El índice del host principal; el índice del host master es siempre 1; la primera réplica normalmente es 2.

Name	Tipo	Descripción
host_list	ref string	<p>La lista de host proporciona información sobre los host en formato xml. El xml contiene información sobre los host que se desea agregar al escenario.</p> <p>Nota: Consulte la pantalla que se muestra después de esta tabla.</p>

Por ejemplo, el código xml siguiente contiene la información sobre dos host.

```
<?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>
```

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Ejemplo: Cómo agregar varias réplicas

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

```
//agregar dos réplicas en el host master
get_mng().add_replica_ex(_session_id, ref escenario_data, master_host_index, ref host_list);
```

Establecimiento de propiedades del escenario: set_escenario_data()

La función de API set_escenario_data permite editar las propiedades del escenario. Esta API puede actualizar la mayoría de las propiedades del escenario.

Argumentos

La función de API `set_scenario_data` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
<code>scenario_id</code>	uint	El ID del escenario.
<code>property_name</code>	cadena	El nombre de la propiedad del host. Cada propiedad tiene un nombre único.
<code>property_value</code>	cadena	El valor de la propiedad.

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Ejemplo: Configuración de las propiedades del escenario

```
//establecer propiedades de datos del escenario  
get_mng().set_scenario_data(_session_id, scenario_id, "Scenario.ReplicateCompressAttr", "True");
```

La tabla siguiente enumera los nombres de la propiedad del escenario comunes.

Nombre de la propiedad	Descripción
<code>Scenario.ScenarioName</code>	El nombre del escenario
<code>Scenario.BuildShares</code>	Sincroniza el recurso compartido de Windows
<code>Scenario.SyncADS</code>	Réplica de NTFS ADS

Establecimiento de propiedades del host: `set_host_data()`

La función de API `set_host_data` permite editar las propiedades del host. Esta API puede actualizar la mayoría de las propiedades del host.

Argumentos

La función de API `set_host_data` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
<code>host_index</code>	uint	El índice del host.
<code>scenario_id</code>	uint	El ID del escenario.
<code>property_name</code>	cadena	El nombre de la propiedad del host. Cada propiedad tiene un nombre único.
<code>property_value</code>	cadena	El valor de la propiedad.

Valores de retorno

Esta función devuelve un valor verdadero cuando la llamada de API se completa correctamente. De lo contrario, esta función devuelve un valor falso.

Ejemplo: Configuración de las propiedades del host

```
//establecer el nombre de host y la dirección IP para el host
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");
```

La tabla siguiente enumera los nombres de la propiedad del host master comunes.

Nombre de la propiedad	Descripción
Scenario.ReplicationTree.ReplNode.CommonHostProps.Host	Nombre del host master
Scenario.ReplicationTree.ReplNode.CommonHostProps.IP	IP del host master
Scenario.ReplicationTree.ReplNode.CommonHostProps.Data_IP	Dirección IP de replicación master
Scenario.ReplicationTree.ReplNode.CommonHostProps.Port	Número de puerto de conexión del host master

Nombre de la propiedad	Descripción
Scenario.ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore	Ejecuta el script antes de la sincronización
Scenario.ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore.Path	La ruta del script
ReplicationTree.ReplNode.CommonHostProps.SyncScriptBefore.Args	Los argumentos del script

Inicio de escenarios: run()

La función de API de ejecución permite ejecutar escenarios. Como práctica recomendable, llame a la API [add_credentials_ex](#) (en la página 25) para autenticar el host máster antes de llamar a la API de ejecución.

Argumentos

La función de API de ejecución incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
scenario_id	uint	El ID de escenario que debe iniciarse.
sync_method	uint	El método de sincronización, que consiste en una de las opciones siguientes: 0: Sincronización de archivos 1: Sincronización de bloques 2: Sincronización de volúmenes (solo para el escenario Todo el sistema).
ignore_same_files	bool	Ignore los mismos archivos de tamaño/hora.
arc_upt	bool	Este parámetro se aplica solamente a los escenarios integrados de Arcserve Backup. Cuando el escenario no sea un parámetro integrado de Arcserve Backup, se debería establecer este parámetro como falso. Cuando el escenario se integra con Arcserve Backup, llame a la API add_bab_credentials (en la página 25) antes de llamar a la API de ejecución.
verification_and_run	uint	Reservado. Debe definirse siempre como 1.
mensaje	out string	Contiene una descripción detallada del error cuando se produce un error en esta API.

Valores de retorno

Esta función devuelve un valor verdadero cuando el escenario se inicia correctamente. De lo contrario, esta función devuelve un valor falso y proporciona una descripción detallada del error en el argumento del mensaje.

Ejemplo

```
public bool run_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        //0:File Sync;1:Block Sync;2:Volume Sync
        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;
}
```

Detención de escenarios: stop()

La función de API de detención permite detener los escenarios en ejecución.

Argumentos

La función de API de detención incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario que debe detenerse.

Name	Tipo	Descripción
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente. Cuando se establece este argumento como verdadero, la función de API no devolverá ningún valor hasta que el escenario se detenga. De lo contrario, la función devuelve un valor inmediatamente.
why_not_reason	out string	Contiene una descripción detallada del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje devuelto en el argumento `why_not_reason` para determinar la causa de error en la API.

Ejemplo

```
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;
}
```

Pausa de la replicación de datos: `suspend_replication()`

La API `suspend_replication` permite suspender las operaciones de replicación de un host.

Argumentos

La función de API `suspend_replication` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
<code>scenario_id</code>	uint	El ID de escenario que debe suspenderse.
<code>replica_index</code>	uint	El índice del host réplica en un escenario. Normalmente, el valor del índice de la réplica es 2.
<code>execute_sync</code>	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente. Cuando se establece este argumento como verdadero, la función de API no devolverá ningún valor hasta que la operación finalice. De lo contrario, la función devuelve un valor inmediatamente.
<code>mensaje</code>	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Finalización del trabajo: close_session()

La función de API close_session permite desconectarse del servicio de control. Después de iniciar sesión en el servicio de control, llame al argumento close_session para desconectarse del servicio de control.

Argumentos

La función de API close_session incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la sesión se cierra. Cuando el valor de retorno es falso, la sesión no se cierra.

Ejemplo

```

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;
}

```

Gestión de credenciales: add_credentials_ex()

La función de API add_credentials_ex permite agregar credenciales para los host.

Argumentos

La función de API add_credentials_ex incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API de función create_session (en la página 9).
scenario_id	uint	El ID de escenario al cual se desean agregar las credenciales.
host_name	cadena	Especifique la dirección IP del host. Ejemplo: 155.35.76.44.
port	uint	El número de puerto del motor. Normalmente, el número de puerto es 25000.
user_name	cadena	El nombre de usuario. Ejemplo: Administrador
password	cadena	La contraseña para el nombre de usuario. Ejemplo: Arcserve
domain_name	cadena	El nombre del dominio. Ejemplo: arcserve.com

Valores de retorno

El tipo de retorno es ulong. Un valor de retorno de cero indica que la API ha finalizado correctamente. Cuando el valor de retorno no es igual a cero, se produce un error en la API.

Ejemplo

```
public bool add_credentials_ex_example()
{
    try
    {
        uint scenario_id = _scenario_id;
        //Add credential for Master
        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);
        //Add credential for Replica
        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;
}
```

Integración con Arcserve Backup: add_bab_credentials()

La función de API add_bab_credentials permite agregar credenciales para acceder a Arcserve Backup.

Argumentos

La función de API add_bab_credentials incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario al cual se desean agregar las credenciales de Arcserve.
username	cadena	El nombre de usuario. Ejemplo: Administrador
password	cadena	La contraseña para el nombre de usuario. Ejemplo: Arcserve
async_id	out ulong	Cuando esta función de API se llame asincrónicamente, el argumento async_id contendrá un valor no nulo que permitirá a la función esperar a que la operación finalice y recupere el resultado.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente.

Ejemplo

```
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;
}
```

Actualización de las propiedades del escenario: update_scenario()

La función de API update_scenario permite actualizar datos del escenario durante el tiempo de ejecución.

Argumentos

La función de API update_scenario incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto tras llamar a la API create_session.
scenario_id	uint	El ID de escenario que desee actualizar.
scenario_data_str	ref string	Los datos sobre el escenario. Cuando la operación ha finalizado correctamente, se proporcionarán los nuevos datos del escenario.
why_not_reason	ref string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

Los valores de retorno proporcionan datos al escenario en el formato de xml.

Ejemplo

```

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);
        //do some changes for the scenario.
        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;
}

```

Eliminación de escenarios: remove_scenario()

La función de API remove_scenario permite eliminar escenarios del servicio de control.

Argumentos

La función de API remove_scenario incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El escenario ID que ejecutará.
arc_upd	bool	Este parámetro se aplica solamente a escenarios integrados de Arcserve Backup.
why_not-reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valor de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```

public bool remove_escenario_example()
{
    try
    {
        uint escenario_id = _escenario_id;
        bool arc_integrated = false;
        string why_not_reason = "";
        return get_mng().remove_escenario(_session_id, escenario_id, arc_integrated, out why_not_reason);
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
    return false;
}

```

Importación de escenarios: import_escenario()

La función de API import_escenario permite importar escenarios al servicio de control.

Argumentos

La función de API import_escenario incluye los argumentos descritos en la tabla siguiente:

Tipo	Name	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
group_id	uint	El ID de grupo de escenarios que desee importar. Si no está seguro del ID del grupo, establézcalo como 0xFFFFFFFF.
escenario_id	out uint	Recupera el ID de escenario cuando la API haya finalizado correctamente.
escenario_data	cadena	Los datos sobre el escenario. Normalmente, se recupera la cadena de datos del escenario desde un archivo de escenario. Los datos están en formato XML.
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

Los valores de retorno proporcionan datos al escenario en el formato de xml.

Ejemplo

```
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;
}
```

Sincronización en la ejecución de escenarios: synchronize()

La función de API de sincronización permite sincronizar los datos para los escenarios.

Argumentos

La función de API de sincronización incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create session (en la página 9).
scenario_id	uint	El ID de escenario que desee sincronizar.
sync_method	uint	El método de sincronización, que consiste en una de las opciones siguientes: 0: Sincronización de archivos 1: Sincronización de bloques 2: Sincronización de volúmenes (solo para el escenario de todo el sistema).
ignore_same_files	bool	Ignore los mismos archivos de tamaño/hora.

Name	Tipo	Descripción
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.
mensaje	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Reanudación de la replicación: resume_replication()

La función de API resume_replication permite reanudar las operaciones de replicación de un host.

Argumentos

La función de API suspend_replication incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto tras llamar a la API create_session.
scenario_id	uint	El ID de escenario para el cual se desea reanudar la replicación.
replica_index	uint	El índice del host réplica en un escenario. Normalmente, el valor del índice de la réplica es 2. Este es el host para el cual se suspenderá la replicación. Por ejemplo, los cambios de datos se agregan a un spool sin copiar los datos a un disco hasta reanudar la operación de replicación.
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.
mensaje	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Adición de los marcadores de rebobinado: set_rewind_bookmark()

La función de API set_rewind_bookmark permite establecer marcadores para escenarios.

Argumentos

La función de API set_rewind_bookmark incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
scenario_id	cadena	El ID de escenario para el cual desee establecer marcadores.
host_index	uint	Always = 1.
bookmark_msg	cadena	El nombre del marcador.
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

API de gestión de escenarios de alta disponibilidad

En las secciones siguientes se describen las API que permiten gestionar los escenarios de alta disponibilidad.

Esta sección incluye los siguientes temas:

[Activación de la recuperación asegurada: start_ar\(\)](#) (en la página 36)

[Desactivación de la pulsación en los escenarios de alta disponibilidad: stop_is_alive\(\)](#) (en la página 37)

[Conmutación de los escenarios de alta disponibilidad: switchover\(\)](#) (en la página 38)

[Activación de las pulsaciones en los escenarios de alta disponibilidad: start_is_alive\(\)](#) (en la página 42)

Activación de la recuperación asegurada: start_ar()

La función de API start_ar permite realizar una operación de recuperación asegurada (AR) de un escenario. Cuando se realiza una AR automática, no es necesario llamar a otras API para detener esta acción. La recuperación asegurada se detendrá después de finalizar la misma operación. Cuando se realiza una recuperación asegurada manual, llame a la API [resume application](#) (en la página 33) para detener esta operación.

Argumentos

La función de API start_ar incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario para el cual desea realizar recuperación asegurada.
replica_index	uint	El índice del host réplica en un escenario. Normalmente, el valor del índice de la réplica es 2. Este es el host para el cual se suspenderá la replicación.
auto_ar	bool	Ejecute la recuperación asegurada automáticamente o manualmente. <ul style="list-style-type: none">■ verdadero: automáticamente■ False-manually
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.
mensaje	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```

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;
}

```

Desactivación de la pulsación en los escenarios de alta disponibilidad: stop_is_alive()

La monitorización es una señal electrónica que envían los servidores de réplica a los servidores máster para identificar el estado del nodo. Mientras se ejecutan los escenarios de alta disponibilidad, el servidor réplica envía periódicamente una señal electrónica (ping) al servidor máster. De forma predeterminada, la frecuencia del ping es 30 segundos. Se puede activar un evento de conmutación cuando el servidor réplica no pueda hacer ping en el máster después de un período predeterminado de tiempo transcurrido (el valor predeterminado es 300 segundos).

La función de API stop_is_alive permite suspender la comprobación de monitorización.

Argumentos

La función de API stop_is_alive incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario para el cual desea suspender la comprobación de monitorización.
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.

Name	Tipo	Descripción
err_message	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Conmutación de los escenarios de alta disponibilidad: switchover()

La función de API switchover permite realizar operaciones de conmutación.

Con escenarios de alta disponibilidad y de todo el sistema, se pueden realizar operaciones de conmutación a cualquier servidor réplica. Cuando desee cambiar a servidores réplica que no dispongan de ninguna conmutación por error, llame a la API execute_action antes de llamar a la API de conmutación.

Nota: La API execute_action se describe en los ejemplos.

Argumentos

La función de API switchover incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario para el cual desea realizar las operaciones de conmutación.
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.
run_reverse_scenario	bool	Ejecute o no ejecute el escenario inverso después de que se produzca la operación de conmutación.
err_message	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplos

Ejemplo 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;
}
```

Ejemplo 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;
}
```

Ejemplo 3

De forma predeterminada, Replication y High Availability realiza operaciones de conmutación en la conmutación por error predeterminada y en los servidores réplica. Con los escenarios de alta disponibilidad y de todo el sistema, se puede cambiar a los servidores réplica que no dispongan de ninguna conmutación por error. Sin embargo, cuando desee cambiar a los servidores que no dispongan de ninguna conmutación por error mediante la API de conmutación, llame a la API `execute_action` antes de llamar a la API de conmutación, tal y como aparece en el ejemplo siguiente:

```
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;
}
```

Activación de las pulsaciones en los escenarios de alta disponibilidad: start_is_alive()

La monitorización es una señal electrónica que envían los servidores de réplica a los servidores máster para identificar el estado del nodo. Mientras se ejecutan los escenarios de alta disponibilidad, el servidor réplica envía periódicamente una señal electrónica (ping) al servidor máster. De forma predeterminada, la frecuencia del ping es 30 segundos. Se puede activar un evento de conmutación cuando el servidor réplica no pueda hacer ping en el máster después de un período predeterminado de tiempo transcurrido (el valor predeterminado es 300 segundos).

La función de API start_is_alive permite reanudar la comprobación de monitorización.

Argumentos

La función de API start_is_alive incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
scenario_id	uint	El ID de escenario para el cual desea iniciar la comprobación de monitorización.
execute_sync	bool	Regula si debe llamarse a esta API sincrónica o asincrónicamente.
err_message	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

API de gestión de instantáneas de VSS

En las secciones siguientes se describen API que permiten gestionar instantáneas de VSS.

Esta sección incluye los siguientes temas:

[Montaje de la instantánea VSS en los servidores réplica: mount_snapshot\(\)](#) (en la página 44)

[Desmontaje de instantáneas VSS desde los servidores réplica: unmount_snapshot\(\)](#) (en la página 45)

[Eliminación de instantáneas VSS desde los servidores réplica: delete_snapshot\(\)](#) (en la página 46)

[Obtención de listas de instantáneas de VSS desde los servidores réplica: get_snapshot_list\(\)](#) (en la página 48)

Montaje de la instantánea VSS en los servidores réplica: mount_snapshot()

La función de API mount_snapshot permite montar instantáneas VSS a una carpeta específica en un servidor réplica.

Argumentos

La función de API mount_snapshot incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
host_name	cadena	El nombre de host del motor.
ip_string	cadena	La dirección IP de host_name.
host_port	ushort	El número de puerto del motor. Normalmente, el número de puerto es 25000.
mount_path	cadena	La carpeta donde desee montar la instantánea.
snapshot_id	cadena	El ID de instantánea de VSS.
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```

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;
}

```

Desmontaje de instantáneas VSS desde los servidores réplica: unmount_snapshot()

La función de API unmount_snapshot permite desmontar instantáneas VSS de una carpeta.

Argumentos

La función de API unmount_snapshot incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
host_name	cadena	El nombre de host del motor.
ip_string	cadena	La dirección IP de host_name.
host_port	ushort	El número de puerto del motor. Normalmente, el número de puerto es 25000.
snapshot_id	cadena	El ID de instantánea de VSS.
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Eliminación de instantáneas VSS desde los servidores réplica: delete_snapshot()

La función de API delete_snapshot permite suprimir instantáneas de VSS desde los servidores réplica.

Argumentos

La función de API delete_snapshot incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
host_name	cadena	El nombre de host del motor.
ip_string	cadena	La dirección IP de host_name.

Name	Tipo	Descripción
host_port	ushort	El número de puerto del motor. Normalmente, el número de puerto es 25000.
snapshot_id	cadena	El ID de instantánea de VSS.
why_not_reason	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Obtención de listas de instantáneas de VSS desde los servidores réplica: `get_snapshot_list()`

La función de API `get_snapshot_list` permite obtener una lista de instantáneas VSS desde un host.

Argumentos

La función de API `get_snapshot_list` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
<code>host_name</code>	cadena	El nombre de host del motor.
<code>ip_string</code>	cadena	La dirección IP de <code>host_name</code> .
<code>host_port</code>	ushort	El número de puerto del motor. Normalmente, el número de puerto es 25000.
<code>snapshot_list</code>	out string	La lista de instantáneas de volumen.
<code>why_not_reason</code>	out string	Contiene el motivo del error cuando se produce un error en esta API.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno es verdadero, la API finalizará correctamente. Cuando el valor de retorno es falso, la API no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplo

```
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;
}
```

Recopilación de las API de estadísticas

En las secciones siguientes se describen las API que permiten gestionar el proceso de recopilación de estadísticas sobre escenarios.

Esta sección incluye los siguientes temas:

[Obtención de la estadística del escenario extendido: `get_data_ex\(\)`](#) (en la página 49)

[Obtención de la estadística de escenario: `get_scenario_data\(\)`](#) (en la página 57)

Obtención de la estadística del escenario extendido: `get_data_ex()`

La función de API `get_data_ex` permite obtener toda la información sobre el escenario, como a continuación:

- Eventos del escenario
- Estados del escenario
- Estadísticas de sincronización y replicación

El parámetro `manager_data` es una cadena en el formato xml. Los datos contienen toda la información sobre el escenario. Por ejemplo, el estado del escenario (en ejecución, detenido, etc.), eventos, estadísticas de escenario, entre otros. La cadena en formato xml se parece a lo siguiente:

```
<?xml version="1.0"?>
<manager_data>
  - <scenarios>
    + <scenario signature="4636778060728034734" ha_type="Forward" is_arcsolve_integrated="False" is_cdp="Fa
      id="1094498606">
    + <scenario signature="4334615870148788711" is_arcsolve_integrated="False" is_cdp="False" is_ass_rec="Fal
    + <scenario signature="14684688067413199200" ha_type="Forward" is_arcsolve_integrated="False" is_cdp="f
    + <scenario signature="15270013466011305316" ha_type="Forward" is_arcsolve_integrated="False" is_cdp="f
      id="43557253">
    + <scenario signature="5773759741404806146" is_arcsolve_integrated="False" is_cdp="False" is_ass_rec="Fal
    + <scenario signature="7020398949829650879" is_arcsolve_integrated="False" is_cdp="False" is_ass_rec="Fal
    + <scenario signature="7044671085026122361" is_arcsolve_integrated="False" is_cdp="False" is_ass_rec="Tru
    + <scenario signature="7920379657132428156" ha_type="Forward" is_arcsolve_integrated="False" is_cdp="Fa
      id="3423940998">
    </scenario>
  </scenarios>
  + <scenario_groups>
</manager_data>
```

Las secciones siguientes demuestran cómo utilizar la cadena de xml.

Argumentos

La API `get_data_ex` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
<code>session_id</code>	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
<code>scenarios_with_statistics</code>	uint	La matriz de los ID de escenario. Recupera la estadística de los escenarios.
<code>last_update_time</code>	ulong	Última actualización de la marca de tiempo.
<code>request_flag</code>	uint	Solicita el tipo de datos. Los valores pueden ser los siguientes: 1: datos del escenario 2: datos cdp [no se utilizan ahora] 4: datos de gestión del host 8: datos de instantánea 15: todos los datos anteriores
<code>manager_data</code>	out string	Devuelve los datos del escenario en el formato xml.

Valores de retorno

El tipo de valor de retorno es booleano. Cuando el valor de retorno sea verdadero, el comando finalizará correctamente. Cuando el valor de retorno sea falso, el comando no finaliza correctamente. Si el valor de retorno es falso, revise el mensaje para determinar el motivo por el cual la API es errónea.

Ejemplos

Ejemplo 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;
}
```

Ejemplo 2:

Esta función analiza los datos del búfer de XML (manager_data) que ha devuelto la función get_data_ex(). El ejemplo siguiente describe cómo obtener el estado del escenario cuando se ejecuta o se detiene:

```
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.ToUInt32(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;  
}
```

Ejemplo 3:

Esta función analiza los datos del búfer de XML (manager_data) que ha devuelto la función get_data_ex(). El ejemplo siguiente describe cómo obtener todos los eventos para el escenario:

```
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.ToUInt32(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));
            }
        }
    }
}
```

Ejemplo 4:

Se debe definir el parámetro `scenarios_with_statistics` para obtener la estadística de sincronización y replicación de un escenario. El parámetro es una matriz. Para obtener la estadística para más de un escenario, se deben agregar los ID de los escenarios a la matriz.

`get_data_ex` obtiene la estadística siguiente:

Nota: Las siguientes pantallas muestran el proceso de transferencia y sincronización de la información de las estadísticas.

Transferred bytes to Replicas:

Host	Total Sent Data	Current File Name	Data To be Sent	Transmission Speed	Current Progress
symibrva	4.98GB	D:/Master Company/MasterCompany.ECD	3.78GB	1.3Mbps	19.3 %

Last synchronization statistics: Block Data synchronization

Synchronization Progress:

- **sym-sql ->symibrva**

C:/

State	Number of Files	Total Size	Compare Progress	Data To be Sent	Send Progress	Starting Time	Finish Time
Finished	68481	18.42GB	100.0 %	9.28GB	100.0 %	3/25/2012 9:47:49 AM	3/25/2012 7:34:07 PM

Código

Esta función analiza los datos del búfer de XML (`manager_data`) que ha devuelto la función `get_data_ex()`. El código siguiente muestra cómo obtener la estadística de transferencia y sincronización. (Consulte las pantallas anteriores.)

```
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.ToUInt32(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));
            }
        }
    }
}
```

Ejemplo 5:

El código siguiente muestra cómo recuperar la estadística de sincronización para los escenarios.

```
ArrayList sync_statistics = new ArrayList();
    get_escenario_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.
            ulong speed = trans.transmission_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
            }
        }
    }
}
```

Obtención de la estadística de escenario: `get_escenario_data()`

La función de API `get_escenario_data` permite obtener los ID de escenario.

Argumentos

La función de API `get_escenario_data` incluye los argumentos descritos en la tabla siguiente:

Name	Tipo	Descripción
session_id	uint	El ID de sesión que se ha devuelto llamando a la API create_session (en la página 9).
escenario_id	uint	El ID de escenario.

Valores de retorno

Los valores de retorno proporcionan datos al escenario.

Ejemplo

```
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;
}
```

Capítulo 3: Index
