

IBM InfoSphere DataStage  
Versión 11 Release 3

*Guía del programador*





IBM InfoSphere DataStage  
Versión 11 Release 3

*Guía del programador*



**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado “Avisos y marcas registradas” en la página 149.

# Contenido

## Capítulo 1. Interfaz de línea de mandatos . . . . . 1

Mandatos para el control de trabajos de InfoSphere DataStage . . . . .	1
Mantener la seguridad de credenciales y valores de parámetro en mandatos <b>dsjob</b> . . . . .	1
La cláusula de inicio de sesión . . . . .	3
Iniciar un trabajo . . . . .	4
Detención de un trabajo . . . . .	6
Listado de proyectos, trabajos, etapas, enlaces, parámetros y colas . . . . .	7
Establecimiento de un alias para un trabajo . . . . .	9
Recuperación de información . . . . .	9
Acceso a los archivos de registro . . . . .	11
Generación de un informe . . . . .	14
Mandatos para administrar proyectos . . . . .	14
Protección de credenciales en los mandatos <b>dsadmin</b> . . . . .	15
La cláusula de inicio de sesión . . . . .	16
Creación de un proyecto . . . . .	17
Supresión de un proyecto. . . . .	18
Protección de un proyecto . . . . .	18
Habilitación/inhabilitación de la depuración automática de los archivos de registro . . . . .	19
Habilitación/inhabilitación de la visualización de OSH. . . . .	19
Habilitación/inhabilitación de la propagación de columnas en tiempo de ejecución . . . . .	20
Habilitación/Inhabilitación de la administración de trabajos desde el cliente del Director . . . . .	20
Habilitación/inhabilitación de la generación de informe XML . . . . .	20
Habilitación/inhabilitación de las propiedades avanzadas de tiempo de ejecución. . . . .	20
Establecimiento del directorio base . . . . .	21
Establecimiento de la plantilla del directorio de despliegue. . . . .	21
Adición de una variable de entorno . . . . .	21
Supresión de una variable de entorno . . . . .	22
Establecimiento del valor de una variable de entorno. . . . .	22
Listado de proyectos . . . . .	23
Listado de propiedades . . . . .	23
Listado de variables de entorno . . . . .	23
Mandatos para la importación desde archivos .dsx . . . . .	23
Importar objetos desde un archivo .dsx . . . . .	23
Listado del contenido de un archivo .dsx . . . . .	27
Mandatos para la comprobación y reparación de proyectos . . . . .	27
Parámetros de autenticación para el mandato <b>SyncProject</b> . . . . .	28
Comprobación de incoherencias en los proyectos . . . . .	29
Reparar proyectos incoherentes. . . . .	31
Comprobación y reparación interactivas de proyectos . . . . .	33

Reconstruir un proyecto . . . . .	34
Creación de una copia de seguridad de un proyecto . . . . .	35
Restaurar un proyecto . . . . .	36

## Capítulo 2. InfoSphere DataStage Development Kit (Interfaces de control de trabajos) . . . . . 37

InfoSphere DataStage Development Kit . . . . .	37
El archivo de cabecera dsapi.h . . . . .	37
Estructuras de datos, datos de resultados y hebras . . . . .	38
Grabación de programas de la API de InfoSphere DataStage . . . . .	38
Creación de una aplicación de la API de InfoSphere DataStage . . . . .	39
Redistribución de aplicaciones . . . . .	39
Funciones de la API . . . . .	40
DSAddEnvVar . . . . .	42
DSAddProject . . . . .	43
DSCloseJob . . . . .	44
DSCloseProject . . . . .	44
DSDeleteEnvVar . . . . .	45
DSDeleteProject . . . . .	45
DSFindFirstLogEntry . . . . .	46
DSFindNextLogEntry . . . . .	48
DSGetCustInfo . . . . .	49
DSGetJobInfo . . . . .	50
DSGetLastError . . . . .	51
DSGetLastErrorMsg . . . . .	52
DSGetLinkInfo . . . . .	53
DSGetLogEntry . . . . .	54
DSGetLogEntryFull . . . . .	55
DSGetLogEventIds . . . . .	56
DSGetNewestLogId . . . . .	57
DSGetParamInfo . . . . .	58
DSGetProjectInfo . . . . .	59
DSGetProjectList . . . . .	60
DSGetReposInfo . . . . .	60
DSGetReposUsage . . . . .	62
DSGetStageInfo . . . . .	63
DSGetVarInfo . . . . .	65
DSLstEnvVars . . . . .	66
DSLstProjectProperties . . . . .	66
DSLockJob . . . . .	68
DSLogEvent . . . . .	68
DSMakeJobReport . . . . .	69
DSOpenJob . . . . .	70
DSOpenProject . . . . .	71
DSRunJob . . . . .	72
DSSetEnvVar . . . . .	73
DSSetGenerateOpMetaData . . . . .	74
DSSetJobLimit . . . . .	74
DSSetParam . . . . .	75
DSSetProjectProperty . . . . .	76

DSSetServerParams . . . . .	78	DSPrepareJob . . . . .	124
DSStopJob . . . . .	78	DSRunJob . . . . .	125
DSUnlockJob . . . . .	79	DSSendMail . . . . .	126
DSWaitForJob . . . . .	80	DSSetDisableJobHandler . . . . .	127
Estructuras de datos . . . . .	80	DSSetDisableProjectHandler . . . . .	127
DSCUSTINFO . . . . .	81	DSSetGenerateOpMetaData . . . . .	128
DSJOBINFO . . . . .	82	DSSetJobLimit . . . . .	128
DSLINKINFO . . . . .	85	DSSetJobQueue . . . . .	129
DSLOGDETAIL . . . . .	86	DSSetParam . . . . .	129
DSLOGDETAILFULL . . . . .	87	DSSetUserStatus . . . . .	130
DSLOGEVENT . . . . .	88	DSStopJob . . . . .	131
DSPARAM . . . . .	88	DSTransformError . . . . .	131
DSPARAMINFO . . . . .	90	DSTranslateCode . . . . .	132
DSPROJECTINFO . . . . .	91	DSWaitForFile . . . . .	132
DSREPOSINFO . . . . .	92	DSWaitForJob . . . . .	133
DSREPOSUSAGE . . . . .	93	Macros de estado de trabajo . . . . .	134
DSSTAGEINFO . . . . .	93		
DSVARINFO . . . . .	95		
Códigos de error . . . . .	95	<b>Capítulo 3. Generación de un informe</b>	
Interfaz de BASIC de InfoSphere DataStage . . . . .	102	<b>XML . . . . .</b>	<b>137</b>
DSAttachJob . . . . .	104	<b>Apéndice A. Accesibilidad de los</b>	
DSCheckRoutine . . . . .	104	<b>productos . . . . .</b>	<b>139</b>
DSDetachJob . . . . .	105	<b>Apéndice B. Lectura de la sintaxis de</b>	
DSExecute . . . . .	105	<b>la línea de mandatos . . . . .</b>	<b>141</b>
DSGetCustInfo . . . . .	106	<b>Apéndice C. Cómo ponerse en</b>	
DSGetJobInfo . . . . .	107	<b>contacto con IBM . . . . .</b>	<b>143</b>
DSGetLinkInfo . . . . .	109	<b>Apéndice D. Acceso a la</b>	
DSGetLogEntry . . . . .	111	<b>documentación del producto. . . . .</b>	<b>145</b>
DSGetLogEntryFull . . . . .	112	<b>Apéndice E. Cómo aportar</b>	
DSGetLogEventIds . . . . .	113	<b>comentarios sobre la documentación</b>	
DSGetLogSummary . . . . .	114	<b>del producto. . . . .</b>	<b>147</b>
DSGetNewestLogId . . . . .	115	<b>Avisos y marcas registradas . . . . .</b>	<b>149</b>
DSGetParamInfo . . . . .	115	<b>Índice. . . . .</b>	<b>155</b>
DSGetProjectInfo . . . . .	117		
DSGetStageInfo . . . . .	118		
DSGetVarInfo . . . . .	120		
DSLogEvent . . . . .	120		
DSLogFatal . . . . .	121		
DSLogInfo . . . . .	121		
DSLogToController . . . . .	122		
DSLogWarn . . . . .	122		
DSMakeJobReport . . . . .	123		
DSMakeMsg . . . . .	124		

---

## Capítulo 1. Interfaz de línea de mandatos

La CLI de InfoSphere DataStage consta de cuatro grupos de mandatos, uno para ejecutar trabajos, uno para administrar proyectos, uno para importar objetos y uno para comprobar y reparar objetos.

---

### Mandatos para el control de trabajos de InfoSphere DataStage

Puede iniciar y detener trabajos y recuperar información sobre ejecuciones de trabajo mediante el mandato `dsjob`.

Las opciones de mandato utilizadas con el mandato `dsjob` proporcionan acceso a la misma funcionalidad que las funciones de la API de InfoSphere DataStage descritas en “Funciones de la API” en la página 40 o las funciones de BASIC descritas en “Interfaz de BASIC de InfoSphere DataStage” en la página 102.

Existe un único mandato, `dsjob`, con una amplia gama de opciones. Estas opciones se describen en los temas siguientes:

- La cláusula de inicio de sesión
- Inicio de un trabajo
- Detención de un trabajo
- Listado de proyectos, trabajos, etapas, enlaces y parámetros
- Establecimiento de un alias para un trabajo
- Recuperación de información
- Acceso a los archivos de registro
- Generación de un informe

Toda la salida del mandato `dsjob` se realiza en texto sin formato y sin cabeceras de columna en las listas o cualquier otra clase de descripción. Esto permite que se utilice el mandato en scripts de shell o de proceso por lotes sin proceso adicional.

La CLI de InfoSphere DataStage devuelve un código de finalización de 0 al sistema operativo si la ejecución ha sido satisfactoria o uno de los códigos de error de la API de InfoSphere DataStage si es anómala. Consulte “Códigos de error” en la página 95. El código de error también se imprime en la secuencia de errores estándar en todos los casos. En servidores UNIX, se devuelve un código de 255 si el código de error es negativo o superior a 254, para ver el código de retorno "real" en estos casos, capture y procese la secuencia de errores estándar.

Tenga en cuenta que, en servidores UNIX, `DSJOB` tiene una dependencia de tiempo de ejecución de la biblioteca compartida `libvmdsapi.so`.

### Mantener la seguridad de credenciales y valores de parámetro en mandatos `dsjob`

Puede cifrar los datos y almacenar los valores cifrados en archivos que se utilizarán en los mandatos **`dsjob`**. Para especificar credenciales en la línea de mandatos, opcionalmente puede utilizar la solicitud de credenciales para ocultar la contraseña en la ventana de mandatos.

## Acerca de esta tarea

Puede utilizar credenciales cifradas y valores de archivo de parámetro cifrados en los mandatos **dsjob** de forma que pueda evitar especificar datos visibles en la pantalla al ejecutar mandatos desde la línea de mandatos. Consulte la documentación del *Mandato encrypt* de IBM® InfoSphere Information Server - Guía de administración para obtener información sobre este mandato.

## Procedimiento

1. Cifre su información mediante la ejecución del mandato **encrypt**, y copie y guarde los valores cifrados en un archivo:
  - Si cifra credenciales, cree el archivo de credenciales (\*.txt) y almacene de forma segura el archivo. Consulte el tema del archivo de credenciales para ver las restricciones y contenido de ejemplo del archivo de credenciales.
  - Si cifra valores de parámetro de trabajo, almacene los valores cifrados en el archivo de parámetros del trabajo correspondiente copiando la salida cifrada y pegando el valor en el archivo de parámetros del trabajo. Por ejemplo:  
*nombre\_parámetro\_trabajo=valor\_parámetro\_trabajo\_cifrado*

Puede crear alias para los valores de parámetro del trabajo. No puede cifrar nombres de parámetro del trabajo.

Guarde el archivo.

2. Ejecute el mandato **dsjob**. Puede especificar las credenciales con seguridad mediante uno de los métodos siguientes:
  - **Mediante el archivo de credenciales.** Si ejecuta un mandato **dsjob** que requiere sus credenciales de usuario, ejecute el mandato con el parámetro **-authfile** y especifique la vía de acceso completa del archivo de credenciales que desea utilizar. A continuación se muestra una sintaxis de ejemplo para la utilización del parámetro **-authfile** en un mandato **dsjob** que utiliza un archivo de parámetros con datos cifrados:
  - **Solicitud de credenciales.** Si desea especificar los datos de credenciales mediante la línea de mandatos, especifique sólo el **-url** o el parámetro **-domain** y el parámetro **-server**. Se le solicitará el nombre de usuario y la contraseña. (La contraseña queda oculta en la ventana de mandatos.) Si en el mandato incluye el parámetro **-url** o **-domain** y los parámetros **-server** y **-user**, se le solicitará la contraseña. A continuación se muestra un mandato e interacción de ejemplo:

```
C:\IBM\InformationServer\Clients\Classic>dsjob
-domain [2002:920:c000:217:9:32:217:32]:9443 -server ServidorRemoto
-ljobs newTest
```

Especifique el nombre de usuario:admin

Especifique la contraseña:

```
Job_ODBC
Job_UDT5
Job_UDT6
Job_Universe
PXJob_DC
Sequence_ODBC
Sequence_UDT5
Sequence_UDT6
Sequence_Universe
```

Código de estado = 0



**Nota:** La contraseña queda oculta en la ventana de mandatos.

## La cláusula de inicio de sesión

De manera predeterminada, la CLI de InfoSphere DataStage se conecta al motor en el sistema local utilizando el nombre de usuario y la contraseña del usuario que ejecuta el mandato.

Para los mandatos **dsjob** y **dsadmin**, puede especificar un dominio, motor, nombre de usuario o contraseña diferente utilizando la cláusula de inicio de sesión, que es equivalente a la función **DSSetServerParams** de la API. Su sintaxis es la siguiente:

```
[-url URLdominio |  
  -domain nombre_dominio ][ -user nombre_usuario ][ -password contraseña ]  
  [ -server nombre_motor ]
```

o bien, sólo para el mandato **dsjob**:

```
-domain NONE -user nombre_usuario -password contraseña -server nombre_motor
```

*URLdominio* especifica un URL de formato completo para que el dominio se conecte a él. El URL incluye información de protocolo, host y puerto para el dominio en este formato: `https://domain:port`. El puerto adopta el valor predeterminado de 9443 si no está especificado.

*nombre\_dominio* especifica el dominio en el que se iniciará sesión. Para **dsjob**, puede establecer **-domain NONE** para iniciar sesión en el motor en lugar de en el dominio. En este caso, el nombre de usuario y la contraseña son para el motor, no para el dominio.

*nombre\_motor* especifica un motor distinto en el que se iniciará sesión.

*nombre\_usuario* especifica un nombre de usuario distinto para utilizar al iniciar la sesión.

*contraseña* especifica una contraseña distinta para utilizar al iniciar la sesión.

Los nombres de usuario cifrados y las contraseñas cifradas no están soportados en la línea de mandatos, utilice un archivo de credenciales si desea utilizar valores cifrados.

Si no desea escribir los credenciales en la línea de mandatos y no desea utilizar un archivo de credenciales, especifique únicamente el parámetro **-url** o **-domain** y el parámetro **-server** y se le solicitará el nombre de usuario y la contraseña. (La contraseña queda oculta cuando la especifica en la línea de mandatos.) Si en el mandato incluye el parámetro **-url** o **-domain** y los parámetros **-server** y **-user**, se le solicitará la contraseña. A continuación se muestra un mandato e interacción de ejemplo:

```
C:\IBM\InformationServer\Clients\Classic>dsjob  
-domain [2002:920:c000:217:9:32:217:32]:9443 -server ServidorRemoto  
-ljobs newTest  
Especifique el nombre de usuario:admin  
Especifique la contraseña:  
Job_ODBC
```

Código de estado = 0

Para sistemas que no utilizan los puertos predeterminados, para conectar a un proyecto que está en un servidor local, especifique la opción **-server** con sólo el

número de puerto y no especifique el nombre de servidor. No necesita especificar el nombre de usuario ni la contraseña. Por ejemplo:

```
dsjob -server :31539 -lprojects
```

Para un inicio de sesión más seguro para los mandatos **dsjob** y **dsadmin**, puede utilizar un archivo de credenciales que contenga los datos cifrados:

```
-authfile nombre_archivo_credenciales
```

*nombre\_archivo\_credenciales* es la vía de acceso completa y el nombre del archivo que contiene los detalles de la conexión. Este archivo da soporte a datos cifrados y no cifrados. Consulte El archivo de credenciales para ver detalles y contenido de ejemplo del archivo.

Opcionalmente, podría especificar los detalles no cifrados en un archivo de credenciales mediante la sintaxis siguiente:

```
-file nombre_archivo_credenciales nombre_dominio nombre_motor
```

**Nota:** El archivo de credenciales **-file** no puede contener datos cifrados.

Para el mandato **dsjob**, también puede utilizar el mandato:

```
-file nombre_archivo_credenciales NONE nombre_motor
```

*nombre\_dominio* especifica el dominio para el que el archivo contiene detalles de inicio de sesión. Para **dsjob**, puede establecer NONE para iniciar la sesión en el motor en lugar de hacerlo en el dominio. En este caso, el nombre de usuario y la contraseña son para el motor, no para el dominio.

*nombre\_motor* especifica el motor para el que el archivo contiene detalles de inicio de sesión.

*nombre\_archivo\_credenciales* es la vía de acceso completa y el nombre del archivo que contiene los detalles de la conexión. Este archivo da soporte sólo a datos no cifrados. El archivo debe contener la siguiente información si se inicia sesión en el dominio:

*nombre\_dominio, nombre\_motor, nombre\_usuario, contraseña*

El archivo debe contener la siguiente información si se inicia sesión en el motor:

*nombre\_motor, nombre\_usuario, contraseña*

La inclusión de la cláusula de inicio de sesión en los mandatos puede exponer el nombre de usuario y la contraseña. Es mejor utilizar la opción **-authfile** y mantener la información de inicio de sesión cifrada, o permitir que el sistema le solicite la contraseña.

## Iniciar un trabajo

Puede iniciar, detener, validar y restablecer trabajos utilizando la opción **-run**.

```
dsjob -run  
[ -mode [ NORMAL | RESET | VALIDATE | RESTART ] ]  
[ -param nombre=valor ]  
[ -paramfile nombre_archivo ]  
[ -warn n ]  
[ -rows n ]  
[ -wait ]  
[ -stop ]
```

```
[ -jobstatus ]
[ -userstatus ]
[ -local ]
[ -opmetadata [ TRUE | FALSE ] ]
[ -disableprjhandler ]
[ -disablejobhandler ]
[ -queue ]
[ -useid ] proyecto trabajo | id_trabajo
```

#### **-mode**

Especifica el tipo de ejecución de trabajo. NORMAL inicia una ejecución de trabajo. RESET restaura el trabajo. VALIDATE valida el trabajo. RESTART reanuda una secuencia de trabajos reinicializable a partir del último punto de comprobación utilizando los valores de parámetros del trabajo original. Si no se especifica **-mode**, se inicia una ejecución de trabajo normal.

#### **-param**

Especifica un valor de parámetro para pasar al trabajo. El valor está en el formato *nombre=valor*, donde *nombre* es el nombre de parámetro y *valor* es el valor que se va a establecer. Si utiliza este valor para pasar un valor de una variable de entorno para un trabajo (y para los trabajos paralelos), debe utilizar comillas simples en la variable de entorno y en el valor. Por ejemplo, **-param** '\$APT\_CONFIG\_FILE=test.apt'. De lo contrario, se utilizará el valor actual de la variable de entorno.

#### **-paramfile**

Especifica un archivo que contiene los valores de parámetro para pasar al trabajo. Los valores de parámetro pueden tener el mismo formato que **-param** o bien los valores pueden cifrarse y almacenarse en un archivo de parámetros tal como se describe en “Mantener la seguridad de credenciales y valores de parámetro en mandatos **dsjob**” en la página 1.

#### **-warn *n***

Establece límites de aviso en el valor especificado por *n* (equivalente a la función **DSSetJobLimit** que se utiliza con DSJ\_LIMITWARN especificado como el parámetro *LimitType*).

#### **-rows *n***

Establece límites de fila en el valor especificado por *n* (equivalente a la función **DSSetJobLimit** que se utiliza con DSJ\_LIMITROWS especificado como el parámetro *LimitType*).

#### **-wait**

Espera a que se complete el trabajo (equivalente a la función **DSSWaitForJob**).

#### **-stop**

Detiene un trabajo en ejecución (equivalente a la función **DSSstopJob**).

#### **-jobstatus**

Espera a que se complete el trabajo y a continuación devuelve un código de salida derivado del estado del trabajo.

#### **-userstatus**

Espera a que se complete el trabajo y a continuación devuelve un código de salida derivado del estado del usuario si dicho estado está definido. El estado de usuario es una serie y se convierte a un código de salida de entero. El código de salida 0 indica que el trabajo se ha completado sin ningún error, pero que la serie de estado de usuario no se ha podido convertir. Si un trabajo devuelve un valor de estado de usuario negativo, se interpreta como error.

#### **-local**

Utilice esta opción al ejecutar un trabajo desde un script de shell en un sistema

UNIX. Siempre que se ejecute el script en el directorio del proyecto, el trabajo adoptará los valores para cualquier variable de entorno que esté establecida en el script y cualquier valor específico del entorno de usuario.

**-opmetadata**

Utilice esta opción para que el trabajo genere metadatos operativos cada vez que se ejecute. Si especifica TRUE, se generan metadatos operativos, independientemente de cuál sea el valor predeterminado del proyecto. Si especifica FALSE, el trabajo no generará metadatos operativos, independientemente de cuál sea el valor predeterminado del proyecto.

**-disableprjhandler**

Utilice esta opción para inhabilitar cualquier manejador de mensajes de error que esté establecido para todo el proyecto.

**-disablejobhandler**

Utilice esta opción para inhabilitar cualquier manejador de mensajes de error que esté establecido para este trabajo.

**-useid**

Especifique esta opción si tiene la intención de utilizar un alias de trabajo (jobid) en vez de un nombre de trabajo (job) para identificar el trabajo.

**-queue**

El nombre de la cola de gestión de carga de trabajo a la que se ha sometido el trabajo. Si no especifica una cola, se utiliza la cola predeterminada del proyecto. Si el trabajo es un trabajo de secuencia, se someterán a esta cola todos los trabajos del trabajo de secuencia.

*proyecto*

El nombre del proyecto que contiene el trabajo.

*trabajo*

El nombre del trabajo. Para ejecutar una invocación de trabajo, utilice el formato *job.invocation\_id*.

*id\_trabajo*

Alias del trabajo establecido mediante el mandato **dsjob -jobid**.

## Detención de un trabajo

Puede detener un trabajo utilizando la opción **-stop**.

**dsjob -stop** [-useid] *proyecto trabajo|id\_trabajo*

**-stop** termina un trabajo en ejecución (equivalente a la función **DSSStopJob**).

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (id\_trabajo) en lugar de un nombre de trabajo (trabajo) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el trabajo.

*trabajo* es el nombre del trabajo. Para detener una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

## Listado de proyectos, trabajos, etapas, enlaces, parámetros y colas

Puede listar proyectos, trabajos, etapas, enlaces, parámetros de trabajo y colas de gestión de carga de trabajo utilizando el mandato **dsjob**.

Las distintas versiones de la sintaxis se describen en las secciones siguientes.

### Listado de proyectos

La siguiente sintaxis muestra una lista de proyectos conocidos en el servidor:

```
dsjob -lprojects
```

Esta sintaxis es equivalente a la función **DSGetProjectList**.

### Listado de trabajos

La sintaxis siguiente muestra una lista de todos los trabajos en el proyecto especificado:

```
dsjob -ljobs proyecto
```

*proyecto* es el nombre del proyecto que contiene los trabajos que se listarán.

Esta sintaxis es equivalente a la función **DSGetProjectInfo**.

### Listado de trabajos con estados de trabajo concretos

La sintaxis siguiente muestra una lista de todos los trabajos en el proyecto especificado con valores de estados de trabajo concretos:

```
dsjob -ljobs [-status lista_estados] proyecto
```

*proyecto* es el nombre del proyecto que contiene los trabajos que se listarán.

*lista\_estados* es una lista de valores de estados de trabajos, tal como se definen en el archivo `dsapi.h`.

El siguiente mandato lista todos los trabajos del proyecto `dstage1` con los estados de `DSJS_CRASHED` o `DSJS_STOPPED`:

```
dsjob -ljobs -status 96/97 dstage1
```

### Listado de etapas

La siguiente sintaxis visualiza una lista de todas las etapas de un trabajo:

```
dsjob -lstages [-useid] proyecto trabajo|id_trabajo
```

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo que contiene las etapas que se listarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

Esta sintaxis es equivalente a la función **DSGetJobInfo** con DSJ\_STAGELIST especificada como el parámetro *TipoInfo*.

## Listado de enlaces

La siguiente sintaxis visualiza una lista de todos los enlaces a o de una etapa:

```
dsjob -llinks [-useid] proyecto trabajo|id_trabajo etapa
```

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo que contiene *etapa*. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*etapa* es el nombre de la etapa que contiene los enlaces que se listarán.

Esta sintaxis es equivalente a la función **DSGetStageInfo** con DSJ\_LINKLIST especificado como el parámetro *TipoInfo*.

## Listado de parámetros

La siguiente sintaxis muestra una lista de todos los parámetros en un trabajo y sus valores:

```
dsjob -lparams [-useid] proyecto trabajo|id_trabajo
```

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo cuyos parámetros se listarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

Esta sintaxis es equivalente a la función **DSGetJobInfo** con DSJ\_PARAMLIST especificado como el parámetro *TipoInfo*.

## Listado de invocaciones

La sintaxis siguiente muestra una lista de invocaciones de un trabajo:

```
dsjob -linvocations [-useid] proyecto trabajo|id_trabajo
```

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo cuyos parámetros se listarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

## Listado de colas de gestión de carga de trabajo

La siguiente sintaxis muestra una lista de las colas de gestión de carga de trabajo:  
**dsjob -lqueues**

## Establecimiento de un alias para un trabajo

El mandato *dsjob* se puede utilizar para especificar su propio ID para un trabajo de InfoSphere DataStage.

A continuación, otros mandatos podrán utilizar ese alias para hacer referencia al trabajo.

**dsjob -jobid** [*mi\_ID*] *proyecto trabajo*

*mi\_ID* es el alias que desea establecer para el trabajo. Si omite *mi\_ID*, el mandato devolverá el alias actual para el trabajo especificado. Un alias debe ser exclusivo dentro del proyecto; si el alias ya existe se mostrará un mensaje de error.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

## Recuperación de información

El mandato *dsjob* se puede utilizar para recuperar y visualizar la información disponible sobre proyectos, trabajos, enlaces o etapas específicos.

Las distintas versiones de la sintaxis se describen en las secciones siguientes.

## Visualización de la información del trabajo

La sintaxis siguiente muestra la información disponible sobre un trabajo especificado:

**dsjob -jobinfo** [-useid] *proyecto trabajo*|*id\_trabajo*

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

Se visualiza la siguiente información:

- El estado actual del trabajo
- El nombre de cualquier trabajo de control para el trabajo

- La fecha y hora a las que se inició el trabajo
- El número de onda de la ejecución última o actual (número de referencia interno de InfoSphere DataStage)
- Estado de usuario

Esta sintaxis es equivalente a la función **DSGetJobInfo**.

## Visualización de la información de la etapa

La sintaxis siguiente visualiza toda la información disponible sobre una etapa:

**dsjob -stageinfo** [-useid] *proyecto trabajo|id\_trabajo etapa*

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo que contiene *etapa*. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*etapa* es el nombre de la etapa.

Se visualiza la siguiente información:

- Último mensaje de error notificado desde cualquier enlace a o desde la etapa
- El nombre de tipo de etapa, por ejemplo, Transformer o Aggregator
- El número de fila de entrada de enlaces primarios

Esta sintaxis es equivalente a la función **DSGetStageInfo**.

## Visualización de la información de enlace

La siguiente sintaxis muestra información sobre un enlace especificado a o de una etapa:

**dsjob -linkinfo** [-useid] *proyecto trabajo|id\_trabajo etapa enlace*

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo que contiene *etapa*. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid** (consulte “Establecimiento de un alias para un trabajo” en la página 9).

*etapa* es el nombre de la etapa que contiene *enlace*.

*enlace* es el nombre de la etapa.



Se visualiza la siguiente información:

- El último mensaje de error notificado por el enlace
- El número de filas que ha pasado un enlace

Esta sintaxis es equivalente a la función **DSGetLinkInfo**.

## Visualización de la información de parámetros

Esta sintaxis muestra información sobre el parámetro especificado:

**dsjob**  
**-paraminfo** [-useid] *proyecto trabajo*|*id\_trabajo* *parámetro*

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo que contiene *parámetro*. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*parámetro* es el nombre del parámetro.

Se visualiza la siguiente información:

- El tipo de parámetro
- El valor de parámetro
- Texto de ayuda para el parámetro proporcionado por el diseñador del trabajo
- Si se debe solicitar el valor
- El valor predeterminado especificado por el diseñador del trabajo
- Cualquier lista de valores
- La lista de valores proporcionados por el diseñador del trabajo

Esta sintaxis es equivalente a la función **DSGetParamInfo**.

## Acceso a los archivos de registro

El mandato **dsjob** se puede utilizar para añadir entradas al archivo de registro de un trabajo o para recuperar y visualizar entradas de registro específicas.

Las distintas versiones de la sintaxis se describen en las secciones siguientes.

### Adición de una entrada de registro

La sintaxis siguiente añade una entrada al archivo de registro especificado. El texto de la entrada se toma de la entrada estándar al terminal, finalizando con **Ctrl-D**.

**dsjob -log** [ **-info** | **-warn** ] [ **-useid** ] *proyecto trabajo*|*id\_trabajo*

**-info** especifica un mensaje de información. Es el valor predeterminado si no se especifica ningún tipo de entrada de registro.

**-warn** especifica un mensaje de aviso.

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el nombre del proyecto que contiene el *trabajo*.

*trabajo* es el nombre del trabajo al que hace referencia la entrada de registro. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

Esta sintaxis es equivalente a la función **DSLogEvent**.

## Visualización de una entrada de registro corta

La sintaxis siguiente muestra un resumen de entradas en un archivo de registro de trabajo:

```
dsjob  
-logsum [-type tipo] [ -max n ] [ -useid ] proyecto trabajo|id_trabajo
```

**-type** *tipo* especifica el tipo de entrada de registro que se recuperará. Si no se especifica **-type** *tipo*, se recuperan todas las entradas. *tipo* puede ser una de las opciones siguientes:

**Esta opción...**

**Recupera este tipo de entrada de registro...**

**INFO** Información.

**WARNING**  
Aviso.

**FATAL**  
Error muy grave.

**REJECT**  
Filas rechazadas de una etapa Transformer.

**STARTED**  
Todos los registros de control.

**RESET**  
Trabajo restablecido.

**BATCH**  
Control de proceso por lotes.

**ANY** Todas las entradas de cualquier tipo. Es el valor predeterminado si no se especifica *tipo*.

**-max** *n* limita el número de entradas recuperadas a *n*.

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el proyecto que contiene el *trabajo*.

*trabajo* es el trabajo cuyas entradas de registro se recuperarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

## Visualización de una entrada de registro específica

La sintaxis siguiente muestra la entrada especificada en un archivo de registro de trabajo:

```
dsjob -logdetail [-full] [-useid] trabajo proyecto|entrada_id_trabajo
```

**-full** especifique esta opción si desea que se visualicen los campos de ID de mensaje y de ID de invocación.

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el proyecto que contiene el *trabajo*.

*trabajo* es el trabajo cuyas entradas de registro se recuperarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*entrada* es el número de suceso asignado a la entrada. La primera entrada del archivo es 0.

Esta sintaxis es equivalente a la función **DSGetLogEntry** o a la función **DSGetLogEntryFull**, si se especifica **-full**.

## Identificación de la entrada más reciente

La sintaxis siguiente muestra el ID de la entrada de registro más reciente del tipo especificado:

```
dsjob -lognewest [ -useid ] proyecto trabajo|id_trabajo tipo
```

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el proyecto que contiene el *trabajo*.

*trabajo* es el trabajo cuyas entradas de registro se recuperarán. Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*tipo* puede ser una de las opciones siguientes:

**Esta opción...**

**Recupera este tipo de entrada de registro...**

**INFO** Información

**WARNING**

Aviso

<b>FATAL</b>	Error muy grave
<b>REJECT</b>	Filas rechazadas de una etapa Transformer
<b>STARTED</b>	Trabajo iniciado
<b>RESET</b>	Trabajo restablecido
<b>BATCH</b>	Proceso por lotes

Esta sintaxis es equivalente a la función **DSGetNewestLogId**.

## Generación de un informe

El mandato **dsjob** se puede utilizar para generar un informe en formato XML que contenga información de trabajo, etapa y enlace.

**dsjob -report [-useid] proyecto trabajo|id\_trabajo [tipo\_informe]**

**-useid** especifíquelo si tiene la intención de utilizar un alias de trabajo (*id\_trabajo*) en lugar de un nombre de trabajo (*trabajo*) para identificar el trabajo.

*proyecto* es el proyecto que contiene el *trabajo*.

*trabajo* especifica el trabajo sobre el que notificará por nombre de trabajo Para identificar una invocación de trabajo, utilice el formato *trabajo.id\_invocación*.

*id\_trabajo* es un alias para el trabajo que se ha establecido utilizando el mandato **dsjob -jobid**.

*tipo\_informe* es uno de los siguientes:

- **BÁSICO**: serie de texto que contiene hora de inicio/finalización, tiempo transcurrido y estado de trabajo.
- **DETALLE**: como el informe básico, pero también contiene información sobre etapas individuales y enlaces dentro del trabajo.
- **XML**: serie de texto que contiene informe XML completo.

El informe generado se graba en la salida estándar.

Esta sintaxis es equivalente a la función **DSMakeJobReport**.

---

## Mandatos para administrar proyectos

Existe un único mandato para la administración de proyectos, **dsadmin**. El mandato tiene una amplia gama de opciones.

Estas opciones se describen en los temas siguientes:

- La cláusula de inicio de sesión
- Creación de un proyecto
- Supresión de un proyecto
- Protección de un proyecto

- Habilitación/inhabilitación de la depuración automática de los archivos de registro
- Habilitación/inhabilitación de la visualización de OSH generados en trabajos paralelos.
- Habilitación/inhabilitación de propagación de columnas en tiempo de ejecución en trabajos paralelos.
- Habilitación/inhabilitación de la disponibilidad de características de administración de trabajos en el cliente del Director para un proyecto determinado.
- Establecimiento de las opciones avanzadas de tiempo de ejecución para trabajos paralelos.
- Establecimiento del nombre del directorio base para trabajos paralelos.
- Establecimiento del directorio de plantilla de trabajo desplegada para trabajos paralelos.
- Establecimiento de opciones de despliegue personalizado para trabajos paralelos.
- Creación de una nueva variable de entorno.
- Supresión de una variable de entorno.
- Establecimiento del valor de una variable de entorno
- Listado de proyectos en un servidor
- Listado de propiedades de proyecto
- Listado de variables de entorno.

## Protección de credenciales en los mandatos **dsadmin**

Puede cifrar sus credenciales y almacenar los valores cifrados en archivos para utilizarlos en los mandatos **dsadmin**. Para especificar credenciales en la línea de mandatos, opcionalmente puede utilizar la solicitud de credenciales para ocultar la contraseña en la ventana de mandatos.

### Acerca de esta tarea

Puede utilizar credenciales cifradas en los mandatos **dsadmin** para evitar tener que especificar datos visibles en la pantalla al ejecutar mandatos en la línea de mandatos. Consulte la documentación del *Mandato encrypt* de IBM InfoSphere Information Server - Guía de administración para obtener información sobre este mandato.

### Procedimiento

1. Cifre sus credenciales mediante la ejecución del mandato **encrypt**, y copie y guarde los valores cifrados en un archivo de credenciales (\*.txt). Consulte el tema del archivo de credenciales para ver las restricciones y contenido de ejemplo del archivo de credenciales.
2. Ejecute el mandato **dsadmin**. Puede especificar las credenciales con seguridad mediante uno de los métodos siguientes:
  - **Mediante el archivo de credenciales.** Si va a ejecutar un mandato **dsadmin** que requiere las credenciales de usuario, ejecute el mandato con el parámetro **-authfile** y especifique la vía de acceso completa del archivo de credenciales que desea utilizar. A continuación se muestra una sintaxis de ejemplo para la utilización del parámetro **-authfile** en un mandato **dsadmin**:

```
dsadmin -authfile c:\cred_file.txt  
-listprojects
```

- **Solicitud de credenciales.** Si desea especificar los datos de credenciales mediante la línea de mandatos, especifique sólo el **-url** o el parámetro **-domain** y el parámetro **-server**. Se le solicitará el nombre de usuario y la contraseña. (La contraseña queda oculta en la ventana de mandatos.) Si en el mandato incluye el parámetro **-url** o **-domain** y los parámetros **-server** y **-user**, se le solicitará la contraseña.

## La cláusula de inicio de sesión

De manera predeterminada, la CLI de InfoSphere DataStage se conecta al motor en el sistema local utilizando el nombre de usuario y la contraseña del usuario que ejecuta el mandato.

Para los mandatos **dsjob** y **dsadmin**, puede especificar un dominio, motor, nombre de usuario o contraseña diferente utilizando la cláusula de inicio de sesión, que es equivalente a la función **DSSetServerParams** de la API. Su sintaxis es la siguiente:

```
[-url URLdominio |  
  -domain nombre_dominio ][ -user nombre_usuario ][ -password contraseña ]  
  [ -server nombre_motor ]
```

o bien, sólo para el mandato **dsjob**:

```
-domain NONE -user nombre_usuario -password contraseña -server nombre_motor
```

*URLdominio* especifica un URL de formato completo para que el dominio se conecte a él. El URL incluye información de protocolo, host y puerto para el dominio en este formato: `https://domain:port`. El puerto adopta el valor predeterminado de 9443 si no está especificado.

*nombre\_dominio* especifica el dominio en el que se iniciará sesión. Para **dsjob**, puede establecer **-domain NONE** para iniciar sesión en el motor en lugar de en el dominio. En este caso, el nombre de usuario y la contraseña son para el motor, no para el dominio.

*nombre\_motor* especifica un motor distinto en el que se iniciará sesión.

*nombre\_usuario* especifica un nombre de usuario distinto para utilizar al iniciar la sesión.

*contraseña* especifica una contraseña distinta para utilizar al iniciar la sesión.

Los nombres de usuario cifrados y las contraseñas cifradas no están soportados en la línea de mandatos, utilice un archivo de credenciales si desea utilizar valores cifrados.

Si no desea escribir los credenciales en la línea de mandatos y no desea utilizar un archivo de credenciales, especifique únicamente el parámetro **-url** o **-domain** y el parámetro **-server** y se le solicitará el nombre de usuario y la contraseña. (La contraseña queda oculta cuando la especifica en la línea de mandatos.) Si en el mandato incluye el parámetro **-url** o **-domain** y los parámetros **-server** y **-user**, se le solicitará la contraseña. A continuación se muestra un mandato e interacción de ejemplo:

```
C:\IBM\InformationServer\Clients\Classic>dsjob  
-domain [2002:920:c000:217:9:32:217:32]:9443 -server ServidorRemoto  
-ljobs newTest  
Especifique el nombre de usuario:admin
```

Especifique la contraseña:  
Job\_ODBC

Código de estado = 0

Para sistemas que no utilizan los puertos predeterminados, para conectar a un proyecto que está en un servidor local, especifique la opción **-server** con sólo el número de puerto y no especifique el nombre de servidor. No necesita especificar el nombre de usuario ni la contraseña. Por ejemplo:

```
dsjob -server :31539 -lprojects
```

Para un inicio de sesión más seguro para los mandatos **dsjob** y **dsadmin**, puede utilizar un archivo de credenciales que contenga los datos cifrados:

**-authfile** *nombre\_archivo\_credenciales*

*nombre\_archivo\_credenciales* es la vía de acceso completa y el nombre del archivo que contiene los detalles de la conexión. Este archivo da soporte a datos cifrados y no cifrados. Consulte El archivo de credenciales para ver detalles y contenido de ejemplo del archivo.

Opcionalmente, podría especificar los detalles no cifrados en un archivo de credenciales mediante la sintaxis siguiente:

**-file** *nombre\_archivo\_credenciales nombre\_dominio nombre\_motor*

**Nota:** El archivo de credenciales **-file** no puede contener datos cifrados.

Para el mandato **dsjob**, también puede utilizar el mandato:

**-file** *nombre\_archivo\_credenciales NONE nombre\_motor*

*nombre\_dominio* especifica el dominio para el que el archivo contiene detalles de inicio de sesión. Para **dsjob**, puede establecer NONE para iniciar la sesión en el motor en lugar de hacerlo en el dominio. En este caso, el nombre de usuario y la contraseña son para el motor, no para el dominio.

*nombre\_motor* especifica el motor para el que el archivo contiene detalles de inicio de sesión.

*nombre\_archivo\_credenciales* es la vía de acceso completa y el nombre del archivo que contiene los detalles de la conexión. Este archivo da soporte sólo a datos no cifrados. El archivo debe contener la siguiente información si se inicia sesión en el dominio:

*nombre\_dominio, nombre\_motor, nombre\_usuario, contraseña*

El archivo debe contener la siguiente información si se inicia sesión en el motor:

*nombre\_motor, nombre\_usuario, contraseña*

La inclusión de la cláusula de inicio de sesión en los mandatos puede exponer el nombre de usuario y la contraseña. Es mejor utilizar la opción **-authfile** y mantener la información de inicio de sesión cifrada, o permitir que el sistema le solicite la contraseña.

## Creación de un proyecto

El mandato *dsadmin* se puede utilizar para crear proyectos.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -createproject NombreProyecto [-location UbicaciónProyecto]  
[-copyroles Proyecto origen]
```

*NombreProyecto* es el nombre del proyecto.

**-location** *UbicaciónProyecto* es la ubicación del proyecto en forma de nombre de vía de acceso y con el nombre de proyecto incluido. En el ejemplo siguiente, *prueba* es el nombre de proyecto.

```
dsadmin -createproject prueba [-location /u1/IS85/IBM/InformationServer/  
Projects/test]
```

Si no se especifica ninguna ubicación, el proyecto se crea en el directorio Projects del directorio de instalación del servidor.

**-copyroles** *Proyecto Origen* es el nombre del proyecto del que se copian los roles de usuario para el nuevo proyecto.

## Supresión de un proyecto

El mandato *dsadmin* se puede utilizar para suprimir proyectos existentes.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -deleteproject NombreProyecto
```

*NombreProyecto* es el proyecto que se suprimirá.

## Protección de un proyecto

El mandato **dsadmin** se puede utilizar para proteger o desproteger un proyecto.

Un proyecto protegido es un tipo de categoría especial de proyecto y normalmente, no se puede añadir, suprimir o modificar nada en dicho proyecto.

Los usuarios pueden ver objetos en el proyecto y realizar tareas que afectan al modo en que se ejecuta un trabajo y no el diseño del trabajo:

- Ejecutar trabajos
- Definir las propiedades del trabajo
- Definir los valores predeterminados de los parámetros del trabajo

Los usuarios con estatus de gestor de producción y administrador pueden importar los componentes de InfoSphere DataStage existentes en un proyecto protegido.

Es necesario tener un estatus de InfoSphere DataStage de tipo administrador para utilizar este mandato.

Para proteger un proyecto, ejecute este mandato:

```
dsadmin -protectproject TRUE NombreProyecto
```

donde *NombreProyecto* es el nombre del proyecto.

Para desproteger un proyecto, ejecute este mandato:

```
dsadmin -protectproject FALSE NombreProyecto
```



donde *NombreProyecto* es el nombre del proyecto.

## Habilitación/inhabilitación de la depuración automática de los archivos de registro

El mandato **dsadmin** se puede utilizar para habilitar o inhabilitar la depuración automática de los archivos de registro de trabajo de un proyecto.

### Habilitación de la depuración automática de los archivos de registro

Ejecute este mandato para habilitar la depuración automática de los archivos de registro de trabajo que tengan una antigüedad superior al número especificado de días para un proyecto:

```
dsadmin -autopurge log TRUE -days N NombreProyecto
```

donde

- *N* es el número de días transcurridos los cuales se depurarán los registros de trabajo. Por ejemplo, si especifica cinco días, se depurarán los archivos de registro de trabajo de las ejecuciones de trabajo que tengan más de cinco días de antigüedad.
- *NombreProyecto* es el proyecto para el cual debe habilitarse la depuración automática de los archivos de registro de trabajo.

Ejecute este mandato para habilitar la depuración automática de archivos de registro de trabajo cuando haya archivos de registro de trabajo que superan un número especificado de ejecuciones de trabajo para un proyecto:

```
dsadmin -autopurge log TRUE -runs N NombreProyecto
```

donde

- *N* es el número de ejecuciones de trabajo para el cual se conservan los archivos de registro. Por ejemplo, si especifica 10 ejecuciones de trabajos, se conservarán los archivos de registro de trabajo de las 10 últimas ejecuciones de trabajos y se eliminarán todos los archivos de registro anteriores.
- *NombreProyecto* es el proyecto para el cual debe habilitarse la depuración automática de los archivos de registro de trabajo.

### Inhabilitación de la depuración automática de los archivos de registro

Ejecute este mandato para inhabilitar la depuración automática de los archivos de registro de trabajos de un proyecto:

```
dsadmin -autopurge log FALSE NombreProyecto
```

donde *NombreProyecto* es el proyecto para el cual deben inhabilitarse los archivos de registro de trabajo.

## Habilitación/inhabilitación de la visualización de OSH

El mandato **dsadmin** se puede utilizar para habilitar o inhabilitar la visualización de OSH generados en trabajos paralelos.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -oshvisible TRUE | FALSE NombreProyecto
```

**Nota:** Aunque este mandato requiere un nombre de proyecto, este valor se aplica a TODOS los proyectos en el servidor.

Este mandato sólo está disponible para trabajos paralelos.

## Habilitación/inhabilitación de la propagación de columnas en tiempo de ejecución

El mandato *dsadmin* se puede utilizar para habilitar o inhabilitar la propagación de columnas en tiempo de ejecución en trabajos paralelos en un proyecto determinado.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

**dsadmin -enablercp TRUE | FALSE *NombreProyecto***

*NombreProyecto* es el proyecto cuyos trabajos paralelos tendrán habilitada o inhabilitada la propagación de columnas en tiempo de ejecución.

Este mandato sólo está disponible para trabajos paralelos.

## Habilitación/inhabilitación de la administración de trabajos desde el cliente del Director

El mandato *dsadmin* se puede utilizar para habilitar o inhabilitar las características de administración de trabajos en el cliente del Director para trabajos en un proyecto determinado.

Es necesario tener estado de administrador de InfoSphere DataStage para utilizar este mandato:

**dsadmin -enablejobadmin TRUE | FALSE *NombreProyecto***

*NombreProyecto* es el proyecto para el que se habilitará o inhabilitará la administración de trabajos en el cliente del Director.

## Habilitación/inhabilitación de la generación de informe XML

Esta opción sólo es relevante para trabajos paralelos que se compilan en un paquete de despliegue.

El paquete de despliegue puede incluir un informe de trabajo en formato XML y este mandato habilita e inhabilita la generación de este informe.

**dsadmin -enablegeneratexml TRUE | FALSE *NombreProyecto***

*NombreProyecto* es el proyecto cuyos trabajos paralelos tendrán informes XML habilitados o inhabilitados.

Este mandato sólo está disponible para trabajos paralelos.

## Habilitación/inhabilitación de las propiedades avanzadas de tiempo de ejecución

El mandato *dsadmin* se puede utilizar para establecer propiedades avanzadas de tiempo de ejecución para trabajos paralelos en un proyecto determinado.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -advancedruntime "OpcionesAvanzadasTiempoEjecución" NombreProyecto
```

*NombreProyecto* es el proyecto para cuyos trabajos paralelos se establecerán las opciones avanzadas de tiempo de ejecución especificadas.

*OpcionesAvanzadasTiempoEjecución* es el valor en el que se establecerá la propiedad y debe ir entre comillas.

Este mandato sólo está disponible para trabajos paralelos.

Para deshacer el establecimiento de las propiedades, repita el mandato con una serie vacía, por ejemplo:

```
dsadmin -advancedruntime "" miproyecto
```

## Establecimiento del directorio base

El mandato *dsadmin* se puede utilizar para establecer el directorio base para trabajos paralelos en un proyecto determinado.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -basedirectory NombreDirectorioBase NombreProyecto
```

*NombreProyecto* es el proyecto para cuyos trabajos paralelos se establece el directorio base.

*NombreDirectorioBase* es el valor en el que se establecerá la propiedad.

Este mandato sólo está disponible para trabajos paralelos.

## Establecimiento de la plantilla del directorio de despliegue

El mandato *dsadmin* se puede utilizar para establecer la plantilla del directorio de despliegue para trabajos paralelos en un proyecto determinado.

Es necesario tener estado de administrador de InfoSphere DataStage a fin de utilizar este mandato:

```
dsadmin -deploymentdirectory PlantillaDirectorio NombreProyecto
```

*NombreProyecto* es el proyecto para cuyos trabajos paralelos se define la plantilla del directorio de despliegue.

*PlantillaDirectorio* es el valor en el que se establecerá la propiedad.

Este mandato sólo está disponible para trabajos paralelos.

## Adición de una variable de entorno

El mandato *dsadmin* se puede utilizar para crear una nueva variable de entorno en un proyecto determinado.

La variable de entorno se añade a la categoría "Definida por el usuario".

```
dsadmin -envadd NombreVarEnt -type STRING | ENCRYPTED  
-prompt "TextoSolicitud" [-value "Valor"] NombreProyecto
```

*NombreVarEnt* es el nombre de la variable de entorno que se está creando.

**-type** especifica el tipo de la variable de entorno y se debe establecer en STRING o ENCRYPTED.

**-prompt** *TextoSolicitud* es el indicador que se asociará con este valor de entorno. El *TextoSolicitud* debe ir entre comas ya que puede contener espacios.

**-value** *Valor* es el valor para la nueva variable de entorno. El valor debe ir entre comillas. Si no se proporciona, será necesario establecer el valor para la variable de entorno utilizando el mandato **dsadmin -envset**.

*NombreProyecto* es el proyecto al que se añade la variable de entorno.

## Supresión de una variable de entorno

El mandato *dsadmin* se puede utilizar para suprimir una variable de entorno en un proyecto determinado.

Es imposible suprimir una variable de entorno incorporada.

**dsadmin -envdelete** *NombreVarEnt* *NombreProyecto*

*NombreVarEnt* es la variable de entorno que se está suprimiendo.

*NombreProyecto* es el proyecto del que se está suprimiendo la variable de entorno.

## Establecimiento del valor de una variable de entorno

El mandato *dsadmin* se puede utilizar para establecer el valor de una variable de entorno para un proyecto determinado.

Si está estableciendo una variable de entorno de tipo lista (por ejemplo, APT\_EXECUTION\_MODE), debe establecerla en uno de los valores internos que se permiten, en lugar de uno de los miembros de la lista como se muestran en el cliente del Administrador. Por ejemplo, si quisiera establecer APT\_EXECUTION\_MODE de forma que los trabajos paralelos se ejecutaran en una única modalidad de proceso, establecería el valor de la variable de entorno en 'ONE\_PROCESS', no 'Un proceso' como se ofrece en el cliente de Administración.

Si está estableciendo una variable de entorno de tipo booleano, establezca el valor en 1 para TRUE y en 0 para FALSE.

Si utiliza \$ENV para establecer el valor de una variable de entorno en su valor actual en el entorno, debe utilizar las comillas para garantizar que adopta el valor correcto (por ejemplo, dsadmin -envset NEW3 -value '\$ENV' dstage).

**dsadmin -envset** *NombreVarEnt* **-value** "*Valor*" *NombreProyecto*

*NombreVarEnt* es la variable de entorno cuyo valor está estableciendo.

**-value** "*Valor*" es el valor de la variable de entorno y debe ir entre comillas.

*NombreProyecto* es el proyecto para el que se está estableciendo la variable de entorno.

## Listado de proyectos

El mandato *dsadmin* se puede utilizar para listar los proyectos en una capa de motor.

**dsadmin** *-listprojects*

Lista todos los proyectos en la capa de motor.

## Listado de propiedades

El mandato *dsadmin* se puede utilizar para listar las propiedades de un proyecto.

Se listan las propiedades siguientes:

- Si el OSH generado está visible en trabajos paralelos.
- Si la propagación de columnas en tiempo de ejecución está habilitada en trabajos paralelos.
- El nombre del directorio base para trabajos paralelos.
- Opciones avanzadas de tiempo de ejecución para trabajos paralelos.
- Mandatos de despliegue personalizados para trabajos paralelos.
- Plantilla de directorio de trabajo desplegada.
- Si la administración de trabajos está habilitada en el cliente del Director.

Las propiedades de trabajos paralelos sólo se listarán si hay trabajos paralelos disponibles.

**dsadmin** *-listproperties NombreProyecto*

*NombreProyecto* es el proyecto para el que se listarán las propiedades.

## Listado de variables de entorno

El mandato *dsadmin* se puede utilizar para listar las variables de entorno en un proyecto.

**dsadmin** *-listenv NombreProyecto*

---

## Mandatos para la importación desde archivos .dsx

Puede importar objetos desde archivos .dsx en el repositorio especificado.

El mandato DSXImportService tiene varias opciones. Puede utilizar el mandato para importar el contenido de todo un archivo .dsx, u objetos especificados dentro de un archivo .dsx, y puede generar un informe del proceso de importación. También puede utilizar el mandato DSXImportService para listar el contenido de un archivo .dsx.

Puede ejecutar el mandato DSXImportService en cualquier sistema que tenga un ASBNode instalado.

## Importar objetos desde un archivo .dsx

Para ejecutar el mandato DSXImportService, debe especificar los detalles de la capa de servicios y la vía de acceso del archivo que desea importar.

### Finalidad

El mandato DSXImportService importa los objetos desde un archivo .dsx hasta un repositorio de IBM InfoSphere DataStage.

## Parámetros

Puede especificar detalles de dominio, nombre de usuario y contraseña para la capa de servicios de las siguientes maneras:

- como valores en un archivo de credenciales, que pueden cifrarse. La especificación de los detalles de conexión cifrados en un archivo de credenciales es la opción más segura.
- como valores sin cifrar en un archivo. Si especifica detalles en un archivo, también puede especificar los detalles en la línea de mandatos para alterar temporalmente parte del contenido del archivo. Por ejemplo, podría especificar un dominio distinto pero tomar el nombre de usuario y la contraseña tal como se especifican en el archivo, de forma que podría utilizar el mismo archivo para conectar a sistemas distintos. La especificación del nombre de usuario y contraseña en un archivo en lugar de en la línea de mandato proporciona mayor seguridad.
- como valores sin cifrar en la línea de mandatos. Si especifica solamente los detalles del dominio, se le solicitarán el nombre de usuario y la contraseña. (La contraseña queda oculta cuando la especifica en la línea de mandatos.) Si especifica los detalles de dominio y de nombre de usuario, se le solicitará la contraseña.

La sintaxis siguiente especifica un archivo de credenciales que contiene detalles de conexión:

```
-ISAuthFile archivoAutorizIS
-DSPProject proyectoDS
-DSXFile archivoDsx
[-Overwrite | -OverwriteReadOnly]
[-Verbose]
[-StopOnError]
[importación_seleccionada]
```

La sintaxis siguiente especifica un archivo que contiene detalles de conexión:

```
-ISFile archivoIS
[-ISHost hostIS[:puerto]]
[-ISUser usuarioIS]
[-ISPassword contraseñaIS]
[-DSHost hostDS[:puerto]]
-DSPProject proyectoDS
-DSXFile archivoDsx
[-Overwrite | -OverwriteReadOnly]
[-Verbose]
[-StopOnError]
[importación_seleccionada]
```

La sintaxis siguiente especifica los detalles de conexión directamente en la línea de mandatos:

```
-ISHost hostIS[:puerto]
[-ISUser usuarioIS [-ISPassword contraseñaIS]]
[-DSHost hostDS[:puerto]]
-DSPProject proyectoDS
-DSXFile archivoDsx
[-Overwrite | -OverwriteReadOnly]
[-Verbose]
[-StopOnError]
[importación_seleccionada]
```

**-ISAuthFile *archivoAutorizIS***

Especifica un archivo de credenciales que contiene detalles de conexión para la

capa de servicios. Los detalles de la conexión se puede cifrar. El archivo de credenciales debe contener estos cuatro argumentos:

```
domain=hostIS[:puerto]
user=usuarioIS
password=contraseñaIS
server=hostDS[:puerto]]
```

El archivo sólo debe contener estos cuatro argumentos y debe especificarse un argumento por línea. Los argumentos son sensibles a mayúsculas y minúsculas y adoptan el mismo formato que los argumentos de la línea de mandatos.

Para obtener más información sobre los archivos de credenciales incluyendo un ejemplo, consulte .

**-ISFile** *archivoIS*

Especifica un archivo de credenciales que contiene detalles de conexión no cifrados para la capa de servicios. Si se especifican detalles de conexión en la línea de mandatos, estos sustituyen a los que están definidos dentro del archivo. El archivo especifica los detalles de conexión en los argumentos siguientes:

```
-ISHost hostIS[:puerto]
-ISUser usuarioIS
-ISPassword contraseñaIS
```

El archivo sólo debe contener estos tres argumentos y debe especificarse un argumento por línea. Los argumentos son sensibles a mayúsculas y minúsculas y adoptan el mismo formato que los argumentos de la línea de mandatos.

**-ISHost** *hostIS[:puerto]*

Especifica el nombre del sistema que alberga el dominio. Si no especifica un número de puerto, se utiliza el número de puerto predeterminado, 9443.

**-ISUser** *usuarioIS*

Especifica el nombre de un usuario en la capa de servicios.

**-ISPassword** *contraseñaIS*

Especifica la contraseña para el usuario de dominio.

**-DSHost** *hostDS[:puerto]*

Si el nivel del motor no está instalado en el mismo sistema que el nivel de servicios, debe especificar el sistema que alberga la capa de motor. Si no especifica un número de puerto, se utilizará el número de puerto predeterminado.

**-DSProject** *proyectoDS*

Especifica el proyecto en el que se importarán los objetos.

**-DSXFile** *archivoDsx*

Especifica el archivo .dsx desde el que se importan los objetos. Puede especificar un nombre de vía de acceso completo, que puede ser local o remoto.

**-Overwrite**

Especifique esta opción para sobrescribir objetos existentes en el repositorio. Si no especifica esta opción, al intentar volver a importar los objetos existentes se producirá un error.

**-OverwriteReadOnly**

Esta opción hace lo mismo que **-Overwrite** pero adicionalmente sustituirá los elementos de sólo lectura que se encuentren. Si no se especifica esta opción, los elementos de sólo lectura existentes no se sobrescribirán.

### **-Verbose**

Especifique esta opción para generar un informe completo de los objetos importados. De manera predeterminada, sólo se notificarán los errores de importación.

### **-StopOnError**

Especifique esta opción para detener la importación si se encuentra un error al importar un objeto.

### *importación\_seleccionada*

Aquí puede especificar las opciones para importar los objetos seleccionados de un archivo .dsx. Especifique el tipo de objeto y el nombre de objeto tal como se especifica en la tabla siguiente. Puede especificar un nombre completo o un nombre abreviado para el tipo de objeto.

*Tabla 1. Opciones de importación seleccionadas*

Opción abreviada	Opción completa	Tipo de objeto
-JB	-JOB	trabajo
-EJ	-EXECUTABLEJOB	ejecutable de trabajo
-DE	-DATAELEMENT	elemento de datos
-TD	-TABLEDEFINITION	definición de tabla
-ST	-STAGETYPE	tipo de etapa
-TR	-TRANSFORM	transformación
-RT	-ROUTINE	rutina
-ID	-IMSDATABASE	base de datos IMS
-IV	-IMSVIEWSET	conjunto de vistas de IMS
-MP	-MACHINEPROFILE	perfiles de máquina
-SC	-SHAREDCONTAINER	contenedor compartido
-QR	-QSRULEASSEMBLY	conjunto de reglas QualityStage
-PS	-PARAMETERSET	conjunto de parámetros
-DC	-DATACONNECTION	conexión de datos

## **Ejemplo**

El siguiente mandato importa los objetos del archivo oldproject.dsx al proyecto de la etapa. Los detalles de conexión se especifican en el archivo miconexión.

```
DSXImportService -ISFile miconexión -DSProject dstage  
-DSXFile c:\export\oldproject.dsx
```

El siguiente mandato importa los objetos del archivo nuevoproyecto.dsx al proyecto dstage. Los detalles de conexión se especifican en el mandato.

```
DSXImportService -ISHost isservier:9443 -ISUser wgamsworth  
-ISPassword paddock -DSProject dstage -DSXFile c:\export\nuevoproyecto.dsx
```

El siguiente mandato importa los objetos del archivo partproject.dsx al proyecto dstage. Los detalles de conexión se especifican en el archivo miconexión.

```
DSXImportService -ISFile miconexión -DSProject dstage  
-DSXFile c:\export\proyectoanterior.dsx -JB trabajo1 trabajo2  
-TD APTSchemas\CFDImport\XYZ_RECORD -ROUTINE rutinal rutina2
```



## Listado del contenido de un archivo .dsx

Puede utilizar el mandato DSXImportService para listar el contenido de un archivo .dsx.

### Finalidad

Utilice el parámetro Lista con el mandato DSXImportService para listar los objetos que contiene un archivo .dsx. No necesita conectar a un dominio para listar el archivo.

### Parámetros

La siguiente sintaxis especifica un archivo a listar:

```
-List -DSXFile <archivodsx>  
]
```

**-DSXFile** *archivoDsx*

Especifica el archivo .dsx a listar. Puede especificar un nombre de vía de acceso completa y la vía de acceso puede ser local o remota.

### Ejemplo

El mandato siguiente lista el contenido del archivo proyectoanterior.dsx.

```
DSXImportService -List -DSXFile c:\archives\proyectoanterior.dsx
```

---

## Mandatos para la comprobación y reparación de proyectos

Puede comprobar si los activos de tiempo de diseño de un proyecto está sincronizados con sus activos del proyecto del servidor de InfoSphere DataStage correspondientes, y realizar reparaciones, si es necesario.

Si los activos del tiempo de diseño del un proyecto que se mantienen en el repositorio de metadatos no están conectados con el proyecto del servidor, es posible que el proyecto del servidor o bien los activos de este queden inutilizables.

Puede utilizar el mandato **SyncProject** para comprobar si existen incoherencias, y para reparar las que se detecten. Puede utilizar el mandato como parte de la planificación de mantenimiento habitual a fin de comprobar si existen problemas. En caso de que se detecten problemas, utilice el mandato en modalidad de reparación para solucionar los problemas. Como alternativa, puede utilizar el mandato de forma interactiva y se le preguntará si es necesario llevar a cabo una acción de reparación.

El mandato **SyncProject** puede detectar e intentar reparar los siguientes problemas:

#### Falta un proyecto en el repositorio de metadatos

Este error surge si un proyecto del servidor existe, pero el proyecto correspondiente no existe en el repositorio de metadatos. **SyncProject** intenta reparar el error eliminando el proyecto del servidor.

#### Falta un proyecto del servidor

Este error surge si un proyecto existe en el repositorio de metadatos, pero el proyecto del servidor correspondiente no existe. **SyncProject** intenta reparar el error volviendo a crear el proyecto del servidor. Si **SyncProject** consigue volver a crear el proyecto, es necesario volver a compilar los trabajos que el proyecto contuviera.

### Los archivos del proyecto del servidor están corruptos

El proyecto del servidor mantiene descripciones de objetos de proyecto en archivos. Si estos archivos están corruptos, **SyncProject** intenta resolver el problema devolviendo el archivo de proyecto corrompido a su estado inicial.

### Falta un trabajo o contenedor compartido del repositorio de metadatos

Este error surge si un trabajo o contenedor compartido del servidor existe, pero el trabajo o el contenedor compartido correspondiente del repositorio de metadatos no existe. **SyncProject** intenta reparar el error eliminando el trabajo o contenedor compartido del servidor.

### Falta un trabajo o contenedor compartido del servidor

Este error surge si existe un trabajo o contenedor compartido en el repositorio de metadatos, pero el trabajo o contenedor compartido del servidor correspondiente no existe. **SyncProject** intenta reparar el error volviendo a crear el trabajo o contenedor compartido del servidor. Si **SyncProject** consigue volver a crear el trabajo correctamente, este deberá volverse a compilar.

### Archivos de trabajo del motor del servidor corrompidos

Este error surge si los archivos en los que el motor del servidor define un trabajo ejecutable y su estado pasan a estar corrompidos. Para recuperar un archivo corrompido, **SyncProject** vuelve a crear un archivo vacío. Como consecuencia, es posible que sea necesario volver a compilar el trabajo y que se pierdan el historial de registros del trabajo y la información de estados del trabajo actual.

### Carpeta de contenedor compartido o trabajo del servidor incorrecta

Este error surge si un trabajo o contenedor compartido está en una carpeta distinta del repositorio de metadatos en el servidor. **SyncProject** intenta reparar el error actualizando el atributo de carpeta del trabajo o contenedor compartido en el proyecto del servidor.

Si el mandato **SyncProject** no puede reparar un proyecto del servidor, puede utilizar el mandato **SyncProject** para reconstruir el proyecto desde el repositorio de metadatos.

## Parámetros de autenticación para el mandato SyncProject

Al utilizar el mandato **SyncProject** para comprobar o arreglar los proyectos, debe especificar los detalles de autenticación.

### Parámetros

Puede especificar el host de la capa de servicios, nombre de usuario y contraseña en un archivo o puede especificar los detalles directamente en la línea de mandatos. Si especifica detalles en un archivo, también puede especificar los detalles en la línea de mandatos para alterar temporalmente parte del contenido del archivo. Por ejemplo, puede especificar un host distinto pero utilizar el nombre de usuario y la contraseña que están especificados en el archivo, y así podrá utilizar el mismo archivo para conectarse a distintos sistemas. La especificación del nombre de usuario y contraseña en un archivo en lugar de en la línea de mandato proporciona mayor seguridad.

La sintaxis siguiente especifica un archivo que contiene detalles de conexión:

```
SyncProject -ISFile archivoIS
[-ISHost hostIS[:puerto]]
[-ISUser usuarioIS]
[-ISPassword contraseñaIS]
[-DSHost hostDS[:puerto]]
```

La sintaxis siguiente especifica los detalles de conexión directamente en la línea de mandatos:

```
SyncProject
-ISHost hostIS[:puerto]
-ISUser usuarioIS
-ISPassword contraseñaIS
[-DSHost hostDS[:puerto]]
```

#### **-ISFile** *archivoIS*

Especifica el nombre de archivo que contiene los detalles de conexión. Este parámetro proporciona un nivel de seguridad ocultando los detalles de inicio de sesión de la vista. Si utiliza este parámetro, no es necesario que proporcione los detalles de conexión en la línea de mandatos. Sin embargo, si se especifican detalles de conexión en la línea de mandatos, estos sobrescribirán los detalles de conexión definidos dentro del archivo. El archivo especifica los detalles de conexión en los argumentos siguientes:

```
-ISHost hostIS -ISUser usuarioIS -ISPassword contraseñaIS
```

El contenido del archivo requiere un argumento por línea. Los argumentos siguientes son sensibles a mayúsculas y minúsculas.

#### **-ISHost** *hostIS[:puerto]*

Especifica el nombre del sistema que alberga el dominio. Si no especifica un número de puerto, se utiliza el número de puerto predeterminado, 9443.

#### **-ISUser** *usuarioIS*

Especifica el nombre de un usuario en la capa de servicios.

#### **-ISPassword** *contraseñaIS*

Especifica la contraseña para el usuario de la capa de servicios.

#### **-DSHost** *hostDS[:puerto]*

Si el nivel del motor no está instalado en el mismo sistema que el nivel de servicios, debe especificar el sistema que alberga el motor. Si no especifica un número de puerto, se utilizará el número de puerto predeterminado.

## Comprobación de incoherencias en los proyectos

Utilice el mandato **SyncProject** para comprobar si los proyectos son coherentes entre los repositorios de tiempo de diseño y de tiempo de ejecución.

### Parámetros

El mandato **SyncProject** tiene la sintaxis siguiente:

```
SyncProject parámetros_autenticación
-Project NombreProyecto...
[-JobNombreTrabajo...]
-Report [NombreArchivo]
[-StopOnError]
```

#### **-Project** *NombreProyecto...*

Especifica uno o varios proyectos que se van a comprobar.

**-Job** *NombreTrabajo...*

Si especifica únicamente un proyecto en el mandato, especifica una lista de trabajos de un proyecto para comprobar.

**-report** [*NombreArchivo*]

Especifica que se requiere un informe. Tiene la opción de especificar un nombre de archivo, de manera que el informe se grabará en dicho archivo.

## Ejemplos

El mandato siguiente solicita un informe de coherencia para el proyecto denominado dstage3. En este caso, tanto la capa de servicios como la capa de motor están en el sistema denominado R101.

```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword -project dstage3  
-report
```

El mandato tiene como resultado el siguiente informe:

Informe del restaurador de DSEngine

5 de febrero de 2010 9:32:00 AM

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage3  
-----

Se han encontrado 0 problemas.

Resumen general  
-----

Se han encontrado 0 problemas.

El mandato siguiente solicita un informe de coherencia para todos los proyectos denominados dstagen. El mandato devuelve resultados para los proyectos dstage3, dstage4, dstage5 y dstage9. El informe se graba en un archivo y también se visualiza en la pantalla.

```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword -project dstage*  
-report c:\myprojrep
```

En este caso, se han encontrado dos incoherencias en el proyecto con nombre dstage9. El resultado es el informe siguiente, que se graba en el archivo c:\myprojrep:

Informe del restaurador de DSEngine

5 de febrero de 2010 9:39:00 AM

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage3  
-----

```

Se han encontrado 0 problemas.

Proyecto de DataStage: dstage4
-----

Se han encontrado 0 problemas.

Proyecto de DataStage: dstage5
-----

Se han encontrado 0 problemas.

Proyecto de DataStage = dstage9
-----

Se han encontrado 2 problemas.

Falta el 'TrabajoPrueba' del trabajo de motor DS.
La categoría 'CategoríaIncorrecta' del trabajo de motor DS 'Trabajo2Prueba'
debería ser 'CategoríaCorrecta'

Resumen general
-----
Se han encontrado 2 problemas.

```

## Reparar proyectos incoherentes

Puede utilizar el mandato **SyncProject** con el parámetro **-fix** para reparar proyectos, de cuyas incoherencias se ha informado.

### Parámetros

El mandato **SyncProject** tiene la sintaxis siguiente:

```

SyncProject parámetros_autenticación
-Project NombreProyecto...
[-JobNombreTrabajo...]
-Fix [NombreArchivo]
-StopOnError

```

**-Project *NombreProyecto...***

Especifica uno o varios proyectos que se deben reparar.

**-Job *NombreTrabajo...***

Si especifica únicamente un proyecto en el mandato, puede especificar una lista de trabajos de un proyecto para reparar.

**-Fix [*NombreArchivo*]**

Especifica que se requieren arreglos para los errores, de los que se ha informado en el proyecto o proyectos mencionados. Tiene la opción de especificar un nombre de archivo, de la misma forma que los resultados de los arreglos que se han intentado se graban en el archivo.

**-StopOnError**

Especifique este parámetro para parar si **SyncProject** no puede reparar una incoherencia. No se intentarán más arreglos.

### Ejemplos

El mandato siguiente solicita que los arreglos se realicen en los proyectos dstage3 y dstage5.

```
SyncProject -ISFile islogin -project dstage3 dstage5 -Fix
```

El mandato puede realizar las reparaciones necesarias, y el resultado es el informe siguiente:

Resultados de los arreglos del restaurador de DSEngine

5 de febrero de 2009 9:39:00 AM

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage3

SOLUCIONADO: Falta el 'TrabajoPrueba' del trabajo de motor DS.  
SOLUCIONADO: La categoría 'CategoríaIncorrecta' del trabajo de motor DS  
'Trabajo2Prueba' debería ser 'CategoríaCorrecta'.

Se han solucionado 2 problemas.  
Quedan 0 problemas.

Proyecto de DataStage: dstage5

SOLUCIONADO: Falta el 'Trabajo2Prueba' del trabajo de motor DS.

Se ha solucionado 1 problema.  
Quedan 0 problemas.

Resumen general

Se han solucionado 3 problemas.  
Quedan 0 problemas.

El mandato siguiente solicita que los arreglos se realicen en el proyecto dstage6, y los resultados se graban en el archivo c:\fixresults:

SyncProject -ISFile islogin -project dstage6 -fix c:\fixresults

El mandato no puede realizar las reparaciones necesarias. El mandato tiene como resultado el siguiente informe, que se graba en el archivo c:\fixresults:

Resultados de los arreglos del restaurador de DSEngine

5 de febrero de 2009 9:39:00 AM

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage6

SIN SOLUCIONAR: Falta el 'SiguienteTrabajoPrueba' del trabajo de motor DS.  
SIN SOLUCIONAR: La carpeta 'CarpetaIncorrecta' del trabajo de motor DS  
'SiguienteTrabajo2Prueba' debería ser 'CarpetaCorrecta'.

Resumen general

Se han solucionado 0 problemas.  
Quedan 2 problemas.

## Comprobación y reparación interactivas de proyectos

Utilice el mandato **SyncProject** de forma interactiva para comprobar que los activos de tiempo de ejecución de un proyecto coinciden con sus activos de tiempo de diseño. Si se encuentra una incoherencia, se le solicitará que decida si desea intentar solucionarla.

### Parámetros

El mandato **SyncProject** tiene la sintaxis siguiente:

*SyncProject parámetros\_autenticación*

**-Project** *NombreProyecto...*

*[-JobNombreTrabajo...]*

**-Report** [*NombreArchivo*]

**-Fix** [*NombreArchivo*]

**-StopOnError**

**-Project** *NombreProyecto...*

Especifique uno o varios proyectos que se van a comprobar.

**-Job***NombreTrabajo...*

Si especifica únicamente un proyecto en el mandato, puede especificar una lista de trabajos de un proyecto para comprobar.

**-report** [*NombreArchivo*]

Especifica que se requiere un informe. Tiene la opción de especificar un nombre de archivo, de manera que el informe se grabará en dicho archivo.

**-fix** [*NombreArchivo*]

Especifica que se han solucionado las incoherencias. Tiene la opción de especificar un nombre de archivo, así como información sobre el arreglo, y su éxito o anomalía se graban en este archivo.

**-StopOnError**

Especifique esta opción para parar si SyncProject no consigue solucionar una incoherencia.

### Ejemplos

El mandato siguiente solicita un informe de coherencia para el proyecto denominado dstage3, y que se solucionen todas las incoherencias que se hayan encontrado.

```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword  
-Project dstage3 -Report -Fix
```

El resultado es el informe siguiente:

Informe del restaurador de DSEngine

5 de febrero de 2009 9:39:00 AM

```
Host IS    = R101  
Puerto IS = 9443  
Usuario IS = admin
```

```
Host DS    = R101  
Puerto DS  = 3158
```

Proyecto de DataStage: dstage3

-----

Se han encontrado 2 problemas.

Falta el 'TrabajoPrueba' del trabajo de motor DS.  
La categoría 'CategoríaIncorrecta' del trabajo de motor DS 'Trabajo2Prueba'  
debería ser 'CategoríaCorrecta'

Resumen general

-----  
Se han encontrado 2 problemas.

Se le solicita si desea solucionar los problemas que se han detectado. SyncProject  
informa a continuación del éxito del arreglo de la siguiente manera:

Resultados de los arreglos del restaurador de DSEngine

5 de febrero de 2009 9:39:00 AM

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage3

-----  
SOLUCIONADO: Falta el 'TrabajoPrueba' del trabajo de motor DS.  
SOLUCIONADO: La categoría 'CategoríaIncorrecta' del trabajo de motor DS  
'Trabajo2Prueba' debería ser 'CategoríaCorrecta'.

Se han solucionado 2 problemas.  
Quedan 0 problemas.

Resumen general

-----  
Se han solucionado 2 problemas.  
Quedan 0 problemas.

## Reconstruir un proyecto

Utilice el mandato **SyncProject** para reconstruir un proyecto desde el repositorio.

### Parámetros

El directorio del proyecto se suprime antes de que el proyecto se reconstruya desde el repositorio. El proyecto reconstruido no mantiene los datos de registro ni de estado de trabajo.

Es necesario volver a compilar los trabajos y rutinas del proyecto reconstruido.

El mandato **SyncProject** tiene la sintaxis siguiente:

SyncProject *parámetros\_authenticación*  
-Project *NombreProyecto*  
-Reconstruct

**-Project** *NombreProyecto*

Especifica el proyecto que se va a reconstruir.

**-Reconstruct**

Especifica que un proyecto va a reconstruirse.

### Ejemplo

El siguiente mandato solicita que se reconstruya un proyecto llamado dstage1 desde el repositorio.



```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword -project dstage1  
-reconstruct
```

El mandato tiene como resultado el siguiente informe:

Resultados de los arreglos del restaurador de DSEngine

Miércoles, 20 de enero, 09:13:35 GMT 2010

Host IS = R101  
Puerto IS = 9443  
Usuario IS = admin

Host DS = R101  
Puerto DS = 3158

Proyecto de DataStage: dstage1  
-----

SOLUCIONADO: Suprimido y creado de nuevo el proyecto de motor DS: dstage1.  
SOLUCIONADO: Falta el trabajo de motor DS 'px1'.  
SOLUCIONADO: Falta el trabajo de motor DS 'sv1'.  
SOLUCIONADO: Falta el contenedor compartido de motor DS 'pxscl'.  
SOLUCIONADO: Falta el contenedor compartido de motor DS 'svscl'.

Resumen general  
-----

Se han solucionado 5 problemas.  
No se han podido solucionar 0 problemas.

## Creación de una copia de seguridad de un proyecto

Utilice el mandato **SyncProject** para crear una copia de seguridad de un proyecto.

### Parámetros

El mandato **SyncProject** tiene la sintaxis siguiente:

```
SyncProject parámetros_autenticación  
-Project NombreProyecto  
-Backup [NombreArchivo]
```

**-Project** *NombreProyecto*

Especifica el proyecto del que se va a hacer una copia de seguridad.

**-Backup** [*NombreArchivo*]

De forma opcional, especifique un nombre de archivo para el archivo de archivado tar que se ha creado. Si no especifica ningún nombre de archivo, el archivo de archivado tar se denominará *NombreProyecto.tar*

### Ejemplo

El mandato siguiente solicita que se realice una copia de seguridad del proyecto denominado dstage3 en un archivo de archivado tar denominado dstage3.tar.

```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword -project dstage3  
-backup c:\archives\dstage3.tar
```

El mandato tiene como resultado el siguiente informe:

Realización de la copia de seguridad.  
La copia de seguridad se ha realizado correctamente.

## Restaurar un proyecto

Utilice el mandato **SyncProject** para restaurar un proyecto de una copia de seguridad.

### Parámetros

El mandato **SyncProject** tiene la sintaxis siguiente:

`SyncProject parámetros_autenticación`

`-Project NombreProyecto`

`-Restore NombreArchivo`

**-Project** *NombreProyecto*

Especifica el proyecto que se va a restaurar.

**-Restore** [*NombreArchivo*]

Especifica el nombre del archivo de archivado tar que ha creado antes.

### Ejemplo

El mandato siguiente solicita que se restaure un proyecto denominado dstage3 a partir del archivo de archivado tar denominado dstage3.tar.

```
SyncProject -ISHost R101:9443 -ISUser admin -ISPassword pword -project dstage3  
-restore dstage3.tar
```

El mandato tiene como resultado el siguiente informe:

Realización de la restauración del proyecto.

La restauración del proyecto se ha realizado correctamente.

---

## Capítulo 2. InfoSphere DataStage Development Kit (Interfaces de control de trabajos)

El kit de despliegue de InfoSphere DataStage y otras interfaces de control de trabajos son herramientas que se pueden utilizar para ejecutar trabajos directamente en el motor. Estas herramientas proporcionan flexibilidad para ejecutar trabajos en varios casos de ejemplo sin la necesidad de utilizar un cliente del Director.

InfoSphere DataStage proporciona una serie de métodos que permiten ejecutar trabajos de servidor o paralelos directamente en el motor, sin utilizar el cliente del Director. Los métodos son:

- API de C/C++ (InfoSphere DataStage development kit)
- Llamadas a BASIC de InfoSphere DataStage
- Mandatos de la interfaz de línea de mandatos (CLI)
- Macros de InfoSphere DataStage

Estos métodos se pueden utilizar en distintas situaciones, de la forma siguiente:

- API. Mediante la API puede compilar un programa autocontenido que se puede ejecutar en cualquier lugar del sistema, siempre que se pueda conectar a un host de la capa de motor a través de la red.
- BASIC. Los programas compilados utilizando la interfaz de BASIC de InfoSphere DataStage se pueden ejecutar desde cualquier host de la capa de motor de la red. Puede utilizar esta interfaz para definir trabajos que ejecuten y controlen otros trabajos. El trabajo de control se puede ejecutar desde el cliente del Director como cualquier otro trabajo, o bien directamente en la máquina del motor del indicador de solicitud de TCL.
- Interfaz de línea de mandatos. La CLI se puede utilizar desde la línea de mandatos de cualquier host de la capa de motor de la red. Además, mediante este método puede ejecutar trabajos en otros motores.
- Macros. Se puede utilizar un conjunto de macros en diseños de trabajos o en programas BASIC. Estos se utilizan habitualmente para recuperar información sobre otros trabajos.

---

### InfoSphere DataStage Development Kit

Utilice InfoSphere DataStage Development Kit para acceder a la API de InfoSphere DataStage, una interfaz de programación de aplicaciones C ó C++

.

Esta sección proporciona información general sobre la utilización de la API de InfoSphere DataStage.

#### El archivo de cabecera dsapi.h

Incluya el archivo de cabecera de la API de InfoSphere DataStage con todos los programas de la API.

La API de DataStage proporciona un archivo de cabecera que se debe incluir con todos los programas de la API. El archivo de cabecera incluye prototipos para

todas las funciones de la API de InfoSphere DataStage. Está ubicado en el directorio `$DSHOME/include`. Su formato depende de las señales que haya definido:

- Si están definidas las señales `_STDC_` o `WIN32`, los prototipos están en estilo C ANSI.
- Si está definida la señal `_cplusplus`, los prototipos están en formato C++ con las declaraciones entre:  

```
extern "C" {...}
```
- De lo contrario, los prototipos están en formato Kernighan y Ritchie.

## Estructuras de datos, datos de resultados y hebras

Las funciones de la API de InfoSphere DataStage devuelven información sobre objetos como punteros a elementos de datos. Esto se realiza directa o indirectamente estableciendo punteros en los elementos de una estructura de datos proporcionada por el emisor de la llamada.

Se asigna a cada hebra dentro de una aplicación de llamada un área de almacenamiento aparte. Cada llamada a una rutina de la API de InfoSphere DataStage sobrescribe el contenido existente de esta área de datos con los resultados de la llamada y devuelve un puntero en el área para los datos solicitados.

Por ejemplo, la función **DSGetProjectList** obtiene una lista de proyectos de InfoSphere DataStage y la función **DSGetProjectInfo** obtiene una lista de trabajos dentro de un proyecto. Cuando se llama a la función **DSGetProjectList**, esta recupera la lista de proyectos, los almacena en el área de datos de la hebra y devuelve un puntero a esta área. Si la misma hebra luego llama a **DSGetProjectInfo**, se recupera la lista de trabajos y se almacena en el área de datos de la hebra, sobrescribiendo la lista de proyectos. El puntero de la lista de trabajos en la estructura de datos proporcionada hace referencia al área de datos de la hebra.

Esto significa que si es necesario volver a utilizar los resultados de una función de la API de InfoSphere DataStage, la aplicación debe realizar su propia copia de los datos antes de realizar una nueva llamada a la API de InfoSphere DataStage. De forma opcional, se pueden utilizar las llamadas en varias hebras.

La API de InfoSphere DataStage almacena errores para cada hebra: una llamada a la función **DSGetLastError** devuelve el último error generado dentro de la hebra de llamada.

## Grabación de programas de la API de InfoSphere DataStage

Los programas de la API de InfoSphere DataStage se graban siguiendo un proceso simple y lógico que utiliza eficazmente las funciones y en la secuencia correcta.

### Acerca de esta tarea

La aplicación debe utilizar las funciones de la API de InfoSphere DataStage en un orden lógico para asegurarse de que las conexiones se abren y cierran correctamente y de que los trabajos se ejecutan de forma eficaz. El siguiente procedimiento sugiere una breve descripción de la lógica de programa a seguir y las funciones a utilizar en cada paso:

## Procedimiento

1. Si es necesario, establezca el nombre de host, nombre de usuario y contraseña a utilizar para conectar a InfoSphere DataStage (**DSSetServerParams**).
2. Obtenga la lista de proyectos válidos (**DSGetProjectList**).
3. Abra un proyecto (**DSSetProject**).
4. Obtenga una lista de trabajos (**DSGetProjectInfo**).
5. Abra uno o más trabajos (**DSSetOpenJob**).
6. Liste los parámetros de trabajo (**DSGetParamInfo**).
7. Bloquee el trabajo (**DSSetLockJob**).
8. Establezca los parámetros y límites del trabajo (**DSSetJobLimit**, **DSSetParam**).
9. Inicie la ejecución del trabajo (**DSSetRunJob**).
10. Sondee el trabajo o espere la finalización del trabajo (**DSSetWaitForJob**, **DSSetStopJob**, **DSGetJobInfo**).
11. Desbloquee el trabajo (**DSSetUnlockJob**).
12. Visualice un resumen de las entradas de registro del trabajo (**DSFindFirstLogEntry**, **DSFindNextLogEntry**).
13. Visualice detalles de los sucesos de registro específicos (**DSGetNewestLogId**, **DSGetLogEntry**, **DSGetLogEntryFull**).
14. Examine y visualice detalles de etapas de trabajo (**DSGetJobInfo** - lista de etapas, **DSGetStageInfo**).
15. Examine y visualice detalles de enlaces dentro de etapas activas (**DSGetStageInfo** - lista de entradas, **DSGetLinkInfo**).
16. Cierre todos los trabajos abiertos (**DSSetCloseJob**).
17. Desconéctese del proyecto (**DSSetCloseProject**).

## Creación de una aplicación de la API de InfoSphere DataStage

Cree una aplicación que utiliza la API de InfoSphere DataStage grabando el programa, compilando el código y enlazando la aplicación.

### Acerca de esta tarea

Todo lo que necesita para crear una aplicación que utiliza la API de InfoSphere DataStage se encuentra en un subdirectorio denominado dsdk (DataStage Development Kit) en el directorio de instalación IBM\InformationServer\Server en la máquina del host de la capa de motor.

## Procedimiento

1. Grabe el programa, incluyendo el archivo de cabecera dsapi.h, en todos los módulos de origen que utilizan la API de InfoSphere DataStage.
2. Compile el código. Asegúrese de que se haya definido la señal WIN32. (Esto sucede automáticamente en el entorno del compilador de Microsoft Visual C/C++.)
3. Enlace la aplicación, incluido vmdsapi.lib, en la lista de bibliotecas que se incluirán.

## Redistribución de aplicaciones

Redistribuya aplicaciones, DLL o bibliotecas cuando vaya a ejecutar aplicaciones de la API de InfoSphere DataStage, en función del rol del sistema en el que se ha instalado la aplicación.

Si tiene la intención de ejecutar la aplicación de la API de InfoSphere DataStage en la capa de motor donde está instalado, no es necesario que incluya las bibliotecas ni DLL de la API de InfoSphere DataStage, ya que éstas se instalan como parte de la capa de motor.

Si desea ejecutar la aplicación desde un sistema utilizado como cliente de InfoSphere DataStage, debe redistribuir la siguiente biblioteca con la aplicación:

`vmdsapi.dll`

Si tiene la intención de ejecutar el programa desde un sistema que no tenga instalados la capa de motor ni el cliente de InfoSphere DataStage, además de la biblioteca mencionada anteriormente, debe también redistribuir lo siguiente:

`dscInt32.dll dsrpc32.dll ACS_common_cpp.dll ACS_client_cpp.dll  
invocation_cpp.dll xmogrt.dll`

Debe localizar estos archivos en los lugares donde estarán en la vía de búsqueda de cualquier usuario que utiliza la aplicación, por ejemplo, en el directorio `%SystemRoot%\System32`.

## Funciones de la API

Utilice las funciones suministradas en la API de InfoSphere DataStage para realizar diversas tareas con proyectos y trabajos.

Esta sección detalla las funciones proporcionadas en la API de InfoSphere DataStage. Estas funciones se describen en orden alfabético. La tabla siguiente describe brevemente las funciones categorizadas por uso:

Uso	Función	Descripción
Acceso a proyectos	<b>DSCloseProject</b>	Cierra un proyecto que se ha abierto con <b>DSOpenProject</b> .
	<b>DSGetProjectList</b>	Recupera una lista de todos los proyectos en el motor.
	<b>DSGetProjectInfo</b>	Recupera una lista de trabajos en un proyecto.
	<b>DSOpenProject</b>	Abre un proyecto.
	<b>DSSetServerParams</b>	Establece el nombre de motor, nombre de usuario y contraseña que se utilizarán para un trabajo.
Acceso a trabajos	<b>DSCloseJob</b>	Cierra un trabajo que se ha abierto con <b>DSOpenJob</b> .
	<b>DSGetJobInfo</b>	Recupera información sobre un trabajo, como por ejemplo la fecha y hora de la última ejecución, nombre de parámetro, etc.
	<b>DSLckJob</b>	Bloquea un trabajo antes de establecer los parámetros de trabajo o de iniciar la ejecución de un trabajo.
	<b>DSOpenJob</b>	Abre un trabajo.

Uso	Función	Descripción
	<b>DSRunJob</b>	Ejecuta un trabajo.
	<b>DSSStopJob</b>	Termina anormalmente un trabajo en ejecución.
	<b>DSUnlockJob</b>	Desbloquea un trabajo, permitiendo que otros procesos lo utilicen.
	<b>DSWaitForJob</b>	Espera hasta que finaliza un trabajo.
Acceso a parámetros de trabajo	<b>DSGetParamInfo</b>	Recupera información sobre un parámetro de trabajo.
	<b>DSSetJobLimit</b>	Establece límites de aviso y proceso de fila para un trabajo.
	<b>DSSetParam</b>	Establece valores de parámetro del trabajo.
Acceso a etapas	<b>DSGetStageInfo</b>	Recupera información sobre una etapa dentro de un trabajo.
Acceso a enlaces	<b>DSGetLinkInfo</b>	Recupera información sobre un enlace de una etapa activa dentro de un trabajo.
Acceso a entradas de registro	<b>DSFindFirstLogEntry</b>	Recupera entradas de un registro que cumplen los criterios especificados.
	<b>DSFindNextLogEntry</b>	Busca la siguiente entrada de registro que cumple los criterios especificados en <b>DSFindFirstLogEntry</b> .
	<b>DSGetLogEntry</b>	Recupera la entrada de registro especificada.
	<b>DSGetLogEntryFull</b>	Recupera la entrada de registro especificada, incluidos el ID de mensaje y el ID de invocación.
	<b>DSGetLogEventIds</b>	Recupera una lista de ID de registro de suceso para una invocación de trabajo determinada.
	<b>DSGetNewestLogId</b>	Recupera la entrada más reciente del registro.
	<b>DSLogEvent</b>	Añade una nueva entrada al registro.
Administración de proyectos y trabajos	<b>DSAddEnvVar</b>	Añade una nueva variable de entorno.
	<b>DSAddProject</b>	Añade un proyecto.
	<b>DSDeleteEnvVar</b>	Suprime una variable de entorno.
	<b>DSDeleteProject</b>	Suprime un proyecto.
	<b>DSListEnvVars</b>	Lista variables de entorno.

Uso	Función	Descripción
	DSListProjectProperties	Lista las propiedades de un proyecto.
	DSSetEnvVar	Establece una variable de entorno.
	DSSetProjectProperty	Establece la propiedad para un proyecto.
Manejo de errores	<b>DSGetLastError</b>	Recupera el valor del último código de error generado por la hebra de llamada.
	<b>DSGetLastErrorMsg</b>	Recupera el texto del último informe notificado.

## DSAddEnvVar

Utilice la función DSAddEnvVar para añadir una variable de entorno al proyecto especificado. La variable se añade a la categoría definida por el usuario.

### Sintaxis

```
int DSAddEnvVar(    DSPROJECT proyectoh,
                  char *NombreVarEnt,
                  char *Tipo,
                  char *TextoSolicitud,
                  char *Valor);
```

### Parámetros

*Proyectoh* es el valor devuelto de **DSOpenProject**

*NombreVarEnt* es el nombre de la variable de entorno

*Tipo* es DSA\_ENVVAR\_TYPE\_STRING para variables de entorno de tipo serie o DSA\_ENVVAR\_TYPE\_ENCRYPTED para variables de entorno cifradas.

*TextoSolicitud* es el texto visible al usuario que describe la variable de entorno.

*Valor* es el valor en el que se establecerá la variable de entorno o "".

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR

Si la función falla, el valor de retorno es uno de los siguientes:

- DSJE\_BADENVVARNAME Nombre de variable de entorno no válido
- DSJE\_BADENVVARTYPE Tipo no válido
- DSJE\_BADENVVARPROMPT No se ha emitido ninguna solicitud
- DSJE\_READENVVARDEFNS No se han podido leer las definiciones de variable de entorno
- DSJE\_READENVVARVALUES No se han podido leer los valores de variable de entorno
- DSJE\_WRITEENVVARDEFNS No se han podido grabar las definiciones de variable de entorno



- DSJE\_WRITEENVVARVALUES No se han podido grabar los valores de variable de entorno
- DSJE\_DUPENVVARNAME La variable de entorno ya existe
- DSJE\_ENCODEFAILED No se ha podido codificar un valor cifrado

## Comentarios

Para utilizar este método, es necesario que el programa se haya conectado anteriormente a un proyecto utilizando **DSOpenProject**. Devuelve un descriptor de contexto al proyecto, *proyectooh*.

## DSAddProject

Utilice la función DSAddProject para crear un proyecto nuevo. El usuario que ejecuta el código que contiene esta función debe ser un administrador de InfoSphere DataStage.

### Sintaxis

```
int DSAddProject(
    char *NombreProyecto,
    char *UbicaciónProyecto);
```

### Parámetros

*NombreProyecto* es el nombre del proyecto a crear.

*UbicaciónProyecto* es el nombre de vía de acceso del directorio en el que crear el proyecto. Para crear un proyecto en el directorio de proyecto predeterminado (*vía de acceso de instalación/Proyectos/nombreProyecto*), este argumento se debe establecer en "".

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR

Si la función falla, el valor de retorno es uno de los siguientes:

- DSJE\_NOTADMINUSER El usuario no es administrador
- DSJE\_ISADMINFAILED No se ha podido determinar si el usuario es administrador
- DSJE\_BADPROJNAME Se ha proporcionado un nombre de proyecto no válido
- DSJE\_GETDEFAULTPATHFAILED No se ha podido determinar el directorio de proyecto predeterminado
- DSJE\_BADPROJLOCATION Se ha proporcionado un nombre de vía de acceso no válido
- DSJE\_INVALIDPROJECTLOCATION Se ha proporcionado un nombre de vía de acceso no válido
- DSJE\_OPENFAILED No se ha podido abrir el archivo UV.ACCOUNT
- DSJE\_READUFAILED No se ha podido bloquear el registro de bloqueo de creación de proyecto
- DSJE\_ADDPROJECTBLOCKED Otro usuario está añadiendo un proyecto
- DSJE\_ADDPROJECTFAILED No se ha podido añadir el proyecto
- DSJE\_LICENSEPROJECTFAILED No se ha podido otorgar licencia al proyecto

- DSJE\_RELEASEFAILED No se ha podido liberar registro de bloqueo de creación de proyecto

## DSCloseJob

Utilice la función DSCloseJob para cerrar un trabajo que se ha abierto utilizando la función DSOpenJob.

### Sintaxis

```
int DSCloseJob(
    DSJOB DescriptorContextoTrabajo
);
```

### Parámetro

*DescriptorContextoTrabajo* es el valor devuelto de DSOpenJob.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es:

DSJE\_BADHANDLE *DescriptorContextoTrabajo* no válido.

### Comentarios

Si el trabajo está bloqueado cuando se llama a **DSCloseJob**, se desbloquea.

Si el trabajo está en ejecución cuando se llama a **DSCloseJob**, se permite que finalice el trabajo y la función devuelve un valor de DSJE\_NOERROR inmediatamente.

## DSCloseProject

Utilice la función DSCloseProject para cerrar un trabajo que se ha abierto utilizando la función DSOpenProject.

### Sintaxis

```
int DSCloseProject(
    DSPROJECT DescriptorContextoProyecto
);
```

### Parámetro

*DescriptorContextoProyecto* es el valor devuelto de DSOpenProject.

### Valor de retorno

Esta función devuelve siempre un valor de DSJE\_NOERROR.

### Comentarios

Los trabajos abiertos en el proyecto se cierran, se permite que finalicen los trabajos en ejecución y la función devuelve inmediatamente el valor de retorno.

## DSDeleteEnvVar

Utilice la función DSDeleteEnvVar para suprimir una variable de entorno definida por el usuario en un proyecto especificado.

### Sintaxis

```
int DSDeleteEnvVar(  
    DSPROJECT Proyecto,  
    char *NombreVarEnt  
);
```

### Parámetros

*Proyecto* es el valor devuelto de DSOpenProject

*NombreVarEnt* es el nombre de la variable de entorno

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR

Si la función falla, el valor de retorno es uno de los siguientes:

- DSJE\_READENVVARDEFNS No se han podido leer las definiciones de variable de entorno
- DSJE\_READENVVARVALUES No se han podido leer los valores de variable de entorno
- DSJE\_BADENVVAR La variable de entorno no existe
- DSJE\_WRITEENVVARDEFNS No se han podido grabar las definiciones de variable de entorno
- DSJE\_WRITEENVVARVALUES No se han podido grabar los valores de variable de entorno
- DSJE\_NOTUSERDEFINED La variable de entorno no está definida por el usuario y por lo tanto no se puede suprimir

Si la función falla, el valor de retorno es uno de los siguientes:

## DSDeleteProject

Utilice la función DSDeleteProject para suprimir un proyecto. El usuario que ejecuta el código que contiene esta función debe ser un administrador de InfoSphere DataStage. Los trabajos planificados para ejecutarse que están incluidos en este proyecto también se suprimen.

### Sintaxis

```
int DSDeleteProject(  
    char *NombreProyecto  
);
```

### Parámetro

*NombreProyecto* es el nombre del proyecto a suprimir.

## Valor de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR

Si la función falla, el valor de retorno es uno de los siguientes:

- DSJE\_NOTADMINUSER El usuario no es administrador
- DSJE\_ISADMINFAILED No se ha podido determinar si el usuario es administrador
- DSJE\_OPENFAILED No se ha podido abrir el archivo UV.ACCOUNT
- DSJE\_READUFAILED No se ha podido bloquear el registro del proyecto
- DSJE\_RELEASEFAILED No se ha podido liberar el registro del proyecto
- DSJE\_LISTSCHEDULEFAILED No ha podido obtener lista de trabajos planificados para el proyecto
- DSJE\_CLEARSCCHEDULEFAILED No se ha podido borrar los trabajos planificados para el proyecto
- DSJE\_DELETEPROJECTBLOCKED Proyecto bloqueado por otro usuario
- DSJE\_NOTAPROJECT No se ha podido iniciar sesión en el proyecto
- DSJE\_ACCOUNTPATHFAILED No se ha podido obtener vía de acceso de cuenta
- DSJE\_LOGTOFAILED No se ha podido iniciar sesión en la cuenta UV
- DSJE\_DELPROJFAILED No se ha podido suprimir la definición de proyecto
- DSJE\_DELPROJFILESFAILED No se han podido suprimir los archivos de proyecto

## DSFindFirstLogEntry

Utilice la función DSFindFirstLogEntry para recuperar las entradas de registro que cumplen los criterios especificados y graban la primera entrada en una estructura de datos. Puede leer las entradas de registro posteriores de lectura utilizando la función DSFindNextLogEntry.

### Sintaxis

```
int DSFindFirstLogEntry(  
    DSJOB DescriptorContextoTrabajo,  
    int TipoSuceso,  
    time_t HoraInicio,  
    time_t HoraFinalización,  
    int NúmeroMáx,  
    DSLOGEVENT *Suceso);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de DSOpenJob.

*TipoSuceso* es una de las claves siguientes:

**Esta clave...**

**Recupera este tipo de mensaje...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

**DSJ\_LOGFATAL**

Muy grave

**DSJ\_LOGREJECT**

Rechazo de fila de Transformer

**DSJ\_LOGSTARTED**

Trabajo iniciado

**DSJ\_LOGRESET**

Trabajo restablecido

**DSJ\_LOGBATCH**

Control de proceso por lotes

**DSJ\_LOGOTHER**

Todos los demás tipos de registro

**DSJ\_LOGANY**

Cualquier tipo de suceso

*HoraInicio* limita los sucesos de registro devueltos a aquellos que se han producido en la hora y fecha especificadas o con posterioridad a ellas. Establezca este valor en 0 para devolver el suceso más reciente.

*HoraFinalización* limita los sucesos de registro devueltos a los que se han producido antes de la fecha y hora especificadas. Establezca este valor en 0 para devolver todas las entradas hasta la más reciente.

*NúmeroMáx* especifica el número máximo de entradas de registro a recuperar, empezando con la última.

*Suceso* es un puntero a una estructura de datos a utilizar para mantener la primera entrada de registro recuperada.

**Valores de retorno**

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR y los detalles de resumen de la primera entrada de registro se graban en *Suceso*.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción****DSJE\_NOMORE**

No existen sucesos que coincidan con los criterios de filtro.

**DSJE\_NO\_MEMORY**

No se ha podido asignar memoria para los resultados del motor.

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_BADTYPE**

Valor de *TipoSuceso* no válido.

**DSJE\_BADTIME**

Valor de *HoraInicio* u *HoraFinalización* no válido.

**DSJE\_BADVALUE**

Valor de *NúmeroMáx* no válido.

## Comentarios

Las entradas de registro recuperadas se almacenan en la memoria caché para poder ser recuperadas por llamadas subsiguientes a **DSFindNextLogEntry**. Las entradas de registro almacenadas en la memoria caché que no son procesadas por una llamada **DSFindNextLogEntry** se descartan en la siguiente llamada a **DSFindFirstLogEntry** (para cualquier trabajo) o cuando se cierra el proyecto.

**Nota:** Las entradas de registro se almacenan en la memoria caché por descriptor de contexto de proyecto. Varias hebras que utilizan el mismo descriptor de contexto de proyecto abierto se deben coordinar para acceder a **DSFindFirstLogEntry** y **DSFindNextLogEntry**.

## DSFindNextLogEntry

Utilice la función **DSFindNextLogEntry** para recuperar la siguiente entrada de registro de la memoria caché.

### Sintaxis

```
int DSFindNextLogEntry(  
    DSJOB DescriptorContextoTrabajo,  
    DSLOGEVENT *Suceso  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*Suceso* es un puntero a una estructura de datos a utilizar para mantener la siguiente entrada de registro.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR** y los detalles de resumen de la siguiente entrada de registro disponible se graban en *Suceso*.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal**    **Descripción**

**DSJE\_NOMORE**

Se han devuelto todos los sucesos que coinciden con los criterios de filtro.

**DSJE\_SERVER\_ERROR**

Error interno. El motor ha devuelto datos no válidos.

## Comentarios

Esta función recupera la siguiente entrada de registro de la memoria caché de entradas producida por una llamada a **DSFindFirstLogEntry**.

**Nota:** Las entradas de registro se almacenan en la memoria caché por descriptor de contexto de proyecto. Varias hebras que utilizan el mismo descriptor de contexto de proyecto abierto se deben coordinar para acceder a **DSFindFirstLogEntry** y **DSFindNextLogEntry**.

## DSGetCustInfo

Utilice la función `DSGetCustInfo` para obtener información que se ha notificado al final de la ejecución de determinadas etapas paralelas. La información recopilada, y disponible para el interrogatorio, se especifica durante el tiempo de diseño. Por ejemplo, la información de la etapa Transformer se especifica en el separador **Desencadenantes** del recuadro de diálogo Propiedades de la etapa Transformer.

### Sintaxis

```
int DSGetCustInfo(    DSJOB DescriptorContextoTrabajo,
                    char *NombreEtapa,
                    char *NombreInfoPers
                    int TipoInfo,
                    DSSTAGEINFO *InfoRetorno
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de `DSOpenJob`.

*NombreEtapa* es un puntero a una serie terminada en nulo que especifica el nombre de la etapa que se interrogará.

*InfoPersNombre* es un puntero a una serie terminada en nulo que especifica el nombre de la variable que se interrogará (como está configurado en el separador **Desencadenantes**).

*TipoInfo* es una de las claves siguientes:

**Esta clave...**

**Devuelve esta información...**

**DSJ\_CUSTINFOVALUE**

El valor de la variable especificada.

**DSJ\_CUSTINFODESC**

Descripción de la variable.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es `DSJE_NOERROR`.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción**

**DSJE\_NOT\_AVAILABLE**

No hay instancias de la información solicitada en la etapa.

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_BADSTAGE**

*NombreEtapa* no hace referencia a una etapa conocida del trabajo.

**DSJE\_BADCUSTINFO**

*NombreInfoPers* no hace referencia a un elemento *infopers* conocido.

**DSJE\_BADTYPE**

*TipoInfo* no válido.

## DSGetJobInfo

Utilice la función DSGetJobInfo para recuperar información sobre el estado de un trabajo.

### Sintaxis

```
int DSGetJobInfo( DSJOB DescriptorContextoTrabajo,  
                 int TipoInfo,  
                 DSJOBINFO *InfoRetorno  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*TipoInfo* es una clave que indica la información que se devolverá y puede tener uno de los valores siguientes:

**Esta clave...**

**Devuelve esta información...**

**DSJ\_JOBSTATUS**

El estado actual del trabajo.

**DSJ\_JOBNAME**

El nombre del trabajo al que hace referencia *DescriptorContextoTrabajo*.

**DSJ\_JOBCONTROLLER**

El nombre del trabajo que controla el trabajo al que hace referencia *DescriptorContextoTrabajo*.

**DSJ\_JOBSTARTTIMESTAMP**

La fecha y hora a las que se inició el trabajo.

**DSJ\_JOBWAVENO**

El número de fase de la ejecución última o actual.

**DSJ\_JOBDESC**

La descripción del trabajo especificada en el recuadro de diálogo Propiedades del trabajo.

**DSJ\_JOBFULLDESSC**

La descripción completa especificada en el recuadro de diálogo Propiedades del trabajo.

**DSJ\_JOBDMISERVICE**

Se establece en true (verdadero) si se trata de un trabajo de servicio web.

**DSJ\_JOBMULTIINVOKABLE**

Se establece en true (verdadero) si este trabajo da soporte a varias invocaciones.

**DSJ\_PARAMLIST**

Una lista de nombres de parámetros de trabajo. Separados por nulos.

**DSJ\_STAGELIST**

Una lista de etapas activas en el trabajo. Separados por nulos.

**DSJ\_USERSTATUS**

El valor, si existe, establecido como el estado de usuario por el trabajo.

**DSJ\_JOBCONTROL**

Si se ha emitido una solicitud de detención para el trabajo al que hace referencia *DescriptorContextoTrabajo*.



**DSJ\_JOBPID**

ID de proceso del proceso DSD.RUN.

**DSJ\_JOBLASTTIMESTAMP**

La fecha y hora a las que finalizó el trabajo.

**DSJ\_JOBINVOCATIONS**

Lista de ID de invocación de trabajo. Los ID están separados por nulos.

**DSJ\_JOBINTERIMSTATUS**

El estado de un trabajo después de que haya ejecutado todas las etapas y trabajos controlados, pero antes de que haya intentado ejecutar una subrutina posterior al trabajo. (Está diseñado para ser utilizado por una subrutina posterior al trabajo para obtener el estado del trabajo actual.)

**DSJ\_JOBINVOCATIONID**

Nombre de invocación del trabajo al que hace referencia  
*DescriptorContextoTrabajo*.

**DSJ\_JOBDESC**

Una descripción del trabajo.

**DSJ\_STAGELIST2**

Una lista de etapas pasivas en el trabajo. Separados por nulos.

**DSJ\_JOBELAPSED**

El tiempo transcurrido del trabajo en segundos.

*InfoRetorno* es un puntero a una estructura de datos **DSJOBINFO** donde se almacena la información solicitada. La estructura de datos **DSJOBINFO** contiene una unión con un elemento para cada uno de los posibles valores de retorno de la llamada a **DSGetJobInfo**. Para obtener más información, consulte "Estructuras de datos" en la página 80.

**Valores de retorno**

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción****DSJE\_NOT\_AVAILABLE**

No hay instancias de la información solicitada en el trabajo.

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo no válido.*

**DSJE\_BADTYPE**

*TipoInfo no válido.*

**Comentarios**

Para trabajos controlados, esta función se puede utilizar antes o después de una llamada a **DSRunJob**.

**DSGetLastError**

Utilice la función **DSGetLastError** para devolver el último valor del código de error de la hebra que realiza la llamada.

## Sintaxis

```
int DSGetLastError(void);
```

## Valores de retorno

El valor de retorno es el valor del último código de error. La sección "Valores de retorno" de cada página de consulta indica las condiciones bajo las cuales la función establece el último código de error.

## Comentarios

Utilice **DSGetLastError** inmediatamente después de cualquier función cuyo valor de retorno cuando el resultado es anómalo pudiera contener datos útiles; de lo contrario, una función satisfactoria podría restablecer el valor de nuevo a 0 (DSJE\_NOERROR).

**Nota:** Varias hebras no se sobrescriben los códigos de error entre ellas.

## DSGetLastErrorMsg

Utilice la función **DSGetLastErrorMsg** para recuperar el texto del último error notificado del motor.

## Sintaxis

```
char *DSGetLastErrorMsg(  
    DSPROJECT DescriptorContextoProyecto  
);
```

## Parámetro

*DescriptorContextoProyecto* es el valor devuelto de **DSOpenProject** o NULO.

## Valores de retorno

El valor de retorno es un puntero a un conjunto de series terminadas en nulo, una para cada línea del mensaje de error asociado con el último error generado por el motor en respuesta a una llamada a la función de la API de InfoSphere DataStage. Utilice **DSGetLastError** para determinar cuál es el número de error.

El ejemplo siguiente muestra el contenido del almacenamiento intermedio donde <nulo> representa el carácter NULO de terminación:

```
línea1<nulo>línea2<nulo>línea3<nulo><nulo>
```

La función **DSGetLastErrorMsg** devuelve NULO si no hay ningún mensaje de error.

## Comentarios

Si *DescriptorContextoProyecto* es NULO, esta función recupera el mensaje de error asociado con la última llamada a **DSOpenProject** o **DSGetProjectList**; de lo contrario, devuelve el último mensaje asociado con el proyecto especificado.

El texto del error se borra después de una llamada a **DSGetLastErrorMsg**.

**Nota:** El texto recuperado por una llamada a **DSGetLastErrorMsg** se relaciona con el último error generado por el motor y no necesariamente con el último error

notificado de vuelta a una hebra que utiliza la API de InfoSphere DataStage API. Las distintas hebras que utilizan la API de InfoSphere DataStage deben cooperar a fin de obtener el texto de mensaje de error correcto.

## DSGetLinkInfo

Utilice la función DSGetLinkInfo para recuperar información que está relacionada con un enlace específico de la etapa activa especificada de un trabajo.

### Sintaxis

```
int DSGetLinkInfo(  
    DSJOB DescriptorContextoTrabajo,  
    char *NombreEtapa,  
    char *NombreEnlace,  
    int TipoInfo,  
    DSLINKINFO *InfoRetorno);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NombreEtapa* es un puntero a una serie de caracteres terminada en nulo que especifica el nombre de la etapa activa que se interrogará.

*NombreEnlace* es un puntero a una serie de caracteres terminada en nulo que especifica el nombre de un enlace (entrada o salida) conectado a la etapa.

*TipoInfo* es una clave que indica la información que se devolverá y es uno de los valores siguientes:

Valor	Descripción
-------	-------------

<b>DSJ_LINKLASTERR</b>	
------------------------	--

	El último mensaje de error notificado por el enlace.
--	------------------------------------------------------

<b>DSJ_LINKNAME</b>	
---------------------	--

	Nombre del enlace.
--	--------------------

<b>DSJ_LINKROWCOUNT</b>	
-------------------------	--

	Número de filas que han descendido por el enlace.
--	---------------------------------------------------

<b>DSJ_LINKEXTROWCOUNT</b>	
----------------------------	--

	Número de filas que han descendido por el enlace, que se han devuelto en el campo rowCountList de la estructura DSLINKINFO. Utilice este valor cuando el campo de enteros DSJ_LINKROWCOUNT haya devuelto más filas de las soportadas.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>DSJ_LINKSQLSTATE</b>	
-------------------------	--

	Valor de SQLSTATE del último mensaje de error.
--	------------------------------------------------

<b>DSJ_LINKDBMSCODE</b>	
-------------------------	--

	Valor de DBMSCODE del último mensaje de error.
--	------------------------------------------------

<b>DSJ_LINKDESC</b>	
---------------------	--

	Descripción del enlace.
--	-------------------------

<b>DSJ_LINKSTAGE</b>	
----------------------	--

	Nombre de la etapa en el otro extremo del enlace.
--	---------------------------------------------------

<b>DSJ_INSTROWCOUNT</b>	
-------------------------	--

	Lista separada por nulos de recuentos de filas, uno para cada instancia para trabajos paralelos.
--	--------------------------------------------------------------------------------------------------

*InfoRetorno* es un puntero a una estructura de datos **DSJOBINFO** donde se almacena la información solicitada. La estructura de datos **DSJOBINFO** contiene una unión con un elemento para cada uno de los posibles valores de retorno de la llamada a **DSGetLinkInfo**.

## Valor de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

### Señal Descripción

#### **DSJE\_NOT\_AVAILABLE**

No existe ninguna instancia disponible de la información solicitada.

#### **DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no era válido.

#### **DSJE\_BADTYPE**

*TipoInfo* no se ha reconocido.

#### **DSJE\_BADSTAGE**

*NombreEtapa* no hace referencia a una etapa conocida del trabajo.

#### **DSJE\_BADLINK**

*NombreEnlace* no hace referencia a un enlace conocido para la etapa en cuestión.

## Comentarios

Esta función se puede utilizar antes o después de una llamada a **DSRunJob**.

## DSGetLogEntry

Recupera información detallada sobre una entrada específica en un registro de trabajo.

### Sintaxis

```
int DSGetLogEntry(  
    DSJOB DescriptorContextoTrabajo,  
    int IDSuceso,  
    DSLOGDETAIL *Suceso  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*IDSuceso* es el identificador para el suceso que se recuperará. Consulte “Comentarios” en la página 55.

*Suceso* es un puntero a una estructura de datos que mantiene detalles de la entrada de registro.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR** y la estructura de sucesos contiene los detalles del suceso solicitado.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_BADHANDLE</b>	
-----------------------	--

	<i>DescriptorContextoTrabajo</i> no válido.
--	---------------------------------------------

<b>DSJE_SERVER_ERROR</b>	
--------------------------	--

	Error interno. El motor ha devuelto datos no válidos.
--	-------------------------------------------------------

<b>DSJE_BADEVENTID</b>	
------------------------	--

	Suceso no válido si es para un trabajo especificado.
--	------------------------------------------------------

## Comentarios

Las entradas en el archivo de registro se numeran secuencialmente a partir de 0. El ID de suceso más reciente se puede obtener mediante una llamada a **DSGetNewestLogId**. Cuando se borra un registro, siempre permanecerá una única entrada que indicará que se ha borrado el registro.

## DSGetLogEntryFull

Recupera información detallada sobre una entrada específica en un registro de trabajo, incluidos el ID de mensaje y el ID de invocación.

### Sintaxis

```
int DSGetLogEntryFull(  
    DSJOB DescriptorContextoTrabajo,  
    int IDsuceso,  
    DSLOGDETAILFULL *Suceso  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*IDsuceso* es el identificador para el suceso que se recuperará. Consulte “Comentarios” en la página 56.

*Suceso* es un puntero a una estructura de datos que mantiene detalles de la entrada de registro.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR** y la estructura de sucesos contiene los detalles del suceso solicitado.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_BADHANDLE</b>	
-----------------------	--

	<i>DescriptorContextoTrabajo</i> no válido.
--	---------------------------------------------

<b>DSJE_SERVER_ERROR</b>	
--------------------------	--

	Error interno. El motor ha devuelto datos no válidos.
--	-------------------------------------------------------

<b>DSJE_BADEVENTID</b>	
------------------------	--

	Suceso no válido si es para un trabajo especificado.
--	------------------------------------------------------

## Comentarios

Las entradas en el archivo de registro se numeran secuencialmente a partir de 0. El ID de suceso más reciente se puede obtener mediante una llamada a **DSGetNewestLogId**. Cuando se borra un registro, siempre permanecerá una única entrada que indicará que se ha borrado el registro.

## DSGetLogEventIds

Recupera una lista de ID de registro de suceso para una invocación de trabajo determinada.

### Sintaxis

```
int DSGetLogEventIds(  
    DSJOB DescriptorContextoTrabajo,  
    int NúmeroEjecución,  
    char *Filtro,  
    char **Lista);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NúmeroEjecución* identifica la ejecución de invocación de trabajo para la que se devuelven los ID de suceso. Normalmente un valor de cero solicita los ID para la ejecución más reciente de la invocación de trabajo. Para recuperar detalles para ejecuciones anteriores, proporcione valores negativos, como por ejemplo -1 para obtener detalles sobre la ejecución anterior a la más reciente, -2 para obtener detalles sobre la ejecución anterior a esa, y así sucesivamente. En los casos dónde conozca los números explícitos de ejecución, puede recuperar detalles proporcionando el número de ejecución como un valor positivo.

*Filtro* restringe los tipos de entrada de registro de suceso para los que se devuelven los ID. De forma predeterminada, se devuelven los ID para todas las entradas de suceso de registro. Incluya caracteres en la serie del filtro para restringir las entradas de la forma siguiente:

I	Informativo
W	Aviso
F	Muy grave
S	Sucesos de inicio o finalización
B	Sucesos de proceso por lotes o control
R	Sucesos de depuración o restablecimiento
J	Sucesos de rechazo

*Lista* es un puntero a un almacenamiento intermedio de caracteres que recibe la lista de ID devueltos.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

**DSJE\_BADHANDLE**

Descriptor de contexto de trabajo no válido.

**DSJE\_BADTYPE**

Valor de *Filtro* no válido.

**DSJE\_BADVALUE**

Valor de *NúmeroEjecución* no válido.

**Comentarios**

Para utilizar este método, es necesario que el programa haya adquirido previamente un descriptor de contexto de trabajo llamando a **DSOpenJob()**.

El número de ejecución para una invocación de trabajo se restablece cuando se compila el trabajo; por lo tanto, no es posible utilizar este método para recuperar ID de suceso de trabajo para ejecuciones que se han producido antes de la compilación más reciente del trabajo.

**DSGetNewestLogId**

Utilice la función **DSGetNewestLogId** para obtener el identificador de la entrada más reciente en el registro de trabajos.

**Sintaxis**

```
int DSGetNewestLogId(
    DSJOB DescriptorContextoTrabajo,
    int TipoSuceso
);
```

**Parámetros**

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*TipoSuceso* es una clave que especifica el tipo de entrada de registro cuyo identificador desea recuperar y puede ser una de las siguientes:

**Esta clave...**

**Recupera este tipo de entrada de registro...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

**DSJ\_LOGFATAL**

Muy grave

**DSJ\_LOGREJECT**

Rechazo de fila de Transformer

**DSJ\_LOGSTARTED**

Trabajo iniciado

**DSJ\_LOGRESET**

Trabajo restablecido

**DSJ\_LOGOTHER**

Cualquier tipo de suceso de registro

**DSJ\_LOGBATCH**  
Control de proceso por lotes

**DSJ\_LOGANY**  
Cualquier tipo de suceso

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es el identificador positivo de la entrada más reciente del tipo solicitado en el archivo de registro del trabajo.

Si la función falla, el valor de retorno es -1. Utilice **DSGetLastError** para recuperar uno de los códigos de error siguientes:

**Señal Descripción**

**DSJE\_BADHANDLE**  
*DescriptorContextoTrabajo* no válido.

**DSJE\_BADTYPE**  
Valor de *TipoSuceso* no válido.

## Comentarios

Utilice esta función para determinar el ID de la entrada más reciente de un archivo de registro antes de iniciar la ejecución de un trabajo. Después de que haya iniciado o finalizado el trabajo, es posible determinar qué entradas han sido añadidas por la ejecución del trabajo.

## DSGetParamInfo

Utilice la función **DSGetParamInfo** para recuperar información sobre un parámetro determinado

### Sintaxis

```
int DSGetParamInfo( DSJOB DescriptorContextoTrabajo, char *NombreParám,  
DPARAMINFO *InfoRetorno  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NombreParám* es un puntero a una serie terminada en nulo que especifica el nombre del parámetro que se interrogará.

*InfoRetorno* es un puntero a la estructura de datos **DSPARAMINFO** donde se almacena la información solicitada. Para obtener más información, consulte “Estructuras de datos” en la página 80.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción**

**DSJE\_SERVER\_ERROR**  
Error interno. El motor ha devuelto datos no válidos.



## DSJE\_BADHANDLE

*DescriptorContextoTrabajo no válido.*

## Comentarios

A diferencia de otras funciones de recuperación de información, **DSGetParamInfo** devuelve toda la información relativa a un elemento especificado en una única llamada. La estructura de datos **DSPARAMINFO** contiene toda la información necesaria para solicitar un nuevo valor de parámetro de un usuario y parcialmente validarlo. Consulte “Estructuras de datos” en la página 80.

Esta función se puede utilizar antes o después de que se haya emitido una llamada a **DSRunJob**:

- Si se llama después de una llamada satisfactoria a **DSRunJob**, la información solicitada hace referencia a esa ejecución del trabajo.
- Si se llama antes de una llamada a **DSRunJob**, la información recuperada hace referencia a una ejecución anterior del trabajo y no a cualquier llamada a **DSSetParam** que se pueda haber emitido.

## DSGetProjectInfo

Utilice la función **DSGetProjectInfo** para obtener una lista de trabajos en un proyecto.

### Sintaxis

```
int DSGetProjectInfo(  
    DSPROJECT DescriptorContextoProyecto,  
    int TipoInfo,  
    DSPROJECTINFO *InfoRetorno  
);
```

### Parámetros

*DescriptorContextoProyecto* es el valor devuelto de **DSOpenProject**.

*TipoInfo* es una clave que indica la información que se devolverá.

**Esta clave...**

**Recupera este tipo de entrada de registro...**

#### DSJ\_JOBLIST

Lista todos los trabajos dentro del proyecto.

#### DSJ\_PROJECTNAME

Nombre del proyecto actual.

#### DSJ\_HOSTNAME

Nombre de host de la capa de motor.

*InfoRetorno* es un puntero a una estructura de datos **DSPROJECTINFO** donde se almacena la información solicitada.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

#### **DSJE\_NOT\_AVAILABLE**

No hay trabajos compilados definidos dentro del proyecto.

#### **DSJE\_BADTYPE**

*TipoInfo no válido.*

### **Comentarios**

La estructura de datos **DSPROJECTINFO** contiene una **unión** con un elemento para cada uno de los posibles valores de retorno desde una llamada a **DSGetProjectInfo**.

**Nota:** La lista devuelta contiene los nombres de todos los trabajos conocidos al proyecto, independientemente de si se pueden abrir o no.

## **DSGetProjectList**

Utilice la función **DSGetProjectList** para obtener una lista de todos los proyectos en el sistema host.

### **Sintaxis**

```
char* DSGetProjectList(void);
```

### **Valores de retorno**

Si la función es satisfactoria, el valor de retorno es un puntero a una conjunto de series terminadas en nulo, una para cada proyecto en el sistema principal, que acaban con un segundo carácter de nulo. El ejemplo siguiente muestra el contenido del almacenamiento intermedio donde <nulo> representa el carácter nulo de terminación:

```
proyecto1<nulo>proyecto2<nulo><nulo>
```

Si la función falla, el valor de retorno es NULO. Y la función **DSGetLastError** recupera el siguiente código de error:

**DSJE\_SERVER\_ERROR** Se ha producido un error de motor no esperado/desconocido.

### **Comentarios**

Se puede llamar a esta función antes que a cualquier otra función de la API de InfoSphere DataStage.

**Nota:** **DSGetProjectList** abre, utiliza y cierra su propio enlace de comunicaciones con el motor, de forma que es posible que tarde algún tiempo en recuperar la lista de proyectos.

## **DSGetReposInfo**

Utilice la función **DSGetReposInfo** para buscar objetos en tiempo de diseño.

### **Sintaxis**

```
int DSGetReposInfo (  
    DSPROJECTProyecto h,  
    int TipoObjeto,
```

```
int TipoInfo,
const char *CriteriosBúsqueda,
const char *CategoríaInicio,
int Subcategorías,
DSREPOSINFO *InfoRetorno);
```

## Parámetros

*proyctoh* es el valor devuelto de **DSOpenProject** para el proyecto cuyos trabajos desea buscar.

*TipoObjeto* actualmente se debe establecer en DSS\_JOBS para indicar que desea buscar trabajos.

*TipoInfo* es una de las claves siguientes:

**Esta clave...**

**Devuelve esta información...**

**DSS\_JOB\_ALL**

Lista todos los trabajos

**DSS\_JOB\_SERVER**

Lista todos los trabajos de servidor

**DSS\_JOB\_PARALLEL**

Lista todos los trabajos paralelos

**DSS\_JOB\_MAINFRAME**

Lista todos los trabajos de sistema principal

**DSS\_JOB\_SEQUENCE**

Lista todas las secuencias de trabajos

*CriteriosBúsqueda* es el nombre con el que se debe coincidir. Se puede utilizar la coincidencia parcial de nombres, con varios caracteres \* utilizados como comodines en cualquier lugar de la serie de búsqueda.

*CategoríaInicio* es la categoría en la que se iniciará la búsqueda. Si no se proporciona ninguna categoría, o se proporciona una serie NULA o vacía, se asumirá la categoría raíz.

*SubcategoríasBúsqueda* puede tener uno de estos dos valores: 1 (TRUE) y 0 (FALSE). Estos definen si la búsqueda incluirá subcategorías.

*InfoRetorno* es un puntero a una estructura que contiene la información de retorno necesaria (consulte "DSREPOSINFO").

## Valor de retorno

Si el resultado es satisfactorio, **DSGetReposInfo** devuelve el número de objetos que se han encontrado.

Si el resultado es anómalo, se devuelve un código de retorno como el siguiente:

- DSJE\_BADTYPE Los valores de *TipoObjeto* o *TipoInfo* no se han reconocido
- DSJE\_REPERERROR Se ha producido un error al intentar acceder al repositorio. Llame a **DSGetLastErrorMsg** para obtener el mensaje de error asociado con el código de error.

- DSJE\_NO\_DATASTAGE Parece que el proyecto conectado no es un proyecto válido de InfoSphere DataStage

## DSGetReposUsage

Utilice la función DSGetReposUsage para devolver una lista de objetos basados en la relación necesaria.

### Sintaxis

```
int DSGetReposUsage(
    DSPROJECT Proyecto,
    int TipoRelación,
    const char *NombreObjeto,
    int Recurso,
    DSREPOSUSAGE *InfoRetorno);
```

### Parámetros

*proyecto* es el valor devuelto de DSOpenProject para el proyecto cuyos trabajos desea buscar.

*TipoRelación* es una de las claves siguientes:

Esta clave...

Devuelve esta información...

**DSS\_JOB\_USES\_JOB**

Devuelve una lista de trabajos que utiliza el trabajo especificado.

**DSS\_JOB\_USEDDBY\_JOB**

Devuelve una lista de trabajos que utilizan el trabajo especificado.

**DSS\_JOB\_HASSOURCE\_DRSTABLE**

Devuelve una lista de trabajos que utilizan la tabla especificada como origen en una etapa DRS.

**DSS\_JOB\_HASTARGET\_DRSTABLE**

Devuelve una lista de trabajos que utilizan la tabla especificada como destino en una etapa DRS.

**DSS\_JOB\_HASSOURCEORTARGET\_DRSTABLE**

Devuelve una lista de trabajos que utilizan la tabla especificada como origen o destino de una etapa DRS.

*NombreObjeto* especifica el trabajo o tabla y varía según el *TipoRelación* que se especifica:

- para relaciones DSS\_JOB\_USES\_JOB y DSS\_JOB\_USEDDBY\_JOB, se debe proporcionar el nombre de trabajo (sin calificación de categoría).
- para las relaciones restantes, se debe proporcionar el nombre de tabla completamente calificado.
- Para las relaciones de definición de tabla de la etapa DRS, se permite la coincidencia parcial del nombre de tabla utilizando caracteres \* como comodines. Se pueden utilizar varios caracteres comodines

*Recurso* lo utilizan las relaciones DSS\_JOB\_USES\_JOB y DSS\_JOB\_USEDDBY\_JOB. Puede tener 2 valores, 1 (TRUE) y 0 (FALSE). Si se establece en TRUE, para cada trabajo encontrado que utiliza el *NombreObjeto*, se encuentran los trabajos en los que se utiliza ese trabajo y así sucesivamente. Para todos los demás tipos de relaciones se ignora el parámetro.

*InfoRetorno* es un puntero a una estructura que contiene los valores devueltos (consulte “DSREPOSUSAGE” en la página 93). El orden en el que los trabajos aparecen en la estructura *InfoRetorno* tal como define *TipoRelación*. Para el *TipoRelación* de DSS\_JOB\_USES\_JOB, los trabajos aparecerán en el orden en el que aparecen en la lista de dependencias de trabajos. Esta lista está en el separador **Dependencias** en el diálogo Propiedades de trabajo.

## Valor de retorno

Si el resultado es satisfactorio, **DSGetReposUsage** devuelve el número de objetos que se han encontrado.

Si el resultado es anómalo, se devuelve un código de retorno como el siguiente:

- **DSJE\_REPERERROR** Se ha producido un error al intentar acceder al repositorio. Llame a **DSGetLastErrorMsg** para obtener el mensaje de error asociado con el código de error.
- **DSJE\_NO\_DATASTAGE** Parece que el proyecto conectado no es un proyecto válido de InfoSphere DataStage.
- **DSJE\_UNKNOWN\_JOBNAME** Cuando el *TipoRelación* es DSS\_JOB\_USES\_JOB o DSS\_JOB\_USED\_BY\_JOB, el nombre de trabajo proporcionado no se puede encontrar en el proyecto.

## DSGetStageInfo

Utilice la función **DSGetStageInfo** para obtener información sobre una etapa determinada dentro de un trabajo.

### Sintaxis

```
int DSGetStageInfo(  
    DSJOB DescriptorContextoTrabajo,  
    char *NombreEtapa,  
    int TipoInfo,  
    DSSTAGEINFO *InfoRetorno  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NombreEtapa* es un puntero a una serie terminada en nulo que especifica el nombre de la etapa que se interrogará.

*TipoInfo* es una de las claves siguientes:

**Esta clave...**

**Devuelve esta información...**

**DSJ\_LINKLIST**

Lista de nombres de enlaces de la etapa, separados por nulos.

**DSJ\_STAGELASTERR**

Último mensaje de error notificado desde cualquier enlace de la etapa.

**DSJ\_STAGENAME**

Nombre de la etapa.

**DSJ\_STAGETYPE**

Nombre del tipo de etapa.

**DSJ\_STAGEINROWNUM**

Número de fila de entrada de enlaces primarios.

**DSJ\_VARLIST**

Lista de nombres de variables de la etapa, separados por nulos.

**DSJ\_STAGESTARTTIMESTAMP**

La fecha y hora a las que se inició la etapa.

**DSJ\_STAGEENDTIMESTAMP**

Fecha y hora a las que finalizó la etapa.

**DSJ\_STAGEDESC**

Descripción de la etapa (desde propiedades de etapa)

**DSJ\_STAGEINST**

Lista de ID de instancia (trabajos paralelos), separados por nulos.

**DSJ\_STAGECPU**

Lista de tiempos de CPU, en segundos.

**DSJ\_LINKTYPES**

Lista de tipos de enlace, separados por nulos.

**DSJ\_STAGEELAPSED**

Tiempo transcurrido, en segundos.

**DSJ\_STAGEPID**

Lista de ID de proceso, separados por nulos.

**DSJ\_STAGESTATUS**

Estado de la etapa.

**DSJ\_CUSTINFOLIST**

Lista de elementos de información de personalización separados por nulos.

*InfoRetorno* es un puntero a una estructura de datos **DSSTAGEINFO** donde se almacena la información solicitada. Consulte “Estructuras de datos” en la página 80.

**Valores de retorno**

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción****DSJE\_NOT\_AVAILABLE**

No hay instancias de la información solicitada en la etapa.

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_BADSTAGE**

*NombreEtapa* no hace referencia a una etapa conocida del trabajo.

**DSJE\_BADTYPE**

*TipoInfo* no válido.

**Comentarios**

Esta función se puede utilizar antes o después de que se haya emitido una función **DSRunJob**.

La estructura de datos **DSSTAGEINFO** contiene una **unión** con un elemento para cada uno de los posibles valores de retorno desde la llamada a **DSGetStageInfo**.

## DSGetVarInfo

Utilice la función **DSGetVarInfo** para obtener información sobre variables utilizadas en etapas Transformer.

### Sintaxis

```
int DSGetVarInfo(  
    DSJOB DescriptorContextoTrabajo,  
    char *NombreEtapa,  
    char *NombreVar,  
    int TipoInfo,  
    DSSTAGEINFO *InfoRetorno  
);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NombreEtapa* es un puntero a una serie terminada en nulo que especifica el nombre de la etapa que se interrogará.

*NombreVar* es un puntero a una serie terminada en nulo que especifica el nombre de la variable que se interrogará.

*TipoInfo* es una de las claves siguientes:

**Esta clave...**

**Devuelve esta información...**

**DSJ\_VARVALUE**

El valor de la variable especificada.

**DSJ\_VARDESC**

Descripción de la variable.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal   Descripción**

**DSJE\_NOT\_AVAILABLE**

No hay instancias de la información solicitada en la etapa.

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_BADSTAGE**

*NombreEtapa* no hace referencia a una etapa conocida del trabajo.

**DSJE\_BADVAR**

*NombreVar* no hace referencia a una variable conocida en el trabajo.

**DSJE\_BADTYPE**

*TipoInfo* no válido.

## DSListEnvVars

Utilice la función DSListEnvVars para obtener una lista de variables de entorno y sus valores en un proyecto especificado.

### Sintaxis

```
char *DSListEnvVars(  
    DSPROJECT proyectorh);
```

### Parámetro

*proyectorh* es el valor devuelto de **DSOpenProject** para el proyecto cuyas variables de entorno desea listar.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es un puntero a un conjunto de series terminadas en nulo, una para cada variable de entorno, que acaban con un segundo carácter nulo. Cada serie tiene el formato *NombreVarEnt=ValorVarEnt*.

Si la función falla, el valor de retorno es NULO y la función DSGetLastError se puede utilizar para recuperar un código de error de la forma siguiente:

- DSJE\_READENVVARDEFNS No se han podido leer las definiciones de variable de entorno
- DSJE\_READENVVARVALUES No se han podido leer los valores de variable de entorno
- DSJE.ISPARALLELLICENCED No ha podido determinar si hay trabajos paralelos disponibles

### Comentarios

Para utilizar este método, es necesario que el programa se haya conectado anteriormente a un proyecto utilizando **DSOpenProject**. Devuelve un descriptor de contexto al proyecto, *proyectorh*.

Las variables de entorno en la categoría paralela sólo se listarán si hay trabajos paralelos disponibles.

## DSListProjectProperties

Utilice la función DSListProjectProperties para obtener una lista de los valores de propiedades de proyecto para un proyecto especificado.

Las propiedades soportadas por la función DSListProjectProperties son:

- Si el OSH generado está visible en trabajos paralelos.
- Si la propagación de columnas en tiempo de ejecución está habilitada en trabajos paralelos.
- El nombre del directorio base para trabajos paralelos.
- Opciones avanzadas de tiempo de ejecución para trabajos paralelos.
- Mandatos de despliegue personalizados para trabajos paralelos.
- Directorio de plantillas de trabajo de despliegue.
- Si la administración de trabajos está habilitada en el cliente del Director.



## Sintaxis

```
char *DSListProjectProperties(  
    DSPROJECT proyectorh  
);
```

## Parámetro

*proyectorh* es el valor devuelto de **DSOpenProject** para el proyecto cuyas propiedades desea listar.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es un puntero a un conjunto de series terminadas en nulo, una para cada variable, que acaban con un segundo carácter de nulo. Cada serie tiene el formato *NombrePropiedad=ValorPropiedad* donde *NombrePropiedad* será uno de los siguientes:

Esta clave...

Indica esta propiedad...

### DSA\_OSHVISIBLEFLAG

El OSH generado es visible en trabajos paralelos. Sólo trabajos paralelos.

### DSA\_PRJ\_RTCP\_ENABLED

La propagación de columnas en tiempo de ejecución está habilitada en trabajos paralelos. Sólo trabajos paralelos.

### DSA\_PRJ\_PX\_ADVANCED\_RUNTIME\_OPTS

Especifica propiedades de tiempo de ejecución avanzadas para trabajos paralelos. Sólo trabajos paralelos.

### DSA\_PRJ\_PX\_BASEDIR

Especifica el directorio base para trabajos paralelos. Sólo trabajos paralelos.

### DSA\_PRJ\_PX\_DEPLOY\_JOBDIR\_TEMPLATE

Especifica la plantilla de directorio de despliegue para trabajos paralelos. Sólo trabajos paralelos.

### DSA\_PRJ\_PX\_DEPLOY\_CUSTOM\_ACTION

Especifica mandatos de despliegue personalizados para trabajos paralelos. *Valor* son los mandatos. Sólo trabajos paralelos.

### DSA\_PRJ\_JOBADMIN\_ENABLED

Los mandatos de administración de trabajos están habilitados en el cliente del Director para trabajos en este proyecto.

### DSA\_PRJ\_PX\_DEPLOY\_GENERATE\_XML

La generación de informes XML está habilitada para paquetes de despliegue de trabajos paralelos.

Estas señales están definidas en *dsapi.h* (consulte “El archivo de cabecera dsapi.h” en la página 37).

Si la función falla, el valor de retorno es NULO y la función **DSGetLastError** se puede utilizar para recuperar uno de los siguientes códigos de error:

- DSJE\_READPROJPROPERTY No se ha podido leer la propiedad
- DSJE\_ISPARALLELLICENCED No se ha podido determinar si los trabajos paralelos están disponibles
- DSJE\_OSHVISIBLEFLAG No se ha podido obtener el valor para OSHVisible

## Comentarios

Para utilizar este método, es necesario que el programa se haya conectado anteriormente a un proyecto utilizando **DSOpenProject**. Devuelve un descriptor de contexto al proyecto, *proyecto*h.

Si los trabajos paralelos no están disponibles, sólo se devolverá el valor de DSA\_PRJ\_JOBADMIN\_ENABLED.

## DSLockJob

Utilice la función DSLockJob para bloquear un trabajo. Debe llamar a esta función antes de establecer los parámetros de ejecución de un trabajo o de iniciar una ejecución del trabajo.

### Sintaxis

```
int DSLockJob(  
    DSJOB DescriptorContextoTrabajo  
);
```

### Parámetro

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

DSJE_BADHANDLE	<i>DescriptorContextoTrabajo</i> no válido.
----------------	---------------------------------------------

## Comentarios

El bloqueo de un trabajo impide que cualquier otro trabajo modifique los detalles o estado del trabajo. Se debe llamar a esta función antes de cualquier llamada de **DSSetJobLimit**, **DSSetParam** o **DSRunJob**.

Si intenta bloquear un trabajo que ya ha bloqueado, la llamada será satisfactoria. Si tiene el mismo trabajo abierto en varios descriptores de contexto de la API de InfoSphere DataStage, el bloqueo del trabajo en un descriptor de contexto bloqueará el trabajo en todos los descriptores de contexto.

## DSLogEvent

Utilice la función DSLogEvent para añadir una nueva entrada a un archivo de registro de trabajo.

### Sintaxis

```
int DSLogEvent(  
    DSJOB DescriptorContextoTrabajo,  
    int TipoSuceso,  
    char *Reservado,  
    char *Mensaje  
);
```

## Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*TipoSuceso* es una de las claves siguientes que especifica el tipo de suceso que se registrará:

**Esta clave...**

**Especifica este tipo de suceso...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

*Reservado* está reservado para uso futuro y se debe especificar como nulo.

*Mensaje* apunta a una serie de caracteres terminada en nulo que especifica el texto del mensaje que se registrará.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción**

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_SERVER\_ERROR**

Error interno. El motor ha devuelto datos no válidos.

**DSJE\_BADTYPE**

Valor de *TipoSuceso* no válido.

## Comentarios

Los mensajes que contienen más de una línea de texto deben contener un carácter de nueva línea (\n) para indicar el fin de una línea.

## DSMakeJobReport

Utilice la función **DSMakeJobReport** para generar un informe que describa el estado completo de un trabajo conectadoválido.

### Sintaxis

```
int DSMakeJobReport(  
    DSJOB DescriptorContextoTrabajo,  
    int TipoInforme,  
    char *SeparadorLíneas,  
    DSREPORTINFO *InfoRetorno);
```

## Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*TipoInforme* es uno de los valores siguientes que especifican el tipo de informe que se generará:

**Este valor...**

**Especifica este tipo de informe...**

- 0 Básico, esta serie de texto contiene la hora de inicio/finalización, el tiempo transcurrido y el estado del trabajo.
- 1 Detalle de etapa/enlace. Como el informe básico, pero también contiene información sobre etapas individuales y enlaces dentro del trabajo.
- 2 Serie de texto que contiene informe XML completo.

*SeparadorLineas* apunta a una serie de caracteres terminada en nulo que especifica el separador de línea en el informe. Los valores especiales reconocidos son:

"CRLF" => CHAR(13):CHAR(10)

"LF" => CHAR(10)

"CR" => CHAR(13)

El valor predeterminado es CRLF si se trata de Windows, de lo contrario LF.

**Valores de retorno**

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

## DSOpenJob

Utilice la función DSOpenJob para abrir un trabajo. Debe llamar a esta función antes de que cualquier otra función manipule el trabajo

**Sintaxis**

```
DSJOB DSOpenJob(  
    DSPROJECT DescriptorContextoProyecto,  
    char *NombreTrabajo  
);
```

**Parámetros**

*DescriptorContextoProyecto* es el valor devuelto de **DSOpenProject**.

*NombreTrabajo* es un puntero a una serie terminada en nulo que especifica el nombre del trabajo que se abrirá. Puede estar en uno de los formatos siguientes:

*trabajo* Encuentra la versión más reciente del trabajo.

*trabajo%Reln.n.n*

Busca un release determinado del trabajo en un sistema de desarrollo.

**Valores de retorno**

Si la función es satisfactoria, el valor de retorno es un descriptor de contexto del trabajo.

Si la función falla, el valor de retorno es NULO. Utilice **DSGetLastError** para recuperar uno de los valores siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_OPENFAIL</b>	
----------------------	--

	El motor no ha podido abrir el trabajo.
--	-----------------------------------------

<b>DSJE_NO_MEMORY</b>	
-----------------------	--

	Anomalía de asignación de memoria.
--	------------------------------------

## Comentarios

La función **DSOpenJob** se debe utilizar para devolver un descriptor de contexto de trabajo antes de poder direccionar un trabajo mediante cualquiera de las funciones de la API de InfoSphere DataStage. Puede obtener acceso exclusivo al trabajo bloqueándolo con **DSLockJob**.

Se puede abrir el mismo trabajo más de una vez y cada llamada a **DSOpenJob** devolverá un descriptor de contexto de trabajo exclusivo. Cada descriptor de contexto se debe cerrar por separado.

## DSOpenProject

Utilice la función **DSOpenProject** para abrir un proyecto. Debe llamar a esta función antes que cualquier otra función de la API de InfoSphere DataStage, excepto la función **DSGetProjectList** o la función **DSGetLastError**.

### Sintaxis

```
DSPROJECT DSOpenProject(  
    char *NombreProyecto  
);
```

### Parámetro

*NombreProyecto* es un puntero a una serie terminada en nulo que especifica el nombre del proyecto que se abrirá.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es un descriptor de contexto del proyecto.

Si la función falla, el valor de retorno es NULO. Utilice **DSGetLastError** para recuperar uno de los valores siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_BAD_VERSION</b>	
-------------------------	--

	El motor es una versión más antigua que la API de InfoSphere DataStage.
--	-------------------------------------------------------------------------

<b>DSJE_INCOMPATIBLE_SERVER</b>	
---------------------------------	--

	El motor es más antiguo o más nuevo que el que está soportado por esta versión de la API de InfoSphere DataStage.
--	-------------------------------------------------------------------------------------------------------------------

<b>DSJE_SERVER_ERROR</b>	
--------------------------	--

	Error interno. El motor ha devuelto datos no válidos.
--	-------------------------------------------------------

<b>DSJE_BADPROJECT</b>	
------------------------	--

	Nombre de proyecto no válido.
--	-------------------------------

<b>DSJE_NO_DATASTAGE</b>	
--------------------------	--

	InfoSphere DataStage no está instalado correctamente en el host de la capa de motor.
--	--------------------------------------------------------------------------------------

## Comentarios

La función **DSGetProjectList** puede devolver el nombre de un proyecto que no contenga trabajos válidos de InfoSphere DataStage, pero esto se detecta cuando se llama a **DSOpenProject**. Un proceso sólo puede tener un proyecto abierto cada vez.

## DSRunJob

Utilice la función **DSRunJob** para iniciar una ejecución del trabajo.

### Sintaxis

```
int DSRunJob(    DSJOB DescriptorContextotrabajo,  
                int ModalidadEjecución  
);
```

### Parámetros

*DescriptorContextoTrabajo* es un valor devuelto de **DSOpenJob**.

*ModalidadEjecución* es una clave para determinar la modalidad de ejecución y debe ser uno de los valores siguientes:

#### DSJ\_RUNNORMAL

Inicia la ejecución de un trabajo.

#### DSJ\_RUNRESET

Restablece el trabajo.

#### DSJ\_RUNVALIDATE

Valida el trabajo.

#### DSJ\_RUNRESTART

Reiniciar una secuencia de trabajos reinicialable con los valores de parámetros del trabajo original.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es una de las siguientes señales:

#### DSJE\_BADHANDLE

*DescriptorContextoTrabajo* no válido.

#### DSJE\_BADSTATE

El trabajo no está en el estado correcto (debe estar compilado, no en ejecución).

#### DSJE\_BADTYPE

*ModalidadRetorno* no se reconoce.

#### DSJE\_SERVER\_ERROR

Error interno. El motor ha devuelto datos no válidos.

## Comentarios

El trabajo especificado por *DescriptorContextoTrabajo* se debe bloquear mediante **DSLockJob** antes de llamar a la función **DSRunJob**.

Si no se establecieron límites al llamar a **DSSetJobLimit**, se utilizarán los límites predeterminados.

## DSSetEnvVar

Utilice la función `DSSetEnvVar` para establecer el valor para una variable de entorno en un proyecto especificado.

### Sintaxis

```
int DSSetEnvVar(  
    DSPROJECT Proyecto,  
    char *NombreVarEnt,  
    char *Valor  
);
```

### Parámetros

*proyecto* es el valor devuelto de `DSOpenProject`.

*NombreVarEnt* es el nombre de la variable de entorno.

*Valor* es el valor en el que se establecerá la variable de entorno.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es `DSJE_NOERROR`

Si la función falla, el valor de retorno es uno de los siguientes:

- `DSJE_READENVVARDEFNS` No se han podido leer las definiciones de variable de entorno
- `DSJE_READENVVARVALUES` No se han podido leer los valores de variable de entorno
- `DSJE_BADENVVAR` La variable de entorno no existe
- `DSJE_WRITEENVVARVALUES` No se han podido grabar los valores de variable de entorno
- `DSJE_ENCODEFAILED` No se ha podido codificar un valor cifrado
- `DSJE_BADBOOLEANVALUE` Valor no válido proporcionado para una variable de entorno booleana
- `DSJE_BADNUMERICVALUE` Valor no válido proporcionado para una variable de entorno de entero
- `DSJE_BADLISTVALUE` Valor no válido proporcionado para una variable de entorno con una lista fija de valores
- `DSJE_PXNOTINSTALLED` La variable de entorno es específica de trabajos paralelos que no están disponibles
- `DSJE_ISPARALLELLICENCED` No se ha podido determinar si los trabajos paralelos están disponibles

### Comentarios

Sólo puede establecer valores para variables de entorno en la categoría paralela si hay disponibles trabajos paralelos.

Si está estableciendo una variable de entorno de tipo lista (por ejemplo, `APT_EXECUTION_MODE`), debe establecerla en uno de los valores internos que se permiten, en lugar de uno de los miembros de la lista como se muestran en el cliente del Administrador. Por ejemplo, si quisiera establecer `APT_EXECUTION_MODE` de forma que los trabajos paralelos se ejecutaran en una

única modalidad de proceso, establecería el valor de la variable de entorno en `ONE\_PROCESS`, no `Un proceso` como se ofrece en el cliente de Administración.

Si está estableciendo una variable de entorno de tipo booleano, establezca el valor en 1 para TRUE y en 0 para FALSE.

## DSSetGenerateOpMetaData

Utilice la función DSSetGenerateOpMetaData para especificar si el trabajo genera o no metadatos operativos. Esta función altera temporalmente el valor predeterminado del proyecto.

### Sintaxis

```
int DSSetGenerateOpMetaData (  
    DescriptorContextoTrabajo,  
    valor  
);
```

### Parámetros

*DescriptorContextoTrabajo* es un valor devuelto de **DSOpenJob**.

*valor* es TRUE (1) para generar metadatos operativos, FALSE (0) para no generar metadatos operativos.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

DSJE_BADHANDLE	
----------------	--

*DescriptorContextoTrabajo* no válido.

DSJE_BADTYPE	
--------------	--

*valor* no se reconoce.

## DSSetJobLimit

Utilice la función DSSetJobLimit para establecer límites de fila o de aviso para un trabajo.

### Sintaxis

```
int DSSetJobLimit(  
    DSJOB DescriptorContextoTrabajo,  
    int TipoLímite,  
    int ValorLímite  
);
```

### Parámetros

*DescriptorContextoTrabajo* es un valor devuelto de **DSOpenJob**.

*TipoLímite* es una de las claves siguientes que especifica el tipo de límite:

Esta clave...	Especifica este tipo de límite...
---------------	-----------------------------------



### DSJ\_LIMITWARN

Trabajo que se detendrá después de *ValorLímite* sucesos de aviso.

### DSJ\_LIMITROWS

Etapas que se limitarán después de *ValorLímite* filas.

*ValorLímite* es el valor en el que se establecerá el límite.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_BADHANDLE</b>	
-----------------------	--

	<i>DescriptorContextoTrabajo</i> no válido.
--	---------------------------------------------

<b>DSJE_BADSTATE</b>	
----------------------	--

	El trabajo no está en el estado correcto (compilado, no en ejecución).
--	------------------------------------------------------------------------

<b>DSJE_BADTYPE</b>	
---------------------	--

	<i>TipoLímite</i> no es el nombre de una condición de limitación conocida.
--	----------------------------------------------------------------------------

<b>DSJE_BADVALUE</b>	
----------------------	--

	<i>ValorLímite</i> no es adecuado para el tipo de condición de limitación.
--	----------------------------------------------------------------------------

<b>DSJE_SERVER_ERROR</b>	
--------------------------	--

	Error interno. El motor ha devuelto datos no válidos.
--	-------------------------------------------------------

## Comentarios

El trabajo especificado por *DescriptorContextoTrabajo* se debe bloquear mediante **DSLockJob** antes de llamar a la función **DSSetJobLimit**.

Los límites de trabajo que no se establecen explícitamente antes de una ejecución utilizarán los valores predeterminados. Realice dos llamadas a **DSSetJobLimit** para establecer ambos tipos de límite.

Establezca el valor en 0 para indicar que no debe haber límite para el trabajo.

## DSSetParam

Utilice la función DSSetParam para establecer los valores de parámetro de trabajo antes de ejecutar un trabajo. Cualquier parámetro que no se establece explícitamente utiliza el valor predeterminado.

### Sintaxis

```
int DSSetParam(  
    DSJOB DescriptorContextoTrabajo,  
    char *NombreParám,  
    DSPARAM *Parám);
```

### Parámetros

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

*NombreParám* es un puntero a una serie terminada en nulo que especifica el nombre del parámetro que se establecerá.

*Parám* es un puntero a una estructura que especifica el nombre, tipo y valor del parámetro que se establecerá.

**Nota:** Es necesario que el tipo especificado en *Parám* no coincida con el tipo especificado para el parámetro en la definición de trabajo, pero debe ser posible convertirlo. Por ejemplo, si el trabajo define el parámetro como una serie, se puede establecer especificándolo como un entero. Sin embargo, causará un error con resultados imprevisibles si el parámetro se define en el trabajo como un entero y **DSSetParam** pasa una serie no numérica.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es **DSJE\_NOERROR**.

Si la función falla, el valor de retorno es uno de los siguientes:

**Señal Descripción**

**DSJE\_BADHANDLE**

*DescriptorContextoTrabajo* no válido.

**DSJE\_BADSTATE**

El trabajo no está en el estado correcto (compilado, no en ejecución).

**DSJE\_BADPARAM**

*Parám* no hace referencia a un parámetro conocido del trabajo.

**DSJE\_BADTYPE**

*Parám* no especifica un tipo de parámetro válido.

**DSJE\_BADVALUE**

*Parám* no especifica un valor adecuado para el tipo de parámetro tal como se especifica en la definición del trabajo.

**DSJE\_SERVER\_ERROR**

Error interno. El motor ha devuelto datos no válidos.

## Comentarios

El trabajo especificado por *DescriptorContextotrabajo* se debe bloquear mediante **DSLockJob** antes de llamar a la función **DSSetParam**.

## DSSetProjectProperty

Utilice la función **DSSetProjectProperty** para establecer el valor de una propiedad en un proyecto especificado. El usuario que ejecuta el código que contiene esta función debe ser un administrador de InfoSphere DataStage.

### Sintaxis

```
int DSSetProjectProperty(  
    DSPROJECT Proyectoh,  
    char *Propiedad,  
    char *Valor  
);
```

### Parámetros

*Proyecto*h es el valor devuelto de **DSOpenProject**

*Propiedad* es el nombre de la propiedad que se establecerá. Se da soporte a las propiedades siguientes:

**Esta clave...**

**Indica esta propiedad...**

**DSA\_OSHVISIBLEFLAG**

El OSH generado está visible en trabajos paralelos. *Valor* es 0 para false o 1 para true. Sólo trabajos paralelos.

**DSA\_PRJ\_RTCP\_ENABLED**

La propagación de columnas en tiempo de ejecución está habilitada en trabajos paralelos. *Valor* es 0 para false o 1 para true. Sólo trabajos paralelos.

**DSA\_PRJ\_PX\_ADVANCED\_RUNTIME\_OPTS**

Especifica propiedades avanzadas de tiempo de ejecución para trabajos paralelos. *Valor* son las propiedades avanzadas que se establecerán. Sólo trabajos paralelos.

**DSA\_PRJ\_PX\_BASEDIR**

Especifica el directorio base para trabajos paralelos. *Valor* es el directorio base. Sólo trabajos paralelos.

**DSA\_PRJ\_PX\_DEPLOY\_JOBDIR\_TEMPLATE**

Especifica la plantilla de directorio de despliegue para trabajos paralelos. *Valor* es la plantilla de directorio de despliegue. Sólo trabajos paralelos.

**DSA\_PRJ\_PX\_DEPLOY\_CUSTOM\_ACTION**

Especifica mandatos de despliegue personalizados para trabajos paralelos. *Valor* son los mandatos. Sólo trabajos paralelos.

**DSA\_PRJ\_JOBADMIN\_ENABLED**

Los mandatos de administración de trabajos están habilitados en el cliente del Director para trabajos en este proyecto. *Valor* es 0 para false o 1 para true.

**DSA\_PRJ\_PX\_DEPLOY\_GENERATE\_XML**

La generación de informes XML está habilitada para paquetes de despliegue de trabajos paralelos. *Valor* es 0 para false o 1 para true.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR

Si la función falla, el valor de retorno es uno de los siguientes:

- DSJE\_NOTADMINUSER El usuario no es administrador
- DSJE\_ISADMINFAILED No se ha podido determinar si el usuario es administrador
- DSJE\_READPROJPROPERTY No se ha podido leer la propiedad
- DSJE\_WRITEPROJPROPERTY No se ha podido grabar la propiedad
- DSJE\_PROPNOTSUPPORTED Propiedad no soportada
- DSJE\_BADPROPERTY Nombre de propiedad desconocido
- DSJE\_BADPROPVALUE Valor no válido para esta propiedad
- DSJE\_PXNOTINSTALLED Trabajos paralelos no disponibles
- DSJE\_ISPARALLELLICENCED No se ha podido determinar si los trabajos paralelos están disponibles
- DSJE\_OSHVISIBLEFLAG No se ha podido establecer el valor para OSHVisible

## Comentarios

Para utilizar este método, es necesario que el programa se haya conectado anteriormente a un proyecto utilizando **DSEOpenProject**. Devuelve un descriptor de contexto al proyecto, *projectoh*.

## DSSetServerParams

Utilice la función DSSetServerParams para establecer los parámetros de conexión que se utilizan para abrir un proyecto o recuperar una lista de proyectos.

### Sintaxis

```
void DSSetServerParams(  
    char *NombreDominio,  
    char *NombreUsuario,  
    char *Contraseña,  
    char *NombreServidor  
);
```

### Parámetros

*NombreDominio* es un puntero a una serie de caracteres terminada en nulo que especifica el nombre de la máquina de dominio.

*NombreUsuario* es un puntero a una serie de caracteres terminada en nulo que especifica el nombre de usuario que se utilizará para la sesión, o NULL.

*Contraseña* es un puntero a una serie de caracteres terminada en nulo que especifica la contraseña para el usuario especificado en *NombreUsuario*, o NULL.

*NombreServidor* es un puntero a una serie de caracteres terminada en nulo que especifica el nombre del servidor del nivel de motor al que se conectará, o NULL.

### Valores de retorno

Esta función no tiene ningún valor de retorno.

## Comentarios

De manera predeterminada, **DSEOpenProject** y **DSEGetProjectList** intentan conectar a un motor en el mismo sistema que el proceso de cliente y, a continuación, crear un proceso de motor que se ejecuta con la misma identificación de usuario y derechos de acceso que el proceso cliente. **DSSetServerParams** altera temporalmente este comportamiento y permite especificar un dominio, nombre de usuario, contraseña, y servidor del nivel de motor distintos.

Las llamadas a **DSSetServerParams** no son acumulativas. Todos los valores de parámetro, incluyendo los punteros NULOS, se utilizan para establecer los parámetros que se utilizarán en la llamada subsiguiente a **DSEOpenProject** o **DSEGetProjectList**.

## DSEStopJob

Utilice DSEStopJob para cancelar un trabajo en ejecución.

## Sintaxis

```
int DSStopJob(  
    DSJOB DescriptorContextoTrabajo  
);
```

## Parámetro

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es:

DSJE\_BADHANDLE *DescriptorContextoTrabajo* no válido.

## Comentarios

La función **DSStopJob** se debe utilizar sólo después de haber emitido una función **DSRunJob**. Se envía la solicitud de detención independientemente del estado actual del trabajo. Para averiguar si se ha detenido el trabajo, utilice la función **DSWaitForJob** en la macro **DSJobStatus**.

## DSUnlockJob

Utilice la función **DSUnlockJob** para desbloquear un trabajo, impidiendo que se realice cualquier otra manipulación del estado de ejecución del trabajo y liberándolo para que otros procesos lo utilicen.

## Sintaxis

```
int DSUnlockJob(  
    DSJOB DescriptorContextoTrabajo  
);
```

## Parámetro

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

## Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJ\_NOERROR.

Si la función falla, el valor de retorno es:

- DSJE\_BADHANDLE *DescriptorContextoTrabajo* no válido.

## Comentarios

La función **DSUnlockJob** devuelve el valor de retorno inmediatamente sin esperar la finalización del trabajo. Si se intenta desbloquear un trabajo que no está bloqueado, no se producirá ningún error. Si tiene abierto el mismo trabajo en varios descriptores de contexto y se desbloquea el trabajo en un descriptor de contexto, se desbloqueará en todos los descriptores de contexto.

## DSWaitForJob

Utilice la función DSWaitForJob para esperar hasta que finalice la ejecución de un trabajo.

### Sintaxis

```
int DSWaitForJob(  
    DSJOB DescriptorContextoTrabajo  
);
```

### Parámetro

*DescriptorContextoTrabajo* es el valor devuelto de **DSOpenJob**.

### Valores de retorno

Si la función es satisfactoria, el valor de retorno es DSJE\_NOERROR.

Si la función falla, el valor de retorno es uno de los siguientes:

Señal	Descripción
-------	-------------

<b>DSJE_BADHANDLE</b>
-----------------------

<i>DescriptorContextoTrabajo no válido.</i>
---------------------------------------------

<b>DSJE_WRONGJOB</b>
----------------------

El proceso actual no ha iniciado el trabajo para este <i>DescriptorContextoTrabajo</i> desde una llamada a <b>DSRunJob</b> .
---------------------------------------------------------------------------------------------------------------------------------

<b>DSJE_TIMEOUT</b>
---------------------

Parece que el trabajo no se ha iniciado después de esperar durante un periodo razonable de tiempo. (Aproximadamente 30 minutos.)
-------------------------------------------------------------------------------------------------------------------------------------

### Comentarios

Esta función sólo es válida si el trabajo actual ha emitido una llamada a **DSRunJob** en el *DescriptorContextoTrabajo* proporcionado. Realiza el retorno si el trabajo se ha iniciado desde el último **DSRunJob** y ha finalizado desde entonces. El estado de finalización se puede encontrar llamando a **DSGetJobInfo**.

## Estructuras de datos

Utilice las estructuras de datos para mantener los datos pasados a funciones o devueltos por ellas.

La API de InfoSphere DataStage utiliza las estructuras de datos que se describen en esta sección para mantener datos pasados a funciones o devueltos por ellas. (Consulte el apartado “Estructuras de datos, datos de resultados y hebras” en la página 38). Las estructuras de datos se resumen a continuación, con descripciones completas de las secciones siguientes:

Esta estructura de datos...	Mantiene este tipo de datos...	Y la utiliza esta función...
DSCUSTINFO	Elementos de información de personalización de determinados tipos de etapa paralela	DSGetCustinfo
DSJOBINFO	Información sobre un trabajo	DSGetJobInfo

Esta estructura de datos...	Mantiene este tipo de datos...	Y la utiliza esta función...
DSLINKINFO	Información sobre un enlace a o de una etapa activa en un trabajo, es decir, una etapa que no es un origen de datos o destino	DSGetLinkInfo
DSLOGDETAIL	Detalles completos de una entrada en un archivo de registro de trabajo	DSGetLogEntry
DSLOGDETAILFULL	Detalles completos de una entrada en un archivo de registro de trabajo, incluidos el ID de mensaje y el ID de invocación	DSGetLogEntryFull
DSLOGEVENT	Detalles de una entrada en un archivo de registro de trabajo	DSLogEvent, DSFindFirstLogEntry, DSFindNextLogEntry
DSPARAM	El tipo y valor de un parámetro de trabajo	DSSetParam
DSPARAMINFO	Información adicional sobre un parámetro de trabajo, como por ejemplo su valor predeterminado y una descripción	DSGetParamInfo
DSPROJECTINFO	Lista de trabajos en el proyecto	DSGetProjectInfo
DSREPOSINFO	Lista de trabajos de tiempo de diseño	DSGetReposInfo
DSREPOSUSGE	Lista de trabajos de tiempo de diseño que cumplen una relación	DSGetReposUsage
DSSTAGEINFO	Información sobre una etapa en un trabajo	DSGetStageInfo
DSVARINFO	Información sobre variables de etapa en etapas Transformer	DSGetVarInfo

## DSCUSTINFO

Utilice la estructura **DSCUSTINFO** para representar diversos valores de información sobre un enlace con o desde una etapa activa en un trabajo de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSCUSTINFO {
    int tipoInfo: /
    union {
        char *custinfoValue;
        char *custinfoDesc;} info;
} DSCUSTINFO;
```

## Miembros

*TipoInfo* es una clave que indica el tipo de información y es uno de los valores siguientes:

**Esta clave...**

**Indica esta información...**

**DSJ\_CUSTINFOVALUE**

El valor del elemento de información de personalización especificado.

**DSJ\_CUSTINFODESC**

La descripción del elemento de información de personalización especificado.

## DSJOBINFO

Utilice la estructura DSJOBINFO para representar valores de información sobre un trabajo de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSJOBINFO {
    int tipoInfo;
    union {
        int estadoTrabajo;
        char *controladorTrabajo;
        time_t tipoHoraTrabajo;
        int númeroOndaTrabajo;
        char *estadoUsuario;
        char *listaParám;
        char *listaEtapa;
        char *jobname;
        int jobcontrol;
        int jobPid;
        time_t jobLastTime;
        char *jobInvocations;
        int jobInterimStatus;
        char *jobInvocationid;
        char *jobDesc;
        char *stageList2;
        char *jobElapsed;
        char *jobFullDesc;
        int jobDMIService;
        int jobMultiInvokable;
    } info;
} DSJOBINFO;
```

## Miembros

*tipoInfo* es una de las claves siguientes que indican el tipo de información:

**Esta clave...**

**Indica esta información...**

**DSJ\_JOBSTATUS**

El estado actual del trabajo.

**DSJ\_JOBNAME**

Nombre de trabajo al que hace referencia *DescripciónContextoTrabajo*

**DSJ\_JOBCONTROLLER**

El nombre del trabajo de control.



**DSJ\_JOBSTARTTIMESTAMP**

La fecha y hora a las que se inició el trabajo.

**DSJ\_JOBWAVENO**

Número de ondas de la ejecución de trabajo actual (o última).

**DSJ\_PARAMLIST**

Una lista de los nombres de los parámetros del trabajo. Separados por nulos.

**DSJ\_STAGELIST**

Una lista de etapas activas en el trabajo. Separados por nulos.

**DSJ\_USERSTATUS**

El estado notificado por el propio trabajo tal como está definido en el diseño del trabajo.

**DSJ\_JOBCONTROL**

Si se ha emitido una solicitud de detención para el trabajo.

**DSJ\_JOBPID**

ID de proceso del proceso DSD.RUN.

**DSJ\_JOBLASTTIMESTAMP**

La fecha y hora del motor a las que ha finalizado el último trabajo.

**DSJ\_JOBINVOCATIONS**

Lista de ID de invocación de trabajo. Separados por nulos.

**DSJ\_JOBINTERIMSTATUS**

Estado temporal actual del trabajo.

**DSJ\_JOBINVOCATIONID**

Nombre de invocación del trabajo al que se hace referencia.

**DSJ\_JOBDESC**

Una descripción del trabajo.

**DSJ\_STAGELIST2**

Una lista de etapas pasivas en el trabajo. Separados por nulos.

**DSJ\_JOBELAPSED**

El tiempo transcurrido del trabajo en segundos.

**DSJ\_JOBFULLDESSC**

La descripción completa especificada en el recuadro de diálogo Propiedades del trabajo.

**DSJ\_JOBDMISERVICE**

Se establece en true (verdadero) si se trata de un trabajo de servicio web.

**DSJ\_JOBMULTIINVOKABLE**

Se establece en true (verdadero) si este trabajo da soporte a varias invocaciones.

*estadoTrabajo* se devuelve cuando *tipoInfo* se establece en DSJ\_JOBSTATUS. Su valor es una de las claves siguientes:

**Esta clave...**

**Indica este estado...**

**DSJS\_RUNNING**

Trabajo en ejecución.

**DSJS\_RUNOK**

El trabajo ha finalizado una ejecución normal sin avisos.

**DSJS\_RUNWARN**

El trabajo ha finalizado una ejecución normal con avisos.

**DSJS\_RUNFAILED**

El trabajo ha finalizado una ejecución normal con un error muy grave.

**DSJS\_VALOK**

El trabajo ha finalizado una ejecución de validación sin avisos.

**DSJS\_VALWARN**

El trabajo ha finalizado una ejecución de validación sin avisos.

**DSJS\_VALFAILED**

El trabajo ha fallado una ejecución de validación.

**DSJS\_RESET**

El trabajo ha finalizado una ejecución de restablecimiento.

**DSJS\_CRASHED**

El trabajo ha sido detenido por alguna acción indeterminada.

**DSJS\_STOPPED**

El trabajo se ha detenido por la intervención del operador (no se puede determinar el tipo de ejecución).

**DSJS\_NOTRUNNABLE**

El trabajo no se ha compilado.

**DSJS\_NOTRUNNING**

Cualquier otro estado. El trabajo se ha detenido por la intervención del operador (no se puede determinar el tipo de ejecución).

*controladorTrabajo* es el nombre del trabajo que controla la referencia de trabajo y se devuelve cuando *TipoInfo* se establece en DSJ\_JOBCONTROLLER. Tenga en cuenta este que puede ser varios nombres de trabajo, separados por puntos, si el trabajo está controlado por un trabajo que a su vez está controlado, y así sucesivamente.

*horaIniciotrabajo* es la fecha y hora a la que se inició la ejecución del trabajo actual o último y se devuelve cuando *TipoInfo* se establece en DSJ\_JOBSTARTTIMESTAMP.

*númeroOnda* es el número de onda de la ejecución de trabajo última o actual y se devuelve cuando *TipoInfo* se establece en DSJ\_JOBWAVENO.

*estadoUsuario* es el valor, si existe, establecido como su estado definido por el usuario y se devuelve cuando *TipoInfo* esta establecido en DSJ\_USERSTATUS.

*listaParám* es un puntero a un almacenamiento intermedio que contiene un conjunto de series terminadas en nulos, una para cada nombre de parámetro de trabajo, que acaban con un segundo carácter nulo. Se devuelve cuando *TipoInfo* se establece en DSJ\_PARAMLIST. El ejemplo siguiente muestra el contenido del almacenamiento intermedio donde <nulo> representa el carácter nulo de terminación:

primero<nulo>segundo<nulo><nulo>

*listaEtapas* es un puntero a un almacenamiento intermedio que contiene un conjunto de series terminadas en nulos, una para cada etapa del trabajo, que acaban con un segundo carácter nulo. Se devuelve cuando *TipoInfo* se establece en

DSJ\_STAGELIST. El ejemplo siguiente muestra el contenido del almacenamiento intermedio donde <nulo> representa el carácter nulo de terminación:  
primero<nulo>segundo<nulo><nulo>

## DSLINKINFO

La estructura de datos DSLINKINFO representa diversos valores de información sobre un enlace con o desde una etapa activa en un trabajo de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSLINKINFO {  
    int tipoInfo: /*  
    union {  
        DSLOGDETAIL últimoError;  
        int númFila;  
        char *nombreEnlace;  
        char *estadoSQLenlace;  
        char *códigoDBMSenlace;  
        char *DescEnlace;  
        char *EtapaEnlazada;  
        char *listaNúmFilas;  
    } info;  
} DSLINKINFO;
```

### Miembros

*TipoInfo* es una clave que indica el tipo de información y es uno de los valores siguientes:

**Esta clave...**

**Indica esta información...**

**DSJ\_LINKLASTERR**

El último mensaje de error notificado desde un enlace.

**DSJ\_LINKNAME**

Enlace real del enlace.

**DSJ\_LINKROWCOUNT**

El número de filas que se han pasado a un enlace.

**DSJ\_LINKSQLSTATE**

Valor de SQLSTATE del último mensaje de error.

**DSJ\_LINKDBMSCODE**

Valor de DBMSCODE del último mensaje de error.

**DSJ\_LINKDESC**

Descripción del enlace.

**DSJ\_LINKSTAGE**

Nombre de la etapa en el otro extremo del enlace.

**DSJ\_INSTROWCOUNT**

Lista de recuentos de fila, separados por coma, uno por instancia (trabajos paralelos)

*últimoError* es una estructura de datos que contiene la entrada de registro de error para el último mensaje de error notificado de un enlace y se devuelve cuando *TipoInfo* se establece en DSJ\_LINKLASTERR.

*NúmFilas* es el número de filas que se han pasado a un enlace hasta ahora y se devuelve cuando *TipoInfo* está establecido en `DSJ_LINKROWCOUNT`.

## DSLOGDETAIL

La estructura **DSLOGDETAIL** representa información detallada para una única entrada para el archivo de registro de un trabajo.

### Sintaxis

```
typedef struct _DSLOGDETAIL {  
    int IDSuceso;  
    time_t indicación_fecha_y_hora;  
    int tipo;  
    char *nombre_usuario;  
    char *mensajeCompleto;  
} DSLOGDETAIL;
```

### Miembros

*IDSuceso* es un número, 0 o superior, que identifica de forma exclusiva la entrada de registro para el trabajo.

*indicación\_fecha\_y\_hora* es la fecha y hora a la que se ha añadido la entrada al archivo de registro del trabajo.

*tipo* es una clave que indica el tipo del suceso y es uno de los valores siguientes:

**Esta clave...**

**Indica este tipo de entrada de registro...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

**DSJ\_LOGFATAL**

Error muy grave

**DSJ\_LOGREJECT**

Rechazo de fila de Transformer

**DSJ\_LOGSTARTED**

Trabajo iniciado

**DSJ\_LOGRESET**

Trabajo restablecido

**DSJ\_LOGBATCH**

Control de proceso por lotes

**DSJ\_LOGOTHER**

Cualquier otro tipo de entrada de registro

*nombre\_usuario* es la información de usuario.

*mensajeCompleto* es la descripción completa de la entrada de registro.

## DSLOGDETAILFULL

La estructura **DSLOGDETAILFULL** representa información detallada para una única entrada de un archivo de registro de trabajo, incluidos el ID de mensaje y el ID de invocación.

### Sintaxis

```
typedef struct _DSLOGDETAILFULL {  
    int IDSuceso;  
    time_t indicación_fecha_y_hora;  
    int tipo;  
    char *nombre_usuario;  
    char *mensajeCompleto;  
    char *Idmensaje;  
    char *Idinvocación;  
} DSLOGDETAILFULL;
```

### Miembros

*IDSuceso* es un número, 0 o superior, que identifica de forma exclusiva la entrada de registro para el trabajo.

*indicación\_fecha\_y\_hora* es la fecha y hora a la que se ha añadido la entrada al archivo de registro del trabajo.

*tipo* es una clave que indica el tipo del suceso y es uno de los valores siguientes:

**Esta clave...**

**Indica este tipo de entrada de registro...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

**DSJ\_LOGFATAL**

Error muy grave

**DSJ\_LOGREJECT**

Rechazo de fila de Transformer

**DSJ\_LOGSTARTED**

Trabajo iniciado

**DSJ\_LOGRESET**

Trabajo restablecido

**DSJ\_LOGBATCH**

Control de proceso por lotes

**DSJ\_LOGOTHER**

Cualquier otro tipo de entrada de registro

*nombre\_usuario* es la información de usuario.

*mensajeCompleto* es la descripción completa de la entrada de registro.

*Idmensaje* es una serie que contiene el ID de mensaje.

*Idinvocación* es un serie que contiene el ID de invocación, si se ha utilizado.

## DSLOGEVENT

Utilice la estructura DSLOGEVENT para representar la información de resumen para una única entrada del registro de sucesos de un trabajo.

### Sintaxis

```
typedef struct _DSLOGEVENT {  
    int IDSuceso;  
    time_t indicación_fecha_y_hora;  
    int tipo;  
    char *mensaje;  
} DSLOGEVENT;
```

### Miembros

*IDSuceso* es un número, 0 o superior, que identifica de forma exclusiva la entrada de registro para el trabajo.

*indicación\_fecha\_y\_hora* es la fecha y hora a la que se ha añadido la entrada al archivo de registro del trabajo.

*tipo* es una clave que indica el tipo del suceso y es uno de los valores siguientes:

**Esta clave...**

**Indica este tipo de entrada de registro...**

**DSJ\_LOGINFO**

Información

**DSJ\_LOGWARNING**

Aviso

**DSJ\_LOGFATAL**

Error muy grave

**DSJ\_LOGREJECT**

Rechazo de fila de Transformer

**DSJ\_LOGSTARTED**

Trabajo iniciado

**DSJ\_LOGRESET**

Trabajo restablecido

**DSJ\_LOGBATCH**

Control de proceso por lotes

**DSJ\_LOGOTHER**

Cualquier otro tipo de entrada de registro

*mensaje* es la primera línea de la descripción de la entrada de registro.

## DSPARAM

Utilice la estructura DSPARAM para representar información sobre el tipo y el valor de un parámetro de trabajo de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSPARAM {  
    int tipoParám;  
    union {  
        char *seriep;  
        char *cifradop;
```

```

int entp;
float flotantep;
char *víaaccesop;
char *valorListap;
char *fechap;
char *horap;
} valorParám;
} DSPARAM;

```

## Miembros

*tipoParám* es una clave que especifica el tipo de parámetro de trabajo. Los valores posibles son los siguientes:

**Esta clave...**

**Indica este tipo de parámetro...**

**DSJ\_PARAMTYPE\_STRING**

Una serie de caracteres.

**DSJ\_PARAMTYPE\_ENCRYPTED**

Una serie de caracteres cifrados (por ejemplo, una contraseña).

**DSJ\_PARAMTYPE\_INTEGER**

Un entero.

**DSJ\_PARAMTYPE\_FLOAT**

Un número de coma flotante.

**DSJ\_PARAMTYPE\_PATHNAME**

Un nombre de vía de acceso del sistema de archivos.

**DDSJ\_PARAMTYPE\_LIST**

Una serie de caracteres que especifica uno de los valores de una lista numerada.

**DDSJ\_PARAMTYPE\_DATE**

Fecha con el formato AAAA-MM-DD.

**DSJ\_PARAMTYPE\_TIME**

Hora con el formato HH:MM:SS.

*seriep* es una serie de caracteres terminada en nulo que se devuelve cuando se establece *tipoParám* en DSJ\_PARAMTYPE\_STRING.

*cifradop* es una serie de caracteres terminada en nulo que se devuelve cuando *tipoParám* se establece en DSJ\_PARAMTYPE\_ENCRYPTED. La serie debe estar en texto sin formato cuando se pasa a o desde la API de InfoSphere DataStage donde se cifra. La aplicación que utiliza la API de InfoSphere DataStage debe presentar este tipo de parámetro en un formato de visualización adecuado, por ejemplo, un asterisco para cada carácter de la serie en lugar del propio carácter.

*Entp* es un entero y se devuelve cuando *tipoParám* se establece en DSJ\_PARAMTYPE\_INTEGER.

*flotantep* es un número de punto flotante y se devuelve cuando *tipoParám* se establece en DSJ\_PARAMTYPE\_FLOAT.

*víaaccesop* es una serie de caracteres terminada en nulo que especifica un nombre de vía de acceso del sistema de archivos y se devuelve cuando *tipoParám* se establece a DSJ\_PARAMTYPE\_PATHNAME.

**Nota:** No es necesario que este parámetro especifique un nombre de vía de acceso válido en el motor. La interpretación y validación del nombre de vía de acceso la realiza el trabajo.

*valorListap* es una serie de caracteres terminada en nulo que especifica uno de los posibles valores de una lista enumerada y se devuelve cuando *tipoParám* se establece en DDSJ\_PARAMTYPE\_LIST.

*fechap* es una serie de caracteres terminada en nulo que especifica una fecha en el formato AAAA-MM-DD y se devuelve cuando *tipoParám* se establece en DSJ\_PARAMTYPE\_DATE.

*tipop* es una serie de caracteres terminada en nulo que especifica una hora con el formato HH:MM:SS y se devuelve cuando *tipoParám* se establece en DSJ\_PARAMTYPE\_TIME.

## DSPARAMINFO

Utilice la estructura DSPARAMINFO para representar diversos valores de información sobre un parámetro de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSPARAMINFO {
    DSPARAM valorPredeterminado;
    char *TextoAyuda;
    char *SolicitudParám;
    int tipoParám;
    DSPARAM ValorPredeterminadoDis;
    char *valoresLista;
    char *valoresListaDis;
    int solicitudALEjecutar;
} DSPARAMINFO;
```

### Miembros

*valorPredeterminado* es el valor predeterminado, si existe, para el parámetro.

*textoAyuda* es una descripción, si existe, para el parámetro.

*solicitudParám* es la solicitud, si existe, para el parámetro.

*tipoParám* es una clave que especifica el tipo de parámetro de trabajo. Los valores posibles son los siguientes:

**Esta clave...**

**Indica este tipo de parámetro...**

**DSJ\_PARAMTYPE\_STRING**

Una serie de caracteres.

**DSJ\_PARAMTYPE\_ENCRYPTED**

Una serie de caracteres cifrados (por ejemplo, una contraseña).

**DSJ\_PARAMTYPE\_INTEGER**

Un entero.

**DSJ\_PARAMTYPE\_FLOAT**

Un número de coma flotante.

**DSJ\_PARAMTYPE\_PATHNAME**

Un nombre de vía de acceso del sistema de archivos.



### DDSJ\_PARAMTYPE\_LIST

Una serie de caracteres que especifica uno de los valores de una lista numerada.

### DDSJ\_PARAMTYPE\_DATE

Fecha con el formato *AAAA-MM-DD*.

### DSJ\_PARAMTYPE\_TIME

Hora con el formato *HH:MM:SS*.

*ValorPredeterminadoDis* es el valor predeterminado establecido para el parámetro por el diseñador del trabajo.

**Nota:** Los valores predeterminados los puede cambiar el administrador de InfoSphere DataStage, así que es posible que un valor no sea el valor actual para el trabajo.

*listaValores* es un puntero a un almacenamiento intermedio que recibe un conjunto de series terminadas en nulo, una para cada serie que se puede utilizar como el valor del parámetro, y que acaban con un segundo carácter nulo como se muestra en el ejemplo siguiente (<nulo> representa el carácter nulo de terminación):

primero<nulo>segundo<nulo><nulo>

*ValoresListaDis* es un puntero a un almacenamiento intermedio que contiene la lista predeterminada de valores establecida para el parámetro por el diseñador del trabajo. El almacenamiento intermedio contiene un conjunto de series terminadas en nulos, una para cada serie válida que se pueda utilizar como valor del parámetro, que finaliza con un segundo carácter nulo. El ejemplo siguiente muestra el contenido del almacenamiento intermedio donde <nulo> representa el carácter nulo de terminación:

primero<nulo>segundo<nulo><nulo>

**Nota:** Los valores predeterminados los puede cambiar el administrador de InfoSphere DataStage, así que es posible que un valor no sea el valor actual para el trabajo.

*SolicitudALEjecutar* es 0 (**False**) o 1 (**True**). 1 indica que se solicita al operador un valor para este parámetro cada vez que se ejecuta el trabajo; 0 indica que no se solicita.

## DSPROJECTINFO

Utilice la estructura DSPROJECTINFO para representar valores de información para un proyecto de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSPROJECTINFO {
    int TipoInfo;
    union {
        char *listaTrabajo;
    } info;
} DSPROJECTINFO;
```

### Miembros

*TipoInfo* es un valor clave que indica el tipo de información a recuperar. Los valores posibles son los siguientes:

**Esta clave...**

**Indica esta información...**

**DSJ\_JOBLIST**

Lista de trabajos en el proyecto.

**DSJ\_PROJECTNAME**

Nombre del proyecto actual.

**DSJ\_HOSTNAME**

Nombre de host de la capa de motor.

*listaTrabajos* es un puntero a un almacenamiento intermedio que contiene un conjunto de series terminadas en nulo, una para cada trabajo en el proyecto, y que acaban con un segundo carácter nulo, como se muestra en el ejemplo siguiente (<nulo> representa el carácter nulo de terminación):

primero<nulo>segundo<nulo><nulo>

## DSREPOSINFO

Utilice la estructura DSREPOSINFO para obtener información sobre objetos en tiempo de diseño que se han buscado.

### Sintaxis

```
struct _DSREPOSJOBINFO;
typedef struct _DSREPOSJOBINFO DSREPOSJOBINFO;
struct _DSREPOSJOBINFO
{
    char* nombretrabajo; /* Incluye categoría */
    int tipotrabajo; /* Constante TipoInfo */
    DSREPOSJOBINFO* trabajosiguiente; /* puntero siguiente trabajo o NULO */
};
typedef struct _DSREPOSINFO
{
    int TipoInfo;
    union
    {
        DSREPOSJOBINFO* trabajos; /*lista enlazada de trabajos encontrados */
    } info;
} DSREPOSINFO;
```

### Miembros

*TipoInfo* es un valor clave que indica el tipo de información a recuperar. Los valores posibles son los siguientes:

**Esta clave...**

**Indica esta información...**

**DSS\_JOBS**

Lista de trabajos.

*trabajos* es un puntero a una estructura enlazada a otra estructura o terminada con un nulo. Existe una estructura para cada trabajo devuelto. Cada estructura contiene el nombre de trabajo (incluyendo categoría) y el tipo de trabajo de la forma siguiente:

**Esta clave...**

**Devuelve esta información...**

**DSS\_JOB\_SERVER**

Trabajo del servidor

**DSS\_JOB\_PARALLEL**  
Trabajo paralelo

**DSS\_JOB\_MAINFRAME**  
Trabajo de sistema principal

**DSS\_JOB\_SEQUENCE**  
Secuencia de trabajos

## DSREPOSUSAGE

Utilice la estructura DSREPOSUSAGE para obtener información sobre los objetos que cumplen una relación especificada.

### Sintaxis

```
struct _DSREPOSUSAGEJOB; typedef struct _DSREPOSUSAGEJOB
DSREPOSUSAGEJOB; struct _DSREPOSUSAGEJOB { char *nombretrabajo; /*
Trabajo y nombre cat */ int tipotrabajo; /* tipo del trabajo */ DSREPOSUSAGEJOB
*trabajosiguiente; /* siguiente trabajo hermano */ DSREPOSUSAGEJOB *trabajohijo;
}; typedef struct _DSREPOSUSAGE { int TipoInfo; union { DSREPOSUSAGEJOB
*trabajos; /*lista enlazada de trabajos*/ } info } DSREPOSUSAGE;
```

### Miembros

*TipoInfo* es un valor clave que indica el tipo de información a recuperar. Los valores posibles son los siguientes:

**Esta clave...**

**Indica esta información...**

**DSS\_JOBS**

Lista de trabajos.

*trabajos* es un puntero a una estructura enlazada a otra estructura o terminada con un nulo. Existe una estructura para cada trabajo devuelto. Cada estructura contiene el nombre de trabajo (incluyendo categoría) y el tipo de trabajo de la forma siguiente:

**Esta clave...**

**Devuelve esta información...**

**DSS\_JOB\_SERVER**

Trabajo del servidor

**DSS\_JOB\_PARALLEL**

Trabajo paralelo

**DSS\_JOB\_MAINFRAME**

Trabajo de sistema principal

**DSS\_JOB\_SEQUENCE**

Secuencia de trabajos

## DSSTAGEINFO

Utilice la estructura DSSTAGEINFO para representar diversos valores de información sobre una etapa activa dentro de un trabajo de InfoSphere DataStage.

## Sintaxis

```
typedef struct _DSSTAGEINFO {
    int TipoInfo;
    union {
        DSLOGDETAIL últimoError;
        char *nombreTipo;
        int NúmFilaEn;
        char *listaEnlaces;
        char *stagename;
        char *varlist;
        char *stageStartTime;
        char *stageEndTime;
        char *linkTypes;
        char *stageDesc;
        char *instList;
        char *cpuList;
        time_t stageElapsed;
        char *pidList;
        int stageStatus;
        char *custInfoList
    } info;
} DSSTAGEINFO;
```

## Miembros

*TipoInfo* es una clave que indica la información que se devolverá y es uno de los valores siguientes:

**Esta clave...**

**Indica esta información...**

**DSJ\_LINKLIST**

Lista de nombres de enlace, separados por nulos.

**DSJ\_STAGELASTERR**

El último mensaje de error generado desde cualquier enlace de la etapa.

**DSJ\_STAGENAME**

Nombre de etapa.

**DSJ\_STAGETYPE**

El nombre de tipo de etapa, por ejemplo, Transformer o BeforeJob.

**DSJ\_STAGEINROWNUM**

El número de fila de entrada del enlace primario.

**DSJ\_VARLIST**

Lista de nombres de variables de la etapa.

**DSJ\_STAGESTARTTIME-STAMP**

La fecha y hora a las que se inició la etapa.

**DSJ\_STAGEENDTIME-STAMP**

Fecha y hora a las que finalizó la etapa.

**DSJ\_STAGEDESC**

Descripción de la etapa (desde propiedades de etapa)

**DSJ\_STAGEINST**

Lista de ID de instancia (trabajos paralelos), separados por nulos.

**DSJ\_STAGECPU**

Lista de tiempos de CPU, separados por coma, en segundos.

**DSJ\_LINKTYPES**

Lista de tipos de enlace, separados por nulos.

## DSJ\_STAGEELAPSED

Tiempo transcurrido, en segundos.

## DSJ\_STAGEPID

Lista de ID de proceso, separados por nulos.

## DSJ\_STAGESTATUS

Estado de la etapa.

## DSJ\_CUSTINFOLIST

Lista de nombres de elementos de información de personalización, separados por nulos.

*últimoError* es una estructura de datos que contiene el mensaje de error para el último error (si existe) notificado de cualquier enlace de la etapa. Se devuelve cuando *TipoInfo* se establece en DSJ\_STAGELASTERR.

*nombreTipo* es el nombre de tipo de etapa y se devuelve cuando *TipoInfo* se establece en DSJ\_STAGETYPE.

*NúmFilaEn* es el número de fila de entrada del enlace primario y se devuelve cuando *TipoInfo* está establecido en DSJ\_STAGEINROWNUM.

*listaEnlaces* es un puntero a un almacenamiento intermedio que contiene un conjunto de series terminadas en nulo, una para cada enlace de la etapa, y que acaban con un segundo carácter nulo, como se muestra en el ejemplo siguiente (<nulo> representa el carácter nulo de terminación):

primero<nulo>segundo<nulo><nulo>

## DSVARINFO

Utilice la estructura DSVARINFO para representar diversos valores de información sobre un enlace con o desde una etapa activa en un trabajo de InfoSphere DataStage.

### Sintaxis

```
typedef struct _DSVARINFO {  
    int tipoInfo: 1  
    union {  
        char *varValue;  
        char *varDesc;  
    } info;  
} DSVARINFO;
```

### Miembros

*TipoInfo* es una clave que indica el tipo de información y es uno de los valores siguientes:

**Esta clave...**

**Indica esta información...**

## DSJ\_VARVALUE

El valor de la variable especificada.

## DSJ\_VARDESC

La descripción de la variable especificada.

## Códigos de error

Utilice los códigos de error para determinar los problemas de la aplicación.

La tabla siguiente lista los códigos de error de la API de InfoSphere DataStage en orden alfabético:

Tabla 2. Códigos de error de la API

Señal de error	Código	Descripción
DSJE_ACCOUNTPATHFAILED	-140	No se ha podido obtener la vía de acceso de cuenta.
DSJE_ADDPROJECTBLOCKED	-134	Otro usuario está añadiendo un proyecto.
DSJE_ADDPROJECTFAILED	-135	No se ha podido añadir el proyecto.
DSJEBADBOOLEANVALUE	-118	Valor no válido proporcionado para una variable de entorno booleana.
DSJE_BADENVVAR	-116	La variable de entorno no existe.
DSJE_BADENVVARNAME	-108	Nombre de variable de entorno no válido.
DSJE_BADENVVARTYPE	-109	Tipo de variable de entorno no válido.
DSJE_BADENVVARPROMPT	-110	No se ha proporcionado ninguna solicitud.
DSJE_BADHANDLE	-1	<i>DescriptorContextoTrabajo</i> no válido.
DSJE_BADLINK	-9	<i>NombreEnlace</i> no hace referencia a un enlace conocido para la etapa en cuestión.
DSJE_BADLISTVALUE	-120	Valor no válido proporcionado para una variable de entorno de lista.
DSJE_BADNAME	-12	Nombre de proyecto no válido.
DSJE_BADNUMERICVALUE	-119	Valor no válido proporcionado para una variable de entorno de entero.
DSJE_BADPARAM	-3	<i>NombreParám</i> no es un nombre de parámetro en el trabajo.
DSJE_BADPROJECT	-1002	<i>NombreProyecto</i> no es un proyecto conocido de InfoSphere DataStage.
DSJE_BADPROJLOCATION	-130	Se ha proporcionado un nombre de vía de acceso no válido.
DSJE_BADPROJNAME	-128	Se ha proporcionado un nombre de proyecto no válido.
DSJE_BADPROPERTY	-104	Nombre de propiedad desconocido.
DSJE_BADPROPVALUE	-106	Valor no válido para esta propiedad.
DSJE_BADSTAGE	-7	<i>NombreEtapa</i> no hace referencia a una etapa conocida del trabajo.
DSJE_BADSTATE	-2	El trabajo no está en el estado correcto (compilado, no en ejecución).
DSJE_BADTIME	-13	Valor de <i>HoraInicio</i> u <i>HoraFinalización</i> no válido.
DSJE_BADTYPE	-5	No se ha reconocido el tipo de suceso o información.
DSJE_BAD_VERSION	-1008	El motor no soporta esta versión de la API de InfoSphere DataStage.
DSJE_BADVALUE	-4	Valor de <i>NúmeroMáx</i> no válido.

Tabla 2. Códigos de error de la API (continuación)

Señal de error	Código	Descripción
DSJE_CLEARSCCHEDULEFAILED	-127	No se han podido deseleccionar los trabajos planificados para el proyecto.
DSJE_DECRYPTERR	-15	No se han podido descifrar los valores cifrados.
DSJE_DELETEPROJECTBLOCKED	-138	Proyecto bloqueado por otro usuario.
DSJE_DELPROJFAILED	-124	No se ha podido suprimir la definición de proyecto.
DSJE_DELPROJFILESFAILED	-125	No se han podido suprimir los archivos de proyecto.
DSJE_DUPENVVARIABLE	-115	La variable de entorno que se está añadiendo ya existe.
DSJE_ENCODEFAILED	-123	No se ha podido codificar un valor cifrado.
DSJE_GETDEFAULTPATHFAILED	-129	No se ha podido determinar el directorio de proyecto predeterminado.
DSJE_INCOMPATIBLE_SERVER	-1009	La versión del motor no es compatible con esta versión de la API de InfoSphere DataStage.
DSJE_ISADMINFAILED	-101	No se ha podido determinar si el usuario es administrador.
DSJE_ISPARALLELLICENSED	-122	No se ha podido determinar si hay trabajos paralelos disponibles.
DSJE_INVALIDPROJECTLOCATION	-131	Se ha proporcionado un nombre de vía de acceso no válido.
DSJE_JOBDELETED	-11	El trabajo se ha suprimido.
DSJE_JOBLOCKED	-10	El trabajo está bloqueado por otro proceso.
DSJE_LICENSEPROJECTFAILED	-136	No se ha podido otorgar licencia al proyecto.
DSJE_LISTSCCHEDULEFAILED	-126	No se ha podido obtener la lista de trabajos planificados para el proyecto.
DSJE_LOGTOFAILED	-141	No se ha podido iniciar sesión en la cuenta UV.
DSJE_NOACCESS	-16	No se pueden obtener valores, valores predeterminados o valores predeterminados de diseño para ningún trabajo excepto el trabajo actual.
DSJE_NO_DATASTAGE	-1003	InfoSphere DataStage no está instalado en el sistema.
DSJE_NOERROR	0	No se ha producido ningún error de la API de InfoSphere DataStage.
DSJE_NO_MEMORY	-1005	No se ha podido asignar memoria dinámica.
DSJE_NOMORE	-1001	Se han devuelto todos los sucesos que coinciden con los criterios de filtro.

Tabla 2. Códigos de error de la API (continuación)

Señal de error	Código	Descripción
DSJE_NOTADMINUSER	-100	El usuario no es administrador.
DSJE_NOTAPROJECT	-139	No se ha podido iniciar sesión en el proyecto.
DSJE_NOT_AVAILABLE	-1007	No se ha encontrado la información solicitada.
DSJE_NOTINSTAGE	-8	Error del motor interno.
DSJE_NOTUSERDEFINED	-117	La variable de entorno no está definida por el usuario y por lo tanto no se puede suprimir.
DSJE_OPENFAIL	-1004	El intento de abrir el archivo ha fallado - quizás no se ha compilado.
DSJE_OPENFAILED	-132	No se ha podido abrir el archivo UV.ACCOUNT.
DSJE_OSHVISIBLEFLAG	-107	No se ha podido obtener el valor para OSHVisible.
DSJE_PROPNOTSUPPORTED	-105	Propiedad no soportada.
DSJE_PXNOTINSTALLED	-121	La variable de entorno es específica para trabajos paralelos que no están disponibles.
DSJE_READENVVARDEFNS	-111	No se han podido leer las definiciones de variable de entorno.
DSJE_READENVVARVALUES	-112	No se han podido leer los valores de variable de entorno.
DSJE_READPROJPROPERTY	-102	No se ha podido leer la propiedad.
DSJE_READUFAILED	-133	No se ha podido bloquear el registro de bloqueo de creación de proyecto.
DSJE_RELEASEFAILED	-137	No se ha podido liberar el registro de bloqueo de creación de proyecto.
DSJE_REPERERROR	-99	Error del motor general.
DSJE_SERVER_ERROR	-1006	Se ha producido un error inesperado o desconocido en el motor.
DSJE_TIMEOUT	-14	Parece que el trabajo no se ha iniciado después de esperar un periodo de tiempo razonable. (Aproximadamente 30 minutos.)
DSJE_UNKNOWN_JOBNAME	-201	El nombre de trabajo proporcionado no se puede encontrar en el proyecto.
DSJE_WRITEENVVARDEFNS	-113	No se han podido grabar las definiciones de variable de entorno.
DSJE_WRITEENVVARVALUES	-114	No se han podido grabar los valores de variable de entorno.
DSJE_WRITEPROJPROPERTY	-103	Propiedad no soportada.
DSJE_WRONGJOB	-6	El proceso actual no ha iniciado el trabajo para este <i>DescriptorContextoTrabajo</i> desde una llamada a <b>DSRunJob</b> .



La tabla siguiente lista los códigos de error de la API de InfoSphere DataStage en orden numérico:

Tabla 3. Códigos de error de la API en orden numérico

Código	Señal de error	Descripción
0	DSJE_NOERROR	No se ha producido ningún error de la API de InfoSphere DataStage.
-1	DSJE_BADHANDLE	<i>DescriptorContextoTrabajo</i> no válido.
-2	DSJE_BADSTATE	El trabajo no está en el estado correcto (compilado, no en ejecución).
-3	DSJE_BADPARAM	<i>NombreParám</i> no es un nombre de parámetro en el trabajo.
-4	DSJE_BADVALUE	Valor de <i>NúmeroMáx</i> no válido.
-5	DSJE_BADTYPE	No se ha reconocido el tipo de suceso o información.
-6	DSJE_WRONGJOB	El proceso actual no ha iniciado el trabajo para este <i>DescriptorContextoTrabajo</i> desde una llamada a <b>DSRunJob</b> .
-7	DSJE_BADSTAGE	<i>NombreEtapa</i> no hace referencia a una etapa conocida del trabajo.
-8	DSJE_NOTINSTAGE	Error del motor interno.
-9	DSJE_BADLINK	<i>NombreEnlace</i> no hace referencia a un enlace conocido para la etapa en cuestión.
-10	DSJE_JOBLOCKED	El trabajo está bloqueado por otro proceso.
-11	DSJE_JOBDELETED	El trabajo se ha suprimido.
-12	DSJE_BADNAME	Nombre de proyecto no válido.
-13	DSJE_BADTIME	Valor de <i>HoraInicio</i> u <i>HoraFinalización</i> no válido.
-14	DSJE_TIMEOUT	Parece que el trabajo no se ha iniciado después de esperar un periodo de tiempo razonable. (Aproximadamente 30 minutos.)
-15	DSJE_DECRYPTERR	No se han podido descifrar los valores cifrados.
-16	DSJE_NOACCESS	No se pueden obtener valores, valores predeterminados o valores predeterminados de diseño para ningún trabajo excepto el trabajo actual.
-99	DSJE_REPERROR	Error del motor general.
-100	DSJE_NOTADMINUSER	El usuario no es administrador.
-101	DSJE_ISADMINFAILED	No se ha podido determinar si el usuario es administrador.
-102	DSJE_READPROJPROPERTY	No se ha podido leer la propiedad.
-103	DSJE_WRITEPROJPROPERTY	Propiedad no soportada.
-104	DSJE_BADPROPERTY	Nombre de propiedad desconocido.

Tabla 3. Códigos de error de la API en orden numérico (continuación)

Código	Señal de error	Descripción
-105	DSJE_PROPNOTSUPPORTED	Propiedad no soportada.
-106	DSJE_BADPROPVALUE	Valor no válido para esta propiedad.
-107	DSJE_OSHVISIBLEFLAG	No se ha podido obtener el valor para OSHVisible.
-108	DSJE_BADENVVARNAME	Nombre de variable de entorno no válido.
-109	DSJE_BADENVVARTYPE	Tipo de variable de entorno no válido.
-110	DSJE_BADENVVARPROMPT	No se ha proporcionado ninguna solicitud.
-111	DSJE_READENVVARDEFNS	No se han podido leer las definiciones de variable de entorno.
-112	DSJE_READENVVARVALUES	No se han podido leer los valores de variable de entorno.
-113	DSJE_WRITEENVVARDEFNS	No se han podido grabar las definiciones de variable de entorno.
-114	DSJE_WRITEENVVARVALUES	No se han podido grabar los valores de variable de entorno.
-115	DSJE_DUPENVVARNAME	La variable de entorno que se está añadiendo ya existe.
-116	DSJE_BADENVVAR	La variable de entorno no existe.
-117	DSJE_NOTUSERDEFINED	La variable de entorno no está definida por el usuario y por lo tanto no se puede suprimir.
-118	DSJE_BADBOOLEANVALUE	Valor no válido proporcionado para una variable de entorno booleana.
-119	DSJE_BADNUMERICVALUE	Valor no válido proporcionado para una variable de entorno de entero.
-120	DSJE_BADLISTVALUE	Valor no válido proporcionado para una variable de entorno de lista.
-121	DSJE_PXNOTINSTALLED	La variable de entorno es específica para trabajos paralelos que no están disponibles.
-122	DSJE_ISPARALLELLICENCED	No se ha podido determinar si hay trabajos paralelos disponibles.
-123	DSJE_ENCODEFAILED	No se ha podido codificar un valor cifrado.
-124	DSJE_DELPROJFAILED	No se ha podido suprimir la definición de proyecto.
-125	DSJE_DELPROJFILESFAILED	No se han podido suprimir los archivos de proyecto.
-126	DSJE_LISTSCHEDULEFAILED	No se ha podido obtener la lista de trabajos planificados para el proyecto.
-127	DSJE_CLEARSCCHEDULEFAILED	No se han podido deseleccionar los trabajos planificados para el proyecto.
-128	DSJE_BADPROJNAME	Se ha proporcionado un nombre de proyecto no válido.

Tabla 3. Códigos de error de la API en orden numérico (continuación)

Código	Señal de error	Descripción
-129	DSJE_GETDEFAULTPATHFAILED	No se ha podido determinar el directorio de proyecto predeterminado.
-130	DSJE_BADPROJLOCATION	Se ha proporcionado un nombre de vía de acceso no válido.
-131	DSJE_INVALIDPROJECTLOCATION	Se ha proporcionado un nombre de vía de acceso no válido.
-132	DSJE_OPENFAILED	No se ha podido abrir el archivo UV.ACCOUNT.
-133	DSJE_READUFAILED	No se ha podido bloquear el registro de bloqueo de creación de proyecto.
-134	DSJE_ADDPROJECTBLOCKED	Otro usuario está añadiendo un proyecto.
-135	DSJE_ADDPROJECTFAILED	No se ha podido añadir el proyecto.
-136	DSJE_LICENSEPROJECTFAILED	No se ha podido otorgar licencia al proyecto.
-137	DSJE_RELEASEFAILED	No se ha podido liberar el registro de bloqueo de creación de proyecto.
-138	DSJE_DELETEPROJECTBLOCKED	Proyecto bloqueado por otro usuario.
-139	DSJE_NOTAPROJECT	No se ha podido iniciar sesión en el proyecto.
-140	DSJE_ACCOUNTPATHFAILED	No se ha podido obtener la vía de acceso de cuenta.
-141	DSJE_LOGTOFAILED	No se ha podido iniciar sesión en la cuenta UV.
-201	DSJE_UNKNOWN_JOBNAME	El nombre de trabajo proporcionado no se puede encontrar en el proyecto.
-1001	DSJE_NOMORE	Se han devuelto todos los sucesos que coinciden con los criterios de filtro.
-1002	DSJE_BADPROJECT	<i>NombreProyecto</i> no es un proyecto conocido de InfoSphere DataStage.
-1003	DSJE_NO_DATASTAGE	InfoSphere DataStage no está instalado en el sistema.
-1004	DSJE_OPENFAIL	El intento de abrir el archivo ha fallado - quizás no se ha compilado.
-1005	DSJE_NO_MEMORY	No se ha podido asignar memoria dinámica.
-1006	DSJE_SERVER_ERROR	Se ha producido un error inesperado o desconocido en el motor.
-1007	DSJE_NOT_AVAILABLE	No se ha encontrado la información solicitada.
-1008	DSJE_BAD_VERSION	El motor no soporta esta versión de la API de InfoSphere DataStage.
-1009	DSJE_INCOMPATIBLE_SERVER	La versión del motor no es compatible con esta versión de la API de InfoSphere DataStage.

La tabla siguiente lista algunos errores comunes que se pueden devolver de las capas de comunicaciones de nivel inferior:

*Tabla 4. Códigos de error de capa de comunicaciones de la API*

Número de error	Descripción
39121	La licencia de InfoSphere DataStage ha caducado.
39134	Se ha alcanzado el límite de usuarios de InfoSphere DataStage.
80011	Se ha proporcionado un nombre de sistema incorrecto o nombre de usuario o contraseña no válidos.
80019	La contraseña ha caducado.

## Interfaz de BASIC de InfoSphere DataStage

Utilice las funciones BASIC de InfoSphere DataStage para realizar diversas tareas.

Estas funciones se pueden utilizar en una rutina de control de trabajo, que se define como parte de las propiedades de un trabajo y permite que otros trabajos se ejecuten y sean controlados desde el primer trabajo. Algunas de estas funciones también se pueden utilizar para obtener información de estado sobre el trabajo actual; son útiles en expresiones de etapa activa y en subrutinas anteriores y posteriores a la etapa.

*Tabla 5. Funciones de BASIC*

Para hacer esto...	Utilice esto...
Especificar el trabajo que desee controlar	DSAttachJob
Establecer parámetros para el trabajo que desee controlar	DSSetParam
Establecer límites para el trabajo que desee controlar	DSSetJobLimit
Solicitar que se ejecute un trabajo	DSRunJob
Esperar a que finalice un trabajo al que se ha llamado	DSWaitForJob
Obtener información de determinadas etapas paralelas.	DSGetCustInfo
Obtener información sobre el proyecto actual	DSGetProjectInfo
Obtener información sobre el trabajo controlado o trabajo actual	DSGetJobInfo
Obtener información sobre una etapa en el trabajo controlado o trabajo actual	DSGetStageInfo
Obtener información sobre un enlace en un trabajo controlado o trabajo actual	DSGetLinkInfo
Obtener información sobre los parámetros de un trabajo controlado	DSGetParamInfo
Obtener el suceso de registro del registro de trabajos	DSGetLogEntry
Obtener el suceso de registro del registro de trabajo, incluidos el ID de mensaje y el ID de invocación	DSGetLogEntryFull

Tabla 5. Funciones de BASIC (continuación)

Para hacer esto...	Utilice esto...
Obtener una lista de ID de suceso de registro para una ejecución determinada de la invocación de un trabajo	DSGetLogEventIds
Obtener una serie de sucesos de registro sobre el tema especificado del registro de sucesos	DSGetLogSummary
Obtener el suceso de registro más reciente, de un tipo especificado, del registro de trabajos	DSGetNewestLogId
Registrar un suceso en el registro de trabajo de un trabajo distinto	DSLogEvent
Detener un trabajo controlado	DSStopJob
Devolver un descriptor de contexto de trabajo obtenido anteriormente de <b>DSAttachJob</b>	DSDetachJob
Registrar un mensaje de error muy grave en el archivo de registro de un trabajo y terminar anormalmente el trabajo.	DSLogFatal
Registrar un mensaje de información en el archivo de registro de un trabajo.	DSLogInfo
Poner un mensaje de información en el registro de trabajo de un trabajo actual de control de trabajo.	DSLogToController
Registrar un mensaje de aviso en el archivo de registro de un trabajo.	DSLogWarn
Generar una serie que describe el estado actual de un trabajo conectado válido.	DSMakeJobReport
Insertar argumentos en la plantilla de mensaje.	DSMakeMsg
Asegurar que un trabajo está en el estado correcto para ser ejecutado y validado.	DSPrepareJob
Interactuar con el recurso de envío de correo del sistema.	DSSendMail
Registrar un mensaje de aviso en el archivo de registro de un trabajo.	DSTransformError
Convertir un estado de control de trabajo o código de error en un mensaje de texto explicativo.	DSTranslateCode
Suspender un trabajo hasta que exista o no exista un archivo nombrado.	DSWaitForFile
Comprobar si está catalogada una rutina de BASIC, en el VOC como elemento invocable o en el espacio de catálogo.	DSCheckRoutine
Ejecuta un mandato del motor o de DOS desde una subrutina anterior/posterior.	DSExecute
Establece un mensaje de estado que devolverá un trabajo como mensaje de terminación cuando finalice	DSSetUserStatus

Tabla 5. Funciones de BASIC (continuación)

Para hacer esto...	Utilice esto...
Especifica si un trabajo genera metadatos operativos cuando se ejecuta. Altera temporalmente el valor predeterminado para el proyecto.	DSSetGenerateOpMetaData

## DSAttachJob

Utilice la función DSAttachJob para ejecutar un trabajo en la secuencia de control de trabajos. Cuando adjunte esta función a un trabajo, se devuelve un descriptor de contexto que se utiliza para direccionar el trabajo. Sólo puede haber un descriptor de contexto abierto para un trabajo determinado en cualquier momento.

### Sintaxis

*DescriptorContextoTrabajo* = DSAttachJob (*NombreTrabajo*, *ModalidadError*)

*DescriptorContextoTrabajo* es el nombre de una variable para mantener el valor de retorno que utiliza posteriormente cualquier otra función o rutina al hacer referencia al trabajo. No presuponga que este valor es un entero.

*NombreTrabajo* es una serie que proporciona el nombre del trabajo al que se conectará.

*ModalidadError* es un valor que especifica cómo otras rutinas que utilizan el descriptor de contexto deben notificar errores. Es una de las siguientes:

- DSJ.ERRFATAL Registra un mensaje muy grave y termina anormalmente el trabajo de control (valor predeterminado).
- DSJ.ERRWARNING Registra un mensaje de aviso pero continúa.
- DSJ.ERRNONE No registra ningún mensaje - el emisor de la llamada tiene toda la responsabilidad (sin embargo, se registrará anomalía del propio DSAttachJob).

### Comentarios

Un trabajo no se puede conectar a sí mismo.

El parámetro *NombreTrabajo* puede especificar una versión exacta del trabajo en forma de *trabajo%Reln.n.n* o la versión más reciente del trabajo con el formato *trabajo*. Si se libera el propio trabajo de control, obtendrá la versión liberada más reciente del *trabajo*. Si el trabajo de control es una versión de desarrollo, obtendrá la versión de desarrollo más reciente de *trabajo*.

### Ejemplo

A continuación se muestra un ejemplo de conexión al Release 11 del trabajo Qsales:

```
Qsales_handle = DSAttachJob ("Qsales%Rel11",
→ DSJ.ERRWARN)
```

## DSCheckRoutine

Utilice la función DSCheckRoutine para ver si una rutina BASIC está catalogada, en el VOC como un elemento invocable o en bien en el espacio de catálogo.

## Sintaxis

*Encontrado* = DSCheckRoutine(*NombreRutina*)

*NombreRutina* es el nombre de la rutina de BASIC que se comprobará.

*Encontrado* es un booleano. @False if *NombreRutina* not findable, else @True.

## Ejemplo

```
rtm$ok = DSCheckRoutine("DSU.DSSendMail")
If(NOT(rtm$ok)) Then
    * aquí manejo de errores
End.
```

## DSDetachJob

Utilice la función DSDetachJob para recuperar un parámetro *JobHandle* adquirido por la función DSAttachJob si no es necesario otro control de un trabajo (lo que permite que otro trabajo se convierta en el controlador). No es necesario llamar a esta función porque los trabajos conectados siempre se desconectan automáticamente cuando el trabajo de control finaliza.

## Sintaxis

*CódigoError* = DSDetachJob (*DescriptorContextoTrabajo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*CódigoError* es 0 si DSStopJob es satisfactorio; de lo contrario puede ser uno de los siguientes:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.

El único error posible es un intento de cerrar DSJ.ME. De lo contrario, la llamada siempre es satisfactoria.

## Ejemplo

El mandato siguiente desconecta el descriptor de contexto del trabajo qsales:

```
Deterr = DSDetachJob (qsales_handle)
```

## DSExecute

Utilice la función DSExecute para ejecutar un mandato DOS, UNIX o de motor desde una subrutina anterior a la etapa o posterior a la etapa.

## Sintaxis

Call DSExecute (*TipoShell*, *Mandato*, *Salida*, *CódigoRetornoSistema*)

*TipoShell* (entrada) especifica el tipo de mandato que desea ejecutar y es NT, UNIX o UV (para motor).

*Mandato* (entrada) es el mandato que se ejecutará. *Mandato* no debe solicitar entrada cuando se ejecute.

*Salida* (salida) es cualquier salida del mandato. Cada línea de salida aparece separada por una marca de campo, @FM. La salida se añade al archivo de registro de trabajo como un mensaje informativo.

*CódigoRetornoSistema* (salida) es el código que indica el resultado satisfactorio del mandato. Un valor de 0 significa que el mandato se ha ejecutado satisfactoriamente. Un valor de 1 (para un mandato de DOS o UNIX) indica que el mandato no se ha encontrado. Cualquier otro valor es un código de salida específico del mandato.

## Comentarios

No utilice DSExecute desde una transformación; la sobrecarga de ejecutar un mandato para cada fila procesada por una etapa degradará el rendimiento del trabajo.

## DSGetCustInfo

Utilice la función DSGetCustInfo para obtener información que se ha notificado al final de la ejecución de determinadas etapas paralelas. Durante el tiempo de diseño, especifique la información recopilada y disponible para el interrogatorio. Por ejemplo, puede especificar información de la etapa Transformen en el separador Desencadenantes del recuadro de diálogo Propiedades de la etapa Transformer.

## Sintaxis

*Resultado* = DSGetCustInfo (*DescriptorContextoTrabajo*, *NombreEtapa*, *NombreInfoPers*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*NombreEtapa* es el nombre de la etapa que se interrogará. También puede ser DSJ.ME para hacer referencia a la etapa actual, si es necesario.

*NombreInfoPers* es el nombre de la variable que se interrogará.

*TipoInfo* especifica la información necesaria y puede ser una de las opciones siguientes:

DSJ.CUSTINFOVALUE

DSJ.CUSTINFODESC

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.CUSTINFOVALUE Serie: valor del elemento de información de personalización especificado.
- DSJ.CUSTINFODESC Serie: descripción del elemento de información de personalización especificado.

*Resultado* puede devolver también una condición de error de la forma siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.
- DSJE.BADTYPE *TipoInfo* no se ha reconocido.
- DSJE.NOTINSTAGE *NombreEtapa* era DSJ.ME y el emisor de la llamada no se está ejecutando dentro de una etapa.
- DSJE.BADSTAGE *NombreEtapa* no hace referencia a una etapa conocida en el trabajo.
- DSJE.BADCUSTINFO *NombreInfoPers* no hace referencia a un elemento de información de personalización conocido.



## DSGetJobInfo

Utilice la función DSGetJobInfo para obtener información sobre un trabajo. Puede utilizar esta información de forma general así como para el control de trabajos. La función DSGetJobInfo puede hacer referencia al trabajo actual de un trabajo controlado, dependiendo del valor de la variable *JobHandle*.

### Sintaxis

*Resultado* = DSGetJobInfo (*DescriptorContextoTrabajo*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*TipoInfo* especifica la información necesaria y puede ser una de las opciones siguientes:

DSJ.JOBSTATUS

DSJ.JOBNAME

DSJ.JOBCONTROLLER

DSJ.JOBSTARTTIMESTAMP

DSJ.JOBWAVENO

DSJ.PARAMLIST

DSJ.STAGELIST

DSJ.USERSTATUS

DSJ.JOBCONTROL

DSJ.JOBPID

DSJ.JPBLASTTIMESTAMP

DSJ.JOBINVOCATIONS

DSJ.JOBINTERIMSTATUS

DSJ.JOBINVOCATIONID

DSJ.JOBDESC

DSJ.JOBFULLDESC

DSJ.STAGELIST2

DSJ.JOBELAPSED

DSJ.JOBEOTCOUNT

DSJ.JOBEOTTIMESTAMP

DSJ.JOBRTISERVICE

DSJ.JOBMULTIINVOKABLE

DSJ.JOBFULLSTAGELIST

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- **DSJ.JOBSTATUS** *Entero*. Estado actual de trabajo global. Los estados posibles que se pueden devolver se dividen actualmente en dos categorías:

En primer lugar, un trabajo en curso se identifica mediante:

DSJS.RESET El trabajo ha finalizado una ejecución de restablecimiento.

DSJS.RUNFAILED El trabajo ha finalizado una ejecución normal con un error muy grave.

DSJS.RUNNING Ejecución de trabajo - es el único estado que indica que el trabajo está realmente en ejecución.

En segundo lugar, es posible que los trabajos que no estén en ejecución tengan los estados siguientes:

DSJS.RUNOK El trabajo ha finalizado una ejecución normal sin avisos.

DSJS.RUNWARN El trabajo ha finalizado una ejecución normal con avisos.

DSJS.STOPPED El trabajo ha sido detenido por la intervención del operador (no se puede determinar el tipo de ejecución).

DSJS.VALFAILED El trabajo no ha podido realizar una ejecución de validación.

DSJS.VALOK - El trabajo ha finalizado una ejecución de validación sin avisos.

DSJS.VALWARN El trabajo ha finalizado una ejecución de validación con avisos.

- **DSJ.JOBNAME** *Serie*. Nombre real del trabajo al que hace referencia el descriptor de contexto del trabajo.
- **DSJ.JOBCONTROLLER** *Serie*. Nombre del trabajo que controla el trabajo al que hace referencia el descriptor de contexto del trabajo. Tenga en cuenta que puede tratarse de varios nombres de trabajo separados por puntos si el trabajo está controlado por un trabajo que está a su vez controlado.
- **DSJ.JOBSTARTTIMESTAMP** *Serie*. Fecha y hora cuando se inició el trabajo en el motor con el formato AAAA-MM-DD hh:mm:ss.
- **DSJ.JOBWAVENO** *Entero*. Número de ondas de ejecución última o actual.
- **DSJ.PARAMLIST**. Devuelve una lista de nombres de parámetros separados por coma.
- **DSJ.STAGELIST**. Devuelve una lista de nombres de etapas activas separados por coma.
- **DSJ.USERSTATUS** *Serie*. Lo que ha registrado la última llamada del trabajo de DSSetUserStatus, de lo contrario una serie vacía.
- **DSJ.JOBCONTROL** *Entero*. Estado de control del trabajo actual, es decir, si se ha emitido una solicitud de detención para el trabajo.
- **DSJ.JOBPID** *Entero*. ID de proceso de trabajo.
- **DSJ.JOBLASTTIMESTAMP** *Serie*. Fecha y hora cuando el trabajo finalizó por última vez una ejecución en el motor con el formato AAAA-MM-DD HH:NN:SS.
- **DSJ.JOBINVOICATIONS**. Devuelve una lista de ID de invocación separados por coma.
- **DSJ.JOBINTERIMSTATUS**. Devuelve el estado de un trabajo después de que haya ejecutado todas las etapas y trabajos controlados, pero antes de haber

intentado ejecutar una subrutina posterior al trabajo. (Está diseñado para ser utilizado por una subrutina posterior al trabajo para obtener el estado del trabajo actual).

- DSJ.JOBINVOCATIONID. Devuelve el ID de invocación del trabajo especificado (utilizado en la macro DSJobInvocationId en un diseño de trabajo para acceder al ID de invocación mediante el cual se invoca el trabajo).
- DSJ.STAGELIST2. Devuelve una lista de nombres de etapas pasivas separadas por coma.
- DSJ.JOBELAPSED *Serie*. El tiempo transcurrido del trabajo en segundos.
- DSJ.JOBDESC *Serie*. La descripción del trabajo especificada en el recuadro de diálogo Propiedades del trabajo.
- DSJ.JOBFULLDESSC *Serie*. La descripción completa especificada en el recuadro de diálogo Propiedades del trabajo.
- DSJ.JOBRTISERVICE *entero*. Se establece en true (verdadero) si se trata de un trabajo de servicio web.
- DSJ.JOBMULTIINVOKABLE *entero*. Se establece en true (verdadero) si este trabajo da soporte a varias invocaciones
- DSJ.JOBEOTCOUNT *entero*. Recuento de bloques EndOfTransmission procesados por este trabajo hasta ahora.
- DSJ.JOBEOTTIMESTAMP *indicación\_fecha\_y\_hora*. Fecha/Hora del último bloque EndOfTransmission procesado por este trabajo.
- DSJ.FULLSTAGELIST. Devuelve una lista de todos los nombres de etapas separados por coma.

*Resultado* también puede devolver condiciones de error de la forma siguiente:

DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.

DSJE.BADTYPE *TipoInfo* no se ha reconocido.

## Comentarios

Cuando se hace referencia a un trabajo controlado, se puede utilizar DSGetJobInfo antes o después de haber emitido un DSRunJob. Cualquier estado devuelto después de una llamada satisfactoria a DSRunJob es una garantía de que está relacionado con esa ejecución del trabajo.

## Ejemplos

El mandato siguiente solicita el estado de trabajo del trabajo qsales:

```
q_status = DSGetJobInfo(qsales_handle, DSJ.JOBSTATUS)
```

El mandato siguiente solicita el nombre real del trabajo actual:

```
whatname = DSGetJobInfo (DSJ.ME, DSJ.JOBNAME)
```

## DSGetLinkInfo

Utilice la función DSGetLinkInfo para obtener información sobre un enlace en una etapa activa. Puede utilizar esta información de forma general así como para el control de trabajos. La función DSGetLinkInfo puede hacer referencia a un trabajo controlado o al trabajo actual, en función del valor del valor de la variable *JobHandle*.

## Sintaxis

*Resultado* = DSGetLinkInfo (*DescriptorContextoTrabajo*, *NombreEtapa*, *NombreEnlace*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*NombreEtapa* es el nombre de la etapa activa que se interrogará. También puede ser DSJ.ME para hacer referencia a la etapa actual, si es necesario.

*NombreEnlace* es el nombre de un enlace (entrada o salida) conectado a la etapa. También puede ser DSJ.ME para hacer referencia al enlace actual (por ejemplo, cuando se utiliza en una expresión de Transformer o función de transformación denominada desde código de enlace).

*TipoInfo* especifica la información necesaria y puede ser una de las opciones siguientes:

DSJ.LINKLASTERR

DSJ.LINKNAME

DSJ.LINKROWCOUNT

DSJ.LINKSQLSTATE

DSJ.LINKDBMSCODE

DSJ.LINKDESC

DSJ.LINKSTAGE

DSJ.INSTROWCOUNT

DSJ.LINKEOTROWCOUNT

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.LINKLASTERR Serie: último mensaje de error (si hay alguno) notificado desde el enlace en cuestión.
- DSJ.LINKNAME Serie: devuelve el nombre del enlace, resulta especialmente útil cuando se utiliza con *DescriptorContextoTrabajo* = DSJ.ME y *NombreEtapa* = DSJ.ME y *NombreEnlace* = DSJ.ME para descubrir su propio nombre.
- DSJ.LINKROWCOUNT Entero: número de filas que han pasado un enlace hasta ahora.
- DSJ.LINKSQLSTATE: el estado de SQL para el último error que se produce en este enlace.
- DSJ.LINKDBMSCODE: el código de DBMS para el último error que se produce en este enlace.
- DSJ.LINKDESC: descripción del enlace.
- DSJ.LINKSTAGE: nombre de la etapa en el otro extremo del enlace.
- DSJ.INSTROWCOUNT: lista de recuentos de fila, separados por coma, uno por instancia (trabajos paralelos)
- DSJ.LINKEOTROWCOUNT: recuento de filas desde el último bloque EndOfTransmission.

*Resultado* también puede devolver condiciones de error de la forma siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.
- DSJE.BADTYPE *TipoInfo* no se ha reconocido.
- DSJE.BADSTAGE *NombreEtapa* no hace referencia a una etapa conocida en el trabajo.
- DSJE.NOTINSTAGE *NombreEtapa* era DSJ.ME y el emisor de la llamada no se está ejecutando dentro de una etapa.
- DSJE.BADLINK *NombreEnlace* no hace referencia a un enlace conocido para la etapa en cuestión.

## Comentarios

Cuando se hace referencia a un trabajo controlado, se puede utilizar DSGetLinkInfo antes o después de haber emitido un DSRunJob. Cualquier estado devuelto después de una llamada satisfactoria a DSRunJob es una garantía de que está relacionado con esa ejecución del trabajo.

## Ejemplo

El siguiente mandato solicita el número de filas que han descendido por el enlace order\_feed en la etapa de cargador del trabajo qsales:

```
link_status = DSGetLinkInfo(qsales_handle, "cargador",  
→ "order_feed", DSJ.LINKROWCOUNT)
```

## DSGetLogEntry

Utilice la función DSGetLogEntry para leer los detalles completos del suceso de un suceso de registro específico con el nombre *IDSuceso*.

### Sintaxis

*DetalleSuceso* = DSGetLogEntry (*DescriptorContextoTrabajo*, *IDSuceso*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*IDSuceso* es un entero que identifica el suceso de registro específico para el que son necesarios los detalles. Se obtiene utilizando la función DSGetNewestLogId.

*DetalleSuceso* es una serie que contiene subseries separadas por \. Las subseries tienen la forma siguiente:

Subserie1 Indicación de fecha y hora en formato AAAA-MM-DD HH:NN:SS

Subserie2 Información de usuario

Subserie3 Tipo de suceso - consulte DSGetNewestLogId

Subserie4 Mensaje del suceso

Si se produce un error, el error se notifica mediante uno de los siguientes códigos de resultado de entero negativo:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADVALUE Error al acceder a *IDSuceso*.

## Ejemplo

Los mandatos siguientes en primer lugar obtienen el IDSuceso para el suceso de registro necesario y a continuación leen todos los detalles del suceso del suceso de registro identificado por LatestLogid en la serie LatestEventString:

```
latestlogid =  
→ DSGetNewestLogId(qsales_handle,DSJ.LOGANY)  
LatestEventString =  
→ DSGetLogEntry(qsales_handle,latestlogid)
```

## DSGetLogEntryFull

Utilice la función DSGetLogEntryFull para leer los detalles completos de un suceso, incluidos el ID de mensaje y el ID de invocación.

### Sintaxis

*DetalleSuceso* = DSGetLogEntryFull (*DescriptorContextoTrabajo*, *IdSuceso*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*Idsuceso* es un entero que identifica el suceso de registro específico para el que son necesarios los detalles. Se obtiene utilizando la función DSGetNewestLogId.

*DetalleSuceso* es una serie que contiene subseries separadas por \. Las subseries tienen la forma siguiente:

Subserie1 Indicación de fecha y hora en formato AAAA-MM-DD HH:NN:SS

Subserie2 Información de usuario

Subserie3 Tipo de suceso - consulte DSGetNewestLogId

Subserie4 Mensaje del suceso

Subserie5 ID de mensaje

Subserie6 ID de invocación

Si se produce un error, el error se notifica mediante uno de los siguientes códigos de resultado de entero negativo:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADVALUE Error al acceder a *Idsuceso*.

## Ejemplo

Los mandatos siguientes en primer lugar obtienen el IDSuceso para el suceso de registro necesario y a continuación leen todos los detalles del suceso del suceso de registro identificado por LatestLogid en la serie LatestEventString:

```
latestlogid =  
→ DSGetNewestLogId(qsales_handle,DSJ.LOGANY)  
LatestEventString =  
→ DSGetLogEntryFull(qsales_handle,latestlogid)
```

## DSGetLogEventIds

Utilice la función DSGetLogEventIds para devolver una lista de ID de suceso para una ejecución de una invocación de trabajo.

### Sintaxis

*ListaID* = DSGetLogEventIds (*DescriptorContextoTrabajo*, *NúmeroEjecución*, *FiltroTipoSuceso*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*NúmeroEjecución* identifica la ejecución de invocación de trabajo para la que se devuelven los ID de suceso. Normalmente un valor de cero solicita los ID para la ejecución más reciente de la invocación de trabajo. Para recuperar detalles para ejecuciones anteriores, proporcione valores negativos, como por ejemplo -1 para obtener detalles sobre la ejecución anterior a la más reciente, -2 para obtener detalles sobre la ejecución anterior a esa, y así sucesivamente. En los casos dónde conozca los números explícitos de ejecución, puede recuperar detalles proporcionando el número de ejecución como un valor positivo.

*FiltroTipoSuceso* restringe los tipos de entrada de registro de suceso para los que se devuelven los ID. De forma predeterminada se devuelven los ID para todas las entradas de registro. Incluya caracteres en la serie del filtro para restringir las entradas de la forma siguiente:

I	Informativo
W	Aviso
F	Muy grave
S	Sucesos de inicio o finalización
B	Sucesos de proceso por lotes o control
R	Sucesos de depuración o restablecimiento
J	Sucesos de rechazo

*ListaID* se devuelve como una lista de enteros positivos que identifican los sucesos de registro necesarios. En caso de un error, *ListaID* también se puede devolver como un entero negativo, en cuyo caso contiene uno de estos códigos de error:

#### DSJE.BADHANDLE

*DescriptorContextoTrabajo* no válido.

#### DSJE.BADTYPE

*FiltroTipoSuceso* no válido.

#### DSJE.BADVALUE

*NúmeroEjecución* no válido.

### Comentarios

Para utilizar este método, es necesario que el programa haya adquirido anteriormente un descriptor de contexto de trabajo llamando a DSAttachJob.

El número de ejecución para una invocación de trabajo se restablece cuando se compila el trabajo; por lo tanto, no es posible utilizar este método para recuperar

ID de suceso de trabajo para ejecuciones que se han producido antes de la compilación más reciente del trabajo.

## DSGetLogSummary

Utilice la función DSGetLogSummary para devolver una lista de detalles cortos del suceso de registro. Los detalles que se devuelven están en función de los valores de determinados filtros. Tenga cuidado con los valores de los filtros, de lo contrario se puede devolver una gran cantidad de información.

### Sintaxis

*MatrizResumen* = DSGetLogSummary (*DescriptorContextoTrabajo*, *TipoSuceso*, *HoraInicio*, *HoraFinalización*, *NúmeroMáx*)

*DescriptorContextotrabajo* es el descriptor de contexto para el trabajo tal como se deriva de *DSAttachJob*.

*TipoSuceso* es el tipo de suceso registrado y es uno de los siguientes:

- DSJ.LOGININFO Mensaje de información
- DSJ.LOGWARNING Mensaje de aviso
- DSJ.LOGFATAL Error muy grave
- DSJ.LOGREJECT El enlace de rechazo estaba activo
- DSJ.LOGSTARTED Se ha iniciado el trabajo
- DSJ.LOGRESET Se ha restablecido el registro
- DSJ.LOGANY Cualquier categoría (el valor predeterminado)

*HoraInicio* es una serie con el formato AAAA-MM-DD HH:NN:SS o AAAA-MM-DD.

*HoraFinalización* es una serie con el formato AAAA-MM-DD HH:NN:SS o AAAA-MM-DD.

*NúmeroMáx* es un entero que restringe el número de sucesos a devolver. 0 significa ninguna restricción. Utilice este valor con precaución.

*MatrizResumen* es una matriz dinámica de campos separados por @FM. Cada campo consta de un conjunto de subseries separadas por \, donde cada campo representa un suceso separado, con las subseries con el formato siguiente:

Subserie1 *IDSuceso* como por DSGetLogEntry

Subserie2 Indicación de fecha y hora con el formato AAAA-MM-DD HH:NN:SS

Subserie3 *TipoSuceso* - consulte DSGetNewestLogId

Subserie4 - *n* Mensaje de suceso

Si se produce un error, el error se notifica mediante uno de los siguientes códigos de resultado de entero negativo:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADTYPE *TipoSuceso* no válido.
- DSJE.BADTIME *HoraInicio* u *HoraFinalización* no válido.
- DSJE.BADVALUE *NúmeroMáx* no válido.



## Ejemplo

El mandato siguiente produce una matriz de sucesos activos de enlace de rechazo registrados para el trabajo qsales entre el 18 de agosto de 1998 y el 18 de septiembre de 1998, hasta un máximo de MAXREJ entradas:

```
RejEntries = DSGetLogSummary (qsales_handle,  
→ DSJ.LOGREJECT, "1998-08-18 00:00:00", "1998-09-18  
→ 00:00:00", MAXREJ)
```

## DSGetNewestLogId

Utilice la función DSGetNewestLogId para obtener el ID del suceso de registro más reciente en una categoría.

### Sintaxis

*IDsuceso* = DSGetNewestLogId (*DescriptorContextoTrabajo*, *TipoSuceso*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*TipoSuceso* es el tipo de suceso registrado y es uno de los siguientes:

- DSJ.LOGININFO Mensaje de información
- DSJ.LOGWARNING Mensaje de aviso
- DSJ.LOGFATAL Error muy grave
- DSJ.LOGREJECT El enlace de rechazo estaba activo
- DSJ.LOGSTARTED Se ha iniciado el trabajo
- DSJ.LOGRESET Se ha restablecido el registro
- DSJ.LOGANY Cualquier categoría (el valor predeterminado)

*IDsuceso* es un entero positivo que identifica el suceso de registro específico. En caso de un error, *IDsuceso* también se puede devolver como un entero negativo, en cuyo caso contiene un código de error como el siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADTYPE *TipoSuceso* no válido.

## Ejemplo

El mandato siguiente obtiene un ID para el mensaje de aviso más reciente en el registro para el trabajo qsales:

```
IDAviso = DSGetNewestLogId (qsales_handle,  
→ DSJ.LOGWARNING)
```

## DSGetParamInfo

Utilice la función DSGetParamInfo para obtener información sobre un trabajo. Puede utilizar esta información para el control de trabajos. La función DSGetParamInfo puede hacer referencia a un trabajo controlado o al trabajo actual, en función del valor de la variable *JobHandle*.

### Sintaxis

*Resultado* = DSGetParamInfo (*DescriptorContextoTrabajo*, *NombreParám*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*NombreParám* es el nombre del parámetro que se interrogará.

*TipoInfo* especifica la información necesaria y puede ser uno de los valores siguientes:

DSJ.PARAMDEFAULT

DSJ.PARAMHELPTTEXT

DSJ.PARAMPROMPT

DSJ.PARAMTYPE

DSJ.PARAMVALUE

DSJ.PARAMDES.DEFAULT

DSJ.PARAMLISTVALUES

DSJ.PARAMDES.LISTVALUES

DSJ.PARAMPROMPT.AT.RUN

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.PARAMDEFAULT Serie: valor predeterminado actual para el parámetro en cuestión. Consulte también DSJ.PARAMDES.DEFAULT.
- DSJ.PARAMHELPTTEXT Serie: texto de ayuda (si existe) para el parámetro en cuestión.
- DSJ.PARAMPROMPT Serie: solicitud (si existe) para el parámetro en cuestión.
- DSJ.PARAMTYPE Entero: describe el tipo de prueba de validación que se debe realizar en cualquier valor que se establezca para este parámetro. Es uno de los siguientes:
  - DSJ.PARAMTYPE.STRING
  - DSJ.PARAMTYPE.ENCRYPTED
  - DSJ.PARAMTYPE.INTEGER
  - DSJ.PARAMTYPE.FLOAT (el parámetro puede contener puntos y E)
  - DSJ.PARAMTYPE.PATHNAME
  - DSJ.PARAMTYPE.LIST (debe ser un conjunto de series separadas por tabulador)
  - DSJ.PARAMTYPE.DATE (debe ser una serie con el formato AAAA-MM-DD)
  - DSJ.PARAMTYPE.TIME (debe ser una serie con el formato HH:MM)
- DSJ.PARAMVALUE Serie: valor actual del parámetro para el trabajo en ejecución o la última ejecución de trabajo si el trabajo ha finalizado.
- DSJ.PARAMDES.DEFAULT Serie: valor predeterminado original del parámetro - puede ser distinto de DSJ.PARAMDEFAULT si este último ha sido cambiado por un administrador desde que se instaló el trabajo.
- DSJ.PARAMLISTVALUES Serie: lista separada por tabuladores de valores permitidos para el parámetro. Consulte también DSJ.PARAMDES.LISTVALUES.
- DSJ.PARAMDES.LISTVALUES Serie: lista original separada por tabuladores de valores permitidos para el parámetro - puede ser diferente de DSJ.PARAMLISTVALUES si este último ha sido cambiado por el administrador desde que se instaló el trabajo.

- DSJ.PROMPT.AT.RUN Serie: 1 significa que se solicitará el parámetro cuando se ejecute el trabajo; cualquier otro valor significa que no se solicitará (DSJ.PARAMDEFAULT Serie que se utilizará directamente).

*Resultado* también puede devolver condiciones de error de la forma siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.
- DSJE.BADPARAM *NombreParám* no es un nombre de parámetro en el trabajo.
- DSJE.BADTYPE *TipoInfo* no se ha reconocido.

## Comentarios

Cuando se hace referencia a un trabajo controlado, DSGetParamInfo se puede utilizar antes o después de haber emitido un DSRunJob. Cualquier estado devuelto después de una llamada satisfactoria a DSRunJob es una garantía de que está relacionado con esa ejecución del trabajo.

## Ejemplo

El mandato siguiente solicita el valor predeterminado del parámetro de trimestre para el trabajo qsales:

```
Qs_quarter = DSGetparamInfo(qsales_handle, "trimestre",
→ DSJ.PARAMDEFAULT)
```

## DSGetProjectInfo

Utilice la función DSGetProjectInfo para obtener información sobre el proyecto actual.

### Sintaxis

*Resultado* = DSGetProjectInfo (*TipoInfo*)

*TipoInfo* especifica la información necesaria y puede ser una de las opciones siguientes:

DSJ.JOBLIST

DSJ.PROJECTNAME

DSJ.HOSTNAME

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.JOBLIST Serie - lista de nombres separados por coma de todos los trabajos conocidos por el proyecto (tanto si los trabajos están conectados actualmente como si no).
- DSJ.PROJECTNAME Serie - nombre del proyecto actual.
- DSJ.HOSTNAME Serie - nombre del host del motor que mantiene el proyecto actual.

Resultado puede devolver también una condición de error de la forma siguiente:

- DSJE.BADTYPE *TipoInfo* no se ha reconocido.

## DSGetStageInfo

Utilice la función DSGetStageInfo para obtener información sobre una etapa. Puede utilizar esta información para el control de trabajos. La función DSGetStageInfo puede hacer referencia al trabajo actual o a un trabajo controlado en función del valor de la variable *JobHandle*.

### Sintaxis

*Resultado* = DSGetStageInfo (*DescriptorContextoTrabajo*, *NombreEtapa*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*NombreEtapa* es el nombre de la etapa que se interrogará. También puede ser DSJ.ME para hacer referencia a la etapa actual, si es necesario.

*TipoInfo* especifica la información necesaria y puede ser uno de los valores siguientes:

DSJ.LINKLIST

DSJ.STAGELASTERR

DSJ.STAGENAME

DSJ.STAGETYPE

DSJ.STAGEINROWNUM

DSJ.VARLIST

DSJ.STAGESTARTTIMESTAMP

DSJ.STAGEENDTIMESTAMP

DSJ.STAGEDESC

DSJ.STAGEINST

DSJ.STAGECPU

DSJ.LINKTYPES

DSJ.STAGEELAPSED

DSJ.STAGEPID

DSJ.STAGESTATUS

DSJ.STAGEEOTCOUNT

DSJ.STAGEEOTTIMESTAMP

DSJ.CUSTINFOLIST

DSJ.STAGEEOTSTART

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.LINKLIST - lista separada por comas de nombres de enlace en la etapa.
- DSJ.STAGELASTERR Serie - último mensaje de error (si hay alguno) notificado desde cualquier enlace de la etapa en cuestión.
- DSJ.STAGENAME Serie - resulta especialmente útil cuando se utiliza con *DescriptorContextoSerie* = DSJ.ME y *NobreEtapa* = DSJ.ME para descubrir su propio nombre.
- DSJ.STAGETYPE Serie - el nombre de tipo de etapa (por ejemplo, "Transformer", "BeforeJob").
- DSJ.STAGEINROWNUM Entero - el número de fila de entrada del enlace primario.
- DSJ.VARLIST - lista separada por comas de nombres de variables de etapa.
- DSJ.STAGESTARTTIMESTAMP - fecha/hora en la que esa etapa ha empezado su ejecución con el formato AAAA-MM-DD HH:NN:SS.
- DSJ.STAGEENDTIMESTAMP - fecha/hora en la que esa etapa ha finalizado su ejecución con el formato AAA-MM-DD HH:NN:SS.
- DSJ.STAGEDESC - descripción de etapa.
- DSJ.STAGEINST - lista separada por comas de ID de instancia (trabajos paralelos).
- DSJ.STAGECPU - porcentaje en entero de CPU utilizada.
- DSJ.LINKTYPES - lista separada por comas de tipos de enlace.
- DSJ.STAGEELAPSED - tiempo transcurrido en segundos.
- DSJ.STAGEPID - lista separada por comas de ID de proceso.
- DSJ.STAGESTATUS - estado de etapa.
- DSJ.STAGEEOTCOUNT - Recuento de bloques EndOfTransmission procesados por esta etapa hasta ahora.
- DSJ.STAGEEOTTIMESTAMP - Fecha/hora del último bloque EndOfTransmission recibido por esta etapa.
- DSJ.CUSTINFOLIST - información personalizada generada por etapas (trabajos paralelos).
- DSJ.STAGEEOTSTART - recuento de filas al inicio del bloque EndOfTransmission actual.

*Resultado* también puede devolver condiciones de error de la forma siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.
- DSJE.BADTYPE *TipoInfo* no se ha reconocido.
- DSJE.NOTINSTAGE *NombreEtapa* era DSJ.ME y el emisor de la llamada no se está ejecutando dentro de una etapa.
- DSJE.BADSTAGE *NombreEtapa* no hace referencia a una etapa conocida en el trabajo.

## Comentarios

Cuando se hace referencia a un trabajo controlado, se puede utilizar DSGGetStageInfo antes o después de emitir un DSRunJob. Cualquier estado devuelto después de una llamada satisfactoria a DSRunJob es una garantía de que está relacionado con esa ejecución del trabajo.

## Ejemplo

El mandato siguiente solicita el último mensaje de error para la etapa de cargador del trabajo qsales:

```
stage_status = DSGetStageInfo(qsales_handle, "cargador",  
→ DSJ.STAGELASTERR)
```

## DSGetVarInfo

Utilice la función DSGetVarInfo para obtener información sobre variables que se utiliza en etapas Transformer.

### Sintaxis

*Resultado* = DSGetVarInfo (*DescriptorContextoTrabajo*, *NombreEtapa*, *NombreVar*, *TipoInfo*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob o puede ser DSJ.ME para hacer referencia al trabajo actual.

*NombreEtapa* es el nombre de la etapa que se interrogará. También puede ser DSJ.ME para hacer referencia a la etapa actual, si es necesario.

*NombreVar* es el nombre de la variable que se interrogará.

*TipoInfo* especifica la información necesaria y puede ser una de las opciones siguientes:

DSJ.VARVALUE

DSJ.VARDESCRIPTION

*Resultado* depende del *TipoInfo* especificado, de la forma siguiente:

- DSJ.VARVALUE Serie: el valor de la variable especificada.
- DSJ.VARDESCRIPTION Serie: descripción de la variable especificada.

*Resultado* puede devolver también una condición de error de la forma siguiente:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no era válido.
- DSJE.BADTYPE *TipoInfo* no se ha reconocido.
- DSJE.NOTINSTAGE *NombreEtapa* era DSJ.ME y el emisor de la llamada no se está ejecutando dentro de una etapa.
- DSJE.BADVAR *NombreVar* no se ha reconocido.
- DSJE.BADSTAGE *NombreEtapa* no hace referencia a una etapa conocida en el trabajo.

## DSLogEvent

Utilice la función DSLogEvent para registrar un mensaje de suceso en un trabajo distinto del actual. (Utilice la función DSLogInfo, DSLogFatal o DSLogWarn para registrar un mensaje de suceso en el trabajo actual.)

### Sintaxis

*CódigoError* = DSLogEvent (*DescriptorContextoTrabajo*, *TipoSuceso*, *MsjSuceso*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*TipoSuceso* es el tipo de suceso registrado y es uno de los siguientes:

- DSJ.LOGINFO Mensaje de información
- DSJ.LOGWARNING Mensaje de aviso

*MsjSuceso* es una serie que contiene el mensaje del suceso.

*CódigoError* es 0 si no hay ningún error. De lo contrario, contiene uno de los errores siguientes:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADTYPE *TipoSuceso* (especialmente, tenga en cuenta que no puede colocar un mensaje muy grave en el registro de otro trabajo).

## Ejemplo

El mandato siguiente, cuando se incluye en el trabajo msales, añade el mensaje "ventas mensuales finalizadas" al registro para el trabajo qsales:

```
Logerror = DsLogEvent (qsales_handle, DSJ.LOGINFO,  
→ "ventas mensuales finalizadas")
```

## DSLogFatal

Utilice la función DSLogFatal para anotar un mensaje de error muy grave en el archivo de registro de un trabajo y terminar el trabajo.

### Sintaxis

Call DSLogFatal (*Mensaje*, *NombreProgLlamada*)

*Mensaje* (entrada) es el mensaje de aviso que desea registrar. *Mensaje* recibe automáticamente como prefijo el nombre de la etapa actual y la subrutina anterior/posterior de llamada.

*NombreProgLlamada* (entrada) es el nombre de la subrutina anterior/posterior que llama a la subrutina DSLogFatal.

### Comentarios

DSLogFatal graba un mensaje de error muy grave en el archivo de registro del trabajo y termina anormalmente el trabajo. DSLogFatal nunca vuelve a la subrutina anterior/posterior de llamada, de forma que se debe utilizar con precaución. Si un trabajo se detiene con un error muy grave, se debe restablecer utilizando el cliente del Director antes de poder volver a ejecutarlo.

En una subrutina anterior/posterior, es mejor registrar un mensaje de aviso (utilizando DSLogWarn) y salida con un código de retorno distinto de cero, lo que permite que InfoSphere DataStage detenga el trabajo limpiamente.

DSLogFatal no se debe utilizar en una transformación. Se utiliza DSTransformError en su lugar.

## Ejemplo

```
Call DSLogFatal("No se puede abrir el archivo", "MiRutina")
```

## DSLogInfo

Utilice la función DSLogInfo para anotar un mensaje de aviso en un archivo de registro de trabajo.

## Sintaxis

```
Call DSLogInfo (Mensaje, NombreProgLlamada)
```

*Mensaje* (entrada) es el mensaje de información que desea registrar. *Mensaje* recibe automáticamente como prefijo el nombre de la etapa actual y del programa de llamada.

*NombreProgLlamada* (entrada) es el nombre de la transformación o subrutina anterior/posterior que llama a la subrutina DSLogInfo.

## Comentarios

DSLogInfo graba el texto del mensaje en el archivo de registro del trabajo como un mensaje de información y vuelve a la rutina de llamada o transformación. Si se llama a DSLogInfo durante la fase de prueba para una rutina recientemente creada en el repositorio, los dos argumentos se visualizan en la ventana de resultados.

Se puede grabar un número ilimitado de mensajes de información en el archivo de registro del trabajo. Sin embargo, si se producen muchos mensajes, es posible que el trabajo se ejecute lentamente y que el cliente del Director tarde algún tiempo en mostrar el archivo de registro del trabajo.

## Ejemplo

```
Call DSLogInfo("Transformando: ":Arg1, "MiTransformación")
```

## DSLogToController

Utilice la función DSLogToController para colocar un mensaje informativo en el archivo de registro del trabajo que controla otro trabajo.

## Sintaxis

```
Call DSLogToController(SerieMsj)
```

*SerieMsj* es el texto que se registrará. El suceso de registro es de tipo Información.

## Comentarios

Si el trabajo actual no está bajo control, se realiza una salida silenciosa.

## Ejemplo

```
Call DSLogToController("Se registra en el padre")
```

## DSLogWarn

Utilice la función DSLogWarn para anotar un mensaje de aviso en un archivo de registro de trabajo.

## Sintaxis

```
Call DSLogWarn (Mensaje, NombreProgLlamada)
```

*Mensaje* (entrada) es el mensaje de aviso que desea registrar. *Mensaje* recibe automáticamente como prefijo el nombre de la etapa actual y la subrutina anterior/posterior de llamada.

*NombreProgLlamada* (entrada) es el nombre de la subrutina anterior/posterior que llama a la subrutina DSLogWarn.



## Comentarios

DSLogWarn graba el mensaje en el archivo de registro del trabajo como aviso y devuelve la subrutina anterior/posterior de llamada. Si el trabajo tiene un límite de aviso definido para el mismo, cuando el número de avisos alcanza ese límite, la llamada no devuelve el valor de retorno y el trabajo termina anormalmente.

DSLogWarn no se debe utilizar en una transformación. Se utiliza DSTransformError en su lugar.

## Ejemplo

```
If InputArg > 100 Then
    Call DSLogWarn("La entrada debe ser =< 100; received
        ":InputArg,"MiRutina")
End Else
    * Continuar procesando a menos que el trabajo termine anormalmente
End
```

## DSMakeJobReport

Utilice la función DSMakeJobReport para generar un informe que describa el estado completo de un trabajo conectado/válido.

### Sintaxis

*TextoInforme* = DSMakeJobReport(*DescriptorContextoTrabajo*, *NivelInforme*, *SeparadorLíneas*)

*DescriptorContextoTrabajo* es la serie tal como se ha devuelto de DSAttachJob.

*NivelInforme* especifica el tipo de informe y es uno de los siguientes:

- 0 - informe básico. Serie de texto que contiene fecha de inicio/finalización, tiempo transcurrido y estado de trabajo.
- 1 - detalle de etapa/enlace. Como el informe básico, pero también contiene información sobre etapas individuales y enlaces dentro del trabajo.
- 2 - serie de texto que contiene el informe XML completo.

De manera predeterminada, el XML generado no contendrá una instrucción de proceso <?xml-stylesheet?>. Si es necesaria una hoja de estilo, especifique un NivelInforme de 2 y añada el nombre del URL de hoja de estilo necesario, es decir, 2;URLHojaEstilo. Esto inserta una instrucción de proceso en el XML generado con el formato:

```
<?xml-stylesheet type=text/xsl" href="URLHojaEstilo"?>
```

*SeparadorLíneas* es la serie utilizada para separar líneas del informe. Los valores especiales reconocidos son:

- "CRLF" => CHAR(13):CHAR(10)
- "LF" => CHAR(10)
- "CR" => CHAR(13)

El valor predeterminado es CRLF si se trata de Windows, de lo contrario LF.

## Comentarios

Si se proporciona un descriptor de contexto de trabajo incorrecto, o se encuentra cualquier otro error, la información se añade a TextoInforme.

## Ejemplo

```
h$ = DSAttachJob("MiTrabajo", DSJ.ERRNONE)
rpt$ = DSMakeJobReport(h$,0,"CRLF")
```

## DSMakeMsg

Utilice la función `DSMakeMsg` para insertar argumentos en una plantilla de mensaje. De forma opcional, puede utilizar la función para buscar un ID de plantilla en el archivo de mensajes estándar de InfoSphere DataStage en lugar de la que se proporciona a la rutina.

### Sintaxis

*TextoCompleto* = `DSMakeMsg(Plantilla, ListaArg)`

*TextoCompleto* es el mensaje con parámetros sustituidos

*Plantilla* es la plantilla de mensaje, en la que %1, %2, etc. se sustituirán por valores en la posición equivalente en *ListaArg*. Si la serie de plantilla empieza con un número seguido de "\", se asume que forma parte de un ID de mensaje que se buscará en el archivo de mensajes de InfoSphere DataStage.

Nota: si una señal de argumento va seguida de "[E]", se asume que el valor de ese argumento es un código de error de control de trabajo y se insertará una explicación del mismo en lugar de "[E]". (Consulte la función `DSTranslateCode`.)

*ListaArg* es la matriz dinámica, un campo por argumento que se sustituirá.

### Comentarios

Se llama a esta rutina desde el código de control de trabajo creado por JobSequence Generator.

También realizará sustitución de parámetros de trabajo local en el texto del mensaje. Es decir, si se llama desde dentro de un trabajo, busca subseries como por ejemplo "#xyz#" y las sustituye por el valor del parámetro de trabajo denominado "xyz".

## Ejemplo

```
t$ = DSMakeMsg("Error al llamar a DSAttachJob(%1)<L>%2",
→jb$:@FM:DSGetLastErrorMsg())
```

## DSPrepareJob

Utilice la función `DSPrepareJob` para asegurarse de que un trabajo compilado está en el estado correcto para ser ejecutado y validado.

### Sintaxis

*DescriptorContextoTrabajo* = `DSPrepareJob(DescriptorContextoTrabajo)`

*DescriptorContextoTrabajo* es el descriptor de contexto, tal como lo devuelve `DSAttachJob()`, del trabajo que se preparará.

*DescriptorContextoTrabajo* es el descriptor de contexto original o uno nuevo. Si se devuelve como 0, se produce un error y se registra un mensaje.

## Ejemplo

```
h$ = DSPrepareJob(h$)
```

## DSRunJob

Utilice la función `DSRunJob` para empezar la ejecución de un trabajo. Esta llamada es asíncrona; la solicitud se pasa al motor de tiempo de ejecución, pero no se informa de su progreso.

### Sintaxis

```
CódigoError = DSRunJob (DescriptorContextoTrabajo, ModalidadEjecución)
```

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de `DSAttachJob`.

*RunMode* es el nombre de la modalidad en la que se ejecutará el trabajo y es una de las siguientes:

- `DSJ.RUNNORMAL` (Valor predeterminado) Ejecución de trabajo estándar.
- `DSJ.RUNRESET` El trabajo se restablecerá.
- `DSJ.RUNVALIDATE` El trabajo sólo se validará.
- `DSJ.RUNRESTART` La secuencia de trabajos reinicializable debe reiniciarse con los valores de parámetros del trabajo original.

*CódigoError* es 0 si `DSRunJob` es satisfactorio; de lo contrario puede ser uno de los enteros negativos siguientes:

- `DSJE.BADHANDLE` *DescriptorContextoTrabajo* no válido.
- `DSJE.BADSTATE` El trabajo no está en el estado correcto (compilado, no en ejecución).
- `DSJE.BADTYPE` *ModalidadEjecución* no es una modalidad conocida.

### Comentarios

Si el trabajo de control se ejecuta en modalidad de validación, cualquier llamada de `DSRunJob` actuará como si *ModalidadEjecución* fuera `DSJ.RUNVALIDATE`, independientemente del valor real.

Un trabajo en modalidad de validación ejecutará su rutina `JobControl` (si existe) en lugar de comprobar solamente su existencia, como es el caso para las rutinas anteriores/posteriores. Esto permite examinar el registro de qué trabajo ha iniciado en modalidad de validación.

Después de llamar a `DSRunJob`, el descriptor de contexto del trabajo controlado se descarga. Si es necesario ejecutar el mismo trabajo de nuevo, debe utilizar `DSDetachJob` y `DSAttachJob` para establecer un nuevo descriptor de contexto. Tenga en cuenta que también necesitará utilizar `DSWaitForJob`, ya que no puede se conectar a un trabajo mientras está en ejecución.

## Ejemplo

El mandato siguiente inicia el trabajo `qsales` en modalidad estándar:

```
RunErr = DSRunJob(qsales_handle, DSJ.RUNNORMAL)
```

## DSSendMail

Utilice la función DSSendMail como una interfaz para enviar un programa sendmail que se supone que existe en algún lugar de la vía de acceso de búsqueda del usuario actual (en el host de la capa de motor). La función oculta las diferentes interfaces de llamada a diversos programas sendmail y ofrece una simple interfaz para enviar texto.

### Sintaxis

*Respuesta* = DSSendMail(*Parámetros*)

*Parámetros* es un conjunto de parámetros nombre:valor, separados por un carácter de marcador o "\n".

Los nombres que se reconocen actualmente (sin distinguir entre mayúsculas y minúsculas) son los siguientes:

- "De" Dirección de correo del remitente, por ejemplo, Aqui@Enalgúnlugar.com  
Sólo se puede dejar en blanco si el archivo de plantilla local no contiene una señal "%from%".
- "Para" Dirección de correo del destinatario, por ejemplo, Tu@Enalgúnlugar.com  
Sólo se puede dejar en blanco si el archivo de plantilla local no contiene una señal "%to%".
- "Asunto" Algo que poner en la línea del asunto del mensaje.  
Hace referencia a la señal "%subject%". Si se deja como "", se creará una línea de asunto estándar, junto con las líneas de "From InfoSphere DataStage job: jobname"
- "Servidor" Nombre de host a través del cual se enviará el correo.  
Se puede omitir en sistemas (por ejemplo, Unix) donde el nombre de host SMTP se puede configurar, y se configura, externamente, en cuyo caso el archivo de plantilla local presumiblemente no contendrá una señal "%server%".
- "Cuerpo" Cuerpo del mensaje.  
Se puede omitir. Se enviará un mensaje vacío. Si se utiliza, debe ser el último parámetro, para permitir la obtención de varias líneas en el mensaje, utilizando "\n" para saltos de línea. Hace referencia a la señal "%body%".

**Nota:** Es posible que el texto del cuerpo contenga las señales "%report%" o "%fullreport%" en cualquier lugar dentro del mismo, lo que hará que se inserte en ese punto un informe en el estado de trabajo actual. Un informe completo contiene información de etapa y enlace así como estado de trabajo.

*Respuesta.* Las respuestas posibles son:

- DSJE.NOERROR (0) OK
- DSJE.NOPARAM Falta nombre de parámetro - el campo no tiene el aspecto 'nombre:valor'
- DSJE.NOTEMPLATE No se puede encontrar archivo de plantilla
- DSJE.BADTEMPLATE Error en archivo de plantilla

### Comentarios

La rutina busca un archivo local en el directorio actual de proyectos, con un nombre conocido. Es decir, una plantilla para describir exactamente cómo ejecutar el mandato sendmail local.

## Ejemplo

```
code = DSSendMail("De:yo@aquí\nPara:Tu@allí\nAsunto:Hola\nCuerpo:Línea1\nLínea2")
```

## DSSetDisableJobHandler

Utilice la función `DSSetDisableJobHandler` para habilitar o inhabilitar el manejo de mensajes a nivel de trabajo.

### Sintaxis

```
ErrCode = DSSetDisableJobHandler (DescriptorContextoTrabajo, valor)
```

*DescriptorContextotrabajo* es el descriptor de contexto para el trabajo tal como se deriva de **DSAttachJob**.

*valor* es TRUE para inhabilitar el manejo de mensajes de nivel de trabajo, o FALSE para habilitar el manejo de mensajes de nivel de trabajo.

*CódigoError* es 0 si `DSSetDisableJobHandler` es satisfactorio; de lo contrario, puede ser uno de los enteros negativos siguientes:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido
- DSJE.BADVALUE *valor* no es adecuado para ese tipo de parámetro.

## Ejemplo

El mandato siguiente inhabilita el manejo de mensajes de nivel de trabajo para el trabajo `qsales`:

```
GenErr = DSSetDisableJobHandler (qsales_handle, TRUE)
```

## DSSetDisableProjectHandler

Utilice la función `DSSetDisableProjectHandler` para habilitar o inhabilitar el manejo de mensajes a nivel de proyecto.

### Sintaxis

```
CódigoError = DSSetDisableProjectHandler (DescriptorContextoProyecto, valor)
```

*DescriptorContextoProyecto* es el valor devuelto de **DSOpenProject**.

*valor* es TRUE para inhabilitar el manejo de mensajes de nivel de proyecto, o FALSE para habilitar el manejo de mensajes de nivel de proyecto.

*CódigoError* es 0 si `DSSetDisableProjectHandler` es satisfactorio; de lo contrario, puede ser uno de los siguientes enteros negativos:

- DSJE.BADHANDLE Invalid *DescriptorContextoProyecto*.
- DSJE.BADVALUE *valor* no es adecuado para ese tipo de parámetro.

## Ejemplo

El mandato siguiente inhabilita el manejo de mensajes de nivel de proyecto para el proyecto `qsales`:

```
GenErr = DSSetDisableProjectHandler (qsales_handle, TRUE)
```

## DSSetGenerateOpMetaData

Utilice la función DSSetGenerateOpMetaData para especificar si el trabajo genera o no metadatos operativos. Esta función altera temporalmente el valor predeterminado del proyecto.

### Sintaxis

*CódigoError* = DSSetGenerateOpMetaData (*DescriptorContextoTrabajo*, *valor*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*valor* es TRUE para generar metadatos operativos, FALSE para no generar metadatos operativos.

*CódigoError* es 0 si DSRunJob es satisfactorio; de lo contrario puede ser uno de los enteros negativos siguientes:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.
- DSJE.BADTYPE *valor* es incorrecto.

### Ejemplo

El mandato siguiente hace que el trabajo qsales genere metadatos operativos independientemente de lo que especifique el valor predeterminado del proyecto:

```
GenErr = DSSetGenerateOpMetaData(qsales_handle, TRUE)
```

## DSSetJobLimit

Utilice la función DSSetJobLimit para alterar temporalmente los límites de la fila o aviso del trabajo de control. De manera predeterminada, un trabajo controlado hereda los límites de filas o avisos del trabajo de control.

### Sintaxis

*CódigoError* = DSSetJobLimit (*DescriptorTrabajoContexto*, *TipoLímite*, *ValorLímite*)

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*TipoLímite* es el nombre del límite que se aplicará al trabajo en ejecución y es uno de los siguientes:

- DSJ.LIMITWARN Trabajo que se detendrá después de *ValorLímite* sucesos de aviso.
- DSJ.LIMITROWS Etapas que se limitarán a *ValorLímite* filas.

*ValorLímite* es un entero que especifica el valor en el que se establecerá el límite. Establézcalo en 0 para especificar un número ilimitado de avisos.

*CódigoError* es 0 si DSSetJobLimit es satisfactorio; de lo contrario, es uno de los siguientes valores negativos:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido
- DSJE.BADSTATE El trabajo no está en el estado correcto (compilado, no en ejecución).
- DSJE.BADTYPE *TipoLímite* no es una condición de límite conocida.

- DSJE.BADVALUE *ValorLímite* no es adecuado para el tipo de condición de limitación.

## Ejemplo

El mandato siguiente establece un límite de 10 avisos en el trabajo *qsales* antes de que se detenga:

```
LimitErr = DSSetJobLimit(qsales_handle,
→ DSJ.LIMITWARN, 10)
```

## DSSetJobQueue

Establece la cola de gestión de carga de trabajo antes de ejecutar un trabajo.

### Sintaxis

*DSSetJobQueue (CódigoError, DescriptorContextoTrabajo, NombreCola)*

*CódigoError* se establece en uno de los siguientes valores:

- DSJE.NOERROR si *DSSetJobQueue* se ha realizado correctamente
- DSJE.BADHANDLE si el *DescriptorContextoTrabajo* no era válido

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de *DSAttachJob*.

*NombreCola* es el nombre de la cola de gestión de carga de trabajo a la que someter el trabajo.

## Ejemplo

El siguiente mandato establece la cola de gestión de carga de trabajo en *HighCPUClass*.

```
Call DSSetJobQueue(ErrCode, qsales_handle, "HighCPUClass")
```

## DSSetParam

Utilice la función *DSSetParam* para especificar los valores de parámetro de trabajo antes de ejecutar un trabajo. Cualquier parámetro que no esté establecido adoptará el valor predeterminado.

### Sintaxis

*CódigoError = DSSetParam (DescriptorContextoTrabajo, NombreParám, ValorParám)*

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de *DSAttachJob*.

*NombreParám* es una serie que proporciona el nombre del parámetro.

*ValorParám* es una serie que proporciona el valor del parámetro.

*CódigoError* es 0 si *DSSetParam* es satisfactorio; de lo contrario, puede ser uno de los siguientes enteros negativos:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido
- DSJE.BADSTATE El trabajo no está en el estado correcto (compilado, no en ejecución).
- DSJE.BADPARAM *NombreParám* no es un parámetro conocido del trabajo.

- DSJE.BADVALUE *ValorParám* no es adecuado para este tipo de parámetro.

## Ejemplo 1

Los mandatos siguientes establecen el parámetro de trimestre en 1 y el parámetro de fecha de inicio en 1/1/97 para el trabajo qsales:

```
paramerr = DSSetParam (qsales_handle, "trimestre", "1")
paramerr = DSSetParam (qsales_handle, "fecha inicial", "1997-01-01")
```

## Ejemplo 2

En este ejemplo, el trabajo qsales utiliza un conjunto de parámetros, denominado **PS1**, que contiene dos parámetros, denominado **P1** y **P2**, cuyos valores predeterminados son P1def y P2def. Existe un conjunto de valores, denominado **VSetA** definido para el conjunto de parámetros en el que **P1** y **P2** tienen los valores de P1A y P2A.

De forma predeterminada, el trabajo utiliza los valores predeterminados del conjunto de parámetros **PS1**. Los valores de parámetro que se utilizan son P1=P1def y P2=P2def

El siguiente mandato establece los valores de los parámetros del conjunto de valores **VSetA**. Los valores de parámetro que se utilizan son P1=P1A y P2=P2A.

```
paramerr = DSSetParam (qsales_handle, "PS1", "VsetA")
```

Los mandatos siguientes establecen los valores de los parámetros del conjunto de valores **VSetA** y luego sustituyen el valor del parámetro **P2** por el valor P2X. Los valores de parámetro que se utilizan son P1=P1A y P2=P2X:

```
paramerr = DSSetParam (qsales_handle, "PS1", "VsetA")
paramerr = DSSetParam (qsales_handle, "PS1.P2", "P2x")
```

## DSSetUserStatus

Utilice la rutina DSSetUserStatus para establecer un código de terminación para que lo interroge otro trabajo. Esta rutina sólo se aplica al trabajo actual, pero no emplea un parámetro *JobHandle*.

Esta rutina la puede utilizar cualquier trabajo en una rutina JobControl o After para que lo interroge otro trabajo. De hecho, el código se puede establecer en cualquier punto del trabajo y el último valor es el que se tomará en cualquier momento. Por lo tanto, para estar seguro de obtener el código de terminación real para un trabajo, el emisor de la llamada debe utilizar en primer lugar DSWaitForJob y DSGetJobInfo, comprobando si hay un estado de finalización satisfactorio.

Esta rutina se define como una subrutina, no como función, porque no hay posibles errores.

## Sintaxis

Call DSSetUserStatus (*EstadoUsuario*)

*EstadoUsuario* La serie es cualquier mensaje de terminación definido por el usuario. La serie se registrará como parte de un suceso de "control" adecuado en el registro del trabajo de llamada y se almacenará para su recuperación por parte de DSGetJobInfo, sobrescribiendo la serie almacenada anterior.



Esta serie no debe ser un entero negativo; de lo contrario es posible que no se distinga de un error interno en llamadas a DSGetJobInfo.

## Ejemplo

El mandato siguiente establece un código de terminación de "trabajo de venta finalizado":

```
Call DSSetUserStatus("trabajo de venta finalizado")
```

## DSStopJob

Utilice la función DSStopJob para enviar inmediatamente una solicitud de detención al motor en tiempo de ejecución. Utilice esta función sólo después de que se emita DSRunJob.

La llamada es asíncrona. Si necesita saber que el trabajo se ha detenido realmente, debe llamar a DSWaitForJob o utilizar la sentencia de Suspensión y buscar DSGetJobStatus. Se envía la solicitud de detención independientemente del estado actual del trabajo.

### Sintaxis

```
CódigoError = DSStopJob (DescriptorContextoTrabajo)
```

*DescriptorContextoTrabajo* es el descriptor de contexto para el trabajo tal como se deriva de DSAttachJob.

*CódigoError* es 0 si DSStopJob es satisfactorio; de lo contrario puede ser uno de los siguientes:

- DSJE.BADHANDLE *DescriptorContextoTrabajo* no válido.

## Ejemplo

El siguiente mandato solicita que se detenga el trabajo qsales:

```
stoperr = DSStopJob(qsales_handle)
```

## DSTransformError

Utilice la función DSTransformerError para registrar un mensaje de aviso a un archivo de registro de trabajo. Se llama a esta función desde transformaciones solamente.

### Sintaxis

```
Call DSTransformError (Mensaje, NombreTransformación)
```

*Mensaje* (entrada) es el mensaje de aviso que desea registrar. *Mensaje* recibe automáticamente como prefijo el nombre de la etapa actual y la transformación de llamada.

*NombreTransformación* (entrada) es el nombre de la transformación que llama a la subrutina DSTransformError.

### Comentarios

DSTransformError graba el mensaje (y otra información) en el archivo de registro del trabajo como aviso y vuelve a la transformación. Si el trabajo tiene un límite de

aviso definido para el mismo, cuando el número de avisos alcanza ese límite, la llamada no devuelve el valor de retorno y el trabajo termina anormalmente.

Además del mensaje de aviso, DSTransformError registra los valores de todas las columnas en las filas actuales para todos los enlaces de entrada y salida conectados a la etapa actual.

### Ejemplo

```
Function MySqrt(Arg1)
If Arg1 < 0 Then
    Call DSTransformError("Valor negativo:"Arg1, "MySqrt")
    Return("0")    ;*la transformación produce 0 en este caso
End
Result = Sqrt(Arg1) ;* de lo contrario devuelve la raíz cuadrada
Return(Result)
```

## DSTranslateCode

Utilice la función DSTranslateCode para convertir un estado de control de trabajos o un

### Sintaxis

*Ans* = DSTranslateCode(*Código*)

*Código* es:

- Si *Código* > 0, se presupone que es un estado de trabajo.
- Si *Código* < 0, se presupone que es un código de error.
- (No se debe pasar nunca 0 y devolverá "ningún error")

*Ans* es el mensaje asociado con el código.

### Comentarios

Si *Código* no se reconoce, se notificará *Ans*.

### Ejemplo

```
code$ = DSGetLastErrorMsg()
ans$ = DSTranslateCode(code$)
```

## DSWaitForFile

Utilice la función DSWaitForFile para suspender un trabajo hasta que exista o no exista un archivo nombrado.

### Sintaxis

*Respuesta* = DSWaitForFile(*Parámetros*)

*Parámetros* es la vía de acceso completa del archivo que se debe esperar. No se realiza ninguna comprobación respecto a si se trata de una vía de acceso razonable (por ejemplo, si existen todos los directorios de la vía de acceso). Un nombre de vía de acceso que empieza con "-" indica un distintivo para comprobar la no-existencia de la vía de acceso. No forma parte del nombre de vía de acceso.

Los parámetros también pueden finalizar con el formato "timeout:NNNN" (o "timeout=NNNN"). Esto indica un tiempo no predeterminado que se debe esperar antes de abandonar. Existen varios formatos posibles, y no distinguen entre mayúsculas y minúsculas:

- nnn número de segundos a esperar (a partir de ahora)
- nnnS ídem
- nnnM número de minutos que se debe esperar (desde ahora)
- nnnH número de horas que se debe esperar (desde ahora)
- nn:nn:nn esperar hasta esta hora en 24HH:NN:SS. Si se ha pasado esta hora o nn:nn, esperará hasta el día siguiente.

El tiempo de espera predeterminado es el mismo que "12H".

El formato puede opcionalmente terminar con "/nn", lo que indica un tiempo de retardo de sondeo en segundos. Si se omite, se utiliza un tiempo de sondeo predeterminado.

*Respuesta* puede ser:

- DSJE.NOERROR (0) OK - El archivo ahora existe o no existe, dependiendo del distintivo.
- DSJE.BADTIME Formato de tiempo de espera no reconocido
- DSJE.NOFILEPATH Falta vía de acceso de archivo
- DSJE.TIMEOUT Se ha esperado demasiado

## Ejemplos

`Respuesta = DSWaitForFile("C:\ftp\incoming.txt timeout:2H")`

(espera 7200 segundos a que exista el archivo en C: antes de abandonar.)

`Respuesta = DSWaitForFile("-incoming.txt timeout=15:00")`

(espera hasta las 3 de la tarde a que NO exista el archivo en el directorio local.)

`Respuesta = DSWaitForFile("incoming.txt timeout:3600/60")`

(espera 1 hora a que exista un archivo local, comprobándolo una vez por minuto.)

## DSWaitForJob

Utilice la función DSWaitForJob para que un trabajo o trabajos esperen antes de que el proceso continúe en la siguiente sentencia BASIC.

Esta función sólo es válida si el trabajo actual ha emitido un DSRunJob en los *DescriptorContextoTrabajo* proporcionados. Si uno de los trabajos cuyos descriptores de contexto se encuentran en la lista ha finalizado, la función DSWaitForJob devuelve inmediatamente su resultado. Si ninguno de los trabajos ha finalizado, la función DSWaitForJob devuelve su resultado tan pronto como finaliza uno de los trabajos.

## Sintaxis

`CódigoError = DSWaitForJob (DescriptorContextoTrabajo)`

*DescriptorContextoTrabajo* es la serie devuelta de DSAttachJob. Si contiene comas, *JobHandle* es un conjunto delimitado por comas de descriptores de contexto de trabajo, que representa una lista de trabajos a los que se debe esperar.

*CódigoError* es 0 sino hay ningún error, de lo contrario los posibles valores de error (<0) son:

- DSJE.BADHANDLE DescriptorContextoTrabajo incorrecto.

- **DSJE.WRONGJOB** El trabajo para este *DescriptorContextoTrabajo* no se ha ejecutado desde dentro de este trabajo.

*CódigoError* es >0 => el descriptor de contexto del trabajo ha finalizado desde una espera de varios trabajos.

## Comentarios

*DSWaitForJob* espera un único trabajo o varios trabajos.

## Ejemplo

Para esperar el retorno del trabajo *qsales*:

```
WaitErr = DSWaitForJob(qsales_handle)
```

---

## Macros de estado de trabajo

Utilice macros que se facilitan en el archivo *JOBCONTROL.H* para obtener información sobre el trabajo actual, así como enlaces y etapas que pertenecen al trabajo actual.

Las macros que se facilitan en el archivo *JOBCONTROL.H* proporcionan la posibilidad de utilizar las funciones de BASIC **DSGetProjectInfo**, **DSGetJobInfo**, **DSGetStageInfo** y **DSGetLinkInfo** de InfoSphere DataStage con la señal de *DSJ.ME*, dado que *DescriptorContextoTrabajo* se puede utilizar en todas las etapas activas y las subrutinas anteriores/posteriores. Las macros proporcionan la funcionalidad para todos los argumentos posibles de *TipoInfo* para las funciones **DSGet...Info**.

Las macros disponibles son:

- **DSHostName**
- **DSProjectName**
- **DSJobStatus**
- **DSJobName**
- **DSJobController**
- **DSJobStartDate**
- **DSJobStartTime**
- **DSJobWaveNo**
- **DSJobInvocations**
- **DSJobInvocationId**
- **DSStageName**
- **DSStageLastErr**
- **DSStageType**
- **DSStageInRowNum**
- **DSStageVarList**
- **DSLLinkRowCount**
- **DSLLinkLastErr**
- **DSLLinkName**

Por ejemplo, para obtener el nombre del trabajo actual:

```
MiNombre = DSJobName
```

Para obtener el nombre de etapa actual:

```
MiNombre = DSJobName : "." : DSStageName
```

Además, se proporcionan las macros siguientes para manipular variables de etapa Transformer:

- **DSGetVar**(*NombreVar*) devuelve el valor actual de la variable de etapa nombrada. Si la etapa actual no tiene una variable de etapa denominada *NombreVar*, se devuelve "" y se registra un mensaje de error. Si la variable de etapa nombrada está definida pero no se ha inicializado, se devuelve "" y se registra un mensaje de error.
- **DSSetVar**(*NombreVar*, *ValorVar*) establece el valor de la variable de etapa nombrada. Si la etapa actual no tiene una variable de etapa denominada *NombreVar*, se registra un mensaje de error.



---

## Capítulo 3. Generación de un informe XML

Puede crear un informe XML que proporcione información sobre un trabajo mediante los métodos siguientes:

- Función de la API DSMakeJobReport (consulte “DSMakeJobReport” en la página 69)
- Función de BASIC DSMakeJobReport (consulte “DSMakeJobReport” en la página 123)
- Mandato dsjob (consulte “Generación de un informe” en la página 14)

InfoSphere DataStage proporciona los archivos siguientes para ayudarle en el manejo de informes XML generados:

- **DSReportSchema.xsd**. Un documento de esquema XML que describe completamente la estructura de los documentos de informe de trabajo XML.
- **DSReport-Monitor.xsl**. Una hoja de estilo XSLT de ejemplo que crea una página web HTML similar a la vista del Supervisor del Director del informe XML.
- **DSReport-Waterfall.xsl**. Una hoja de estilo XSLT de ejemplo que crea una página web HTML que muestra un informe en cascada que describe cómo fluyen los datos entre todos los procesos del trabajo desde el informe XML.

Todos los archivos están ubicados en el directorio del cliente de InfoSphere DataStage (\IBM\InformationServer\Clients\Classic).

Puede incorporar una referencia a una hoja de estilo al crear un informe utilizando cualquiera de los mandatos listados anteriormente. Después de que se haya generado el informe, puede visualizarlo en un navegador de Internet.

De forma alternativa, puede utilizar un procesador xslt como por ejemplo saxon o msxsl para convertir un informe ya generado. Por ejemplo:

```
java - jar saxon.jar informetrabajo.xml DSReport-Monitor.xsl > supervisortrabajo.htm
```

generaría un archivo HTML denominado supervisortrabajo.htm a partir del informe informetrabajo.xml, mientras que:

```
maxsl informetrabajo.xml DSReport-Waterfall.xsl > cascadatrabajo.htm
```

generaría un archivo HTML denominado cascadatrabajo.htm a partir del informe informetrabajo.xml.





---

## Apéndice A. Accesibilidad de los productos

Puede obtener información sobre el estado de accesibilidad de los productos de IBM.

Los módulos de producto y las interfaces de usuario de IBM InfoSphere Information Server no son totalmente accesibles.

Para obtener información sobre el estado de accesibilidad de los productos de IBM, consulte la información de accesibilidad de productos de IBM en [http://www.ibm.com/able/product\\_accessibility/index.html](http://www.ibm.com/able/product_accessibility/index.html).

### Documentación sobre accesibilidad

Se proporciona documentación accesible para los productos en IBM Knowledge Center. IBM Knowledge Center presenta la documentación en formato XHTML 1.0, que se puede ver en la mayoría de navegadores web. Dado que IBM Knowledge Center utiliza XHTML, puede establecer preferencias de visualización en el navegador. Esto también le permite utilizar lectores de pantalla y otras tecnologías de asistencia para acceder a la documentación.

La documentación que está en IBM Knowledge Center se proporciona en archivos PDF, que no son totalmente accesibles.

### IBM y la accesibilidad

Consulte el sitio web IBM Human Ability and Accessibility Center para obtener más información sobre el compromiso de IBM con la accesibilidad.



---

## Apéndice B. Lectura de la sintaxis de la línea de mandatos

Esta documentación utiliza caracteres especiales para definir la sintaxis de la línea de mandatos.

Los siguientes caracteres especiales definen la sintaxis de la línea de mandatos:

- [ ] Identifica un argumento opcional. Se necesitan los argumentos que no están entre delimitadores.
- ... Indica que puede especificar varios valores para el argumento anterior.
- | Indica información que se excluye mutuamente. Puede utilizar el argumento a la izquierda del separador o el argumento a la derecha del separador. No puede utilizar los dos argumentos en un único uso del mandato.
- { } Delimita un conjunto de argumentos que se excluyen mutuamente cuando se necesita uno de los argumentos. Si los argumentos son opcionales, se escriben entre delimitadores ([ ]).

### Nota:

- El número máximo de caracteres de un argumento es de 256.
- Escriba los valores de argumentos que tengan espacios incrustados entre comillas simples o dobles.

Por ejemplo:

```
wsetsrc[-S server] [-l label] [-n name] origen
```

El argumento *origen* es el único argumento necesario para el mandato **wsetsrc**. Los delimitadores de los otros argumentos indican que dichos argumentos son opcionales.

```
wlsac [formato -l | -f] [clave...] perfil
```

En este ejemplo, los argumentos de formato -l y -f se excluyen mutuamente y son opcionales. El argumento *perfil* es necesario. El argumento *clave* es opcional. La elipsis (...) que sigue al argumento *clave* indica que puede especificar varios nombres de clave.

```
wrb -import {rule_pack | rule_set}...
```

En este ejemplo, los argumentos *rule\_pack* y *rule\_set* se excluyen mutuamente, pero debe especificarse uno de ellos. Además, los puntos suspensivos (...) indican que puede especificar varios paquetes de reglas y conjuntos de reglas.



---

## Apéndice C. Cómo ponerse en contacto con IBM

Puede ponerse en contacto con IBM para obtener soporte al cliente, servicios de software, información sobre productos e información general. También puede facilitar comentarios a IBM sobre los productos y la documentación.

En la tabla siguiente se listan los recursos para soporte al cliente, servicios de software, formación e información sobre productos y soluciones.

*Tabla 6. Recursos de IBM*

Recurso	Descripción y ubicación
Portal de soporte de IBM	Puede personalizar la información de soporte eligiendo los productos y los temas que le interesen en <a href="http://www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server">www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server</a>
Servicios de software	Puede encontrar información sobre servicios de software, de tecnologías de la información y de consultoría de negocio en el sitio de soluciones, en <a href="http://www.ibm.com/businesssolutions/">www.ibm.com/businesssolutions/</a>
Mi IBM	Puede gestionar enlaces a sitios web de IBM y a información que satisfaga sus necesidades específicas de soporte técnico creando una cuenta en el sitio Mi IBM en <a href="http://www.ibm.com/account/">www.ibm.com/account/</a>
Formación y certificación	Puede obtener información sobre formación técnica y servicios de educación diseñados para personas, empresas y organizaciones públicas, a fin de adquirir, mantener y optimizar sus habilidades de TI en <a href="http://www.ibm.com/training">http://www.ibm.com/training</a>
Representantes de IBM	Puede contactar con un representante de IBM para obtener información sobre soluciones en <a href="http://www.ibm.com/connect/ibm/us/en/">www.ibm.com/connect/ibm/us/en/</a>



---

## Apéndice D. Acceso a la documentación del producto

La documentación se proporciona en diversos formatos: en el IBM Knowledge Center en línea, en un centro de información opcional instalado localmente y como manuales PDF. Puede acceder a la ayuda en línea o instalada localmente directamente desde las interfaces de cliente del producto.

IBM Knowledge Center es el mejor lugar para encontrar la información más actualizada de InfoSphere Information Server. IBM Knowledge Center contiene ayuda para la mayoría de las interfaces del producto, así como documentación completa para todos los módulos de producto de la suite. Puede abrir IBM Knowledge Center desde el producto instalado o desde un navegador web.

### Cómo acceder a IBM Knowledge Center

Existen varias maneras de acceder a la documentación en línea:

- Pulse el enlace **Ayuda** en la parte superior derecha de la interfaz de cliente.
- Pulse la tecla F1. Normalmente, la tecla F1 abre el tema que describe el contexto actual de la interfaz de cliente.

**Nota:** La tecla F1 no funciona en clientes web.

- Escriba la dirección en un navegador web, por ejemplo, cuando no tenga iniciada una sesión en el producto.

Escriba la siguiente dirección para acceder a todas las versiones de la documentación de InfoSphere Information Server:

<http://www.ibm.com/support/knowledgecenter/SSZJPZ/>

Si desea acceder a un tema concreto, especifique el número de versión con el identificador de producto, el nombre del plug-in de documentación y la vía de acceso al tema en el URL. Por ejemplo, el URL para la versión 11.3 de este tema es el siguiente. (El símbolo  $\Rightarrow$  indica una continuación de línea):

[http://www.ibm.com/support/knowledgecenter/SSZJPZ\\_11.3.0/com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html](http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html)

#### Consejo:

El Knowledge Center tiene también un URL corto:

<http://ibm.biz/knowctr>

Para especificar un URL corto a una página de producto, versión o tema específico, utilice un carácter de almohadilla (#) entre el URL corto y el identificador de producto. Por ejemplo, el URL corto a toda la documentación de InfoSphere Information Server es el siguiente URL:

<http://ibm.biz/knowctr#SSZJPZ/>

Y el URL corto al tema anterior para crear un URL ligeramente más corto es el siguiente URL (El símbolo  $\Rightarrow$  indica una continuación de línea):

[http://ibm.biz/knowctr#SSZJPZ\\_11.3.0/com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html](http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html)

## Cambiar los enlaces de ayuda para que hagan referencia a la documentación instalada localmente

IBM Knowledge Center contiene la versión más actualizada de la documentación. Sin embargo, puede instalar una versión local de la documentación como un centro de información y configurar los enlaces de ayuda para que apunten a él. Un centro de información local es útil si su empresa no proporciona acceso a Internet.

Siga las instrucciones de instalación que vienen con el paquete de instalación del centro de información para instalarlo en el sistema que elija. Después de instalar e iniciar el centro de información, puede utilizar el mandato **iisAdmin** en el sistema de la capa de servicios para cambiar la ubicación de la documentación a la que hacen referencia la tecla F1 y los enlaces de ayuda del producto. (El símbolo  $\Rightarrow$  indica una continuación de línea):

### Windows

```
vía_instalación_IS\ASBServer\bin\iisAdmin.bat -set -key  $\Rightarrow$   
com.ibm.iis.infocenter.url -value http://<host>:<puerto>/help/topic/
```

### AIX Linux

```
vía_instalación_IS/ASBServer/bin/iisAdmin.sh -set -key  $\Rightarrow$   
com.ibm.iis.infocenter.url -value http://<host>:<puerto>/help/topic/
```

Donde <host> es el nombre del sistema donde está instalado el centro de información y <puerto> es el número de puerto para el centro de información. El número de puerto predeterminado es 8888. Por ejemplo, en un sistema llamado server1.example.com que utilice el puerto predeterminado, el valor del URL sería <http://server1.example.com:8888/help/topic/>.

## Obtener la documentación en PDF y en copia impresa

- Los manuales en archivos PDF están disponibles en línea y puede accederse a ellos desde este documento de soporte: <https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1>.
- También puede solicitar publicaciones de IBM en formato impreso en línea o a través de su representante local de IBM. Para solicitar publicaciones en línea, vaya al Centro de Publicaciones de IBM en <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.



---

## Apéndice E. Cómo aportar comentarios sobre la documentación del producto

Puede aportar valiosos comentarios en relación a la documentación de IBM.

Sus comentarios ayudarán a IBM a ofrecer información de calidad. Puede utilizar cualquiera de los métodos siguientes para enviar sus comentarios:

- Para proporcionar un comentario acerca de un tema del IBM Knowledge Center que está alojado en el sitio web de IBM, inicie la sesión y pulse el botón **Añadir comentario** en la parte inferior del tema. Los comentarios enviados de esta manera serán visibles para todos los usuarios.
- Para enviar un comentario acerca de un tema del IBM Knowledge Center a IBM y que ningún otro usuario pueda ver, inicie la sesión y pulse en el enlace **Comentarios** en la parte inferior del IBM Knowledge Center.
- Envíe sus comentarios utilizando el formulario de comentarios del lector que encontrará en [www.ibm.com/software/awdtools/rcf/](http://www.ibm.com/software/awdtools/rcf/).
- Envíe sus comentarios por correo electrónico a [comments@us.ibm.com](mailto:comments@us.ibm.com). Incluya el nombre y el número de versión del producto, así como el nombre y el número de pieza de la información (si es pertinente). Si su comentario es sobre un texto específico, incluya la ubicación del texto (por ejemplo, un título, un número de tabla o un número de página).



---

## Avisos y marcas registradas

Esta información ha sido desarrollada para productos y servicios ofrecidos en los Estados Unidos. Este material puede estar disponible en IBM en otros idiomas. Sin embargo, es posible que deba tener una copia del producto o de la versión del producto en ese idioma para poder acceder al mismo.

### Avisos

Es posible que IBM no ofrezca en otros países los productos, servicios o características que se describen en este documento. Póngase en contacto con el representante local de IBM para obtener información acerca de los productos y servicios que actualmente están disponibles en su localidad. Cualquier referencia a un producto, programa o servicio de IBM no implica ni establece que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes de aprobación que cubran temas tratados en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por escrito, a la siguiente dirección:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785 EE.UU.

Para realizar consultas relativas a la información de juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokio 103-8510, Japón

**El párrafo siguiente no se aplica en el Reino Unido ni en ningún otro país en el que las disposiciones en él expuestas sean incompatibles con la legislación local:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍAS DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN E IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información incluida en este documento está sujeta a cambios periódicos, que se

incorporarán en nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Cualquier referencia hecha en esta información a sitios web que no sean de IBM se proporciona únicamente para su comodidad y no debe considerarse en modo alguno como una aprobación de dichos sitios web. Los materiales de estos sitios web no forman parte de los materiales de este producto de IBM y el uso que haga de estos sitios web es de la entera responsabilidad del usuario.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información acerca del mismo con el fin de: (i) intercambiar la información entre los programas creados independientemente y otros programas (incluido éste) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003 EE.UU.

Esta información puede estar disponible, sujeta a los términos y condiciones pertinentes, incluido en algunos casos el pago de una cantidad determinada.

IBM proporciona el programa bajo licencia descrito en este documento, y todo el material bajo licencia disponible para el mismo, bajo los términos del Acuerdo de cliente de IBM, el Acuerdo internacional de licencia de programa de IBM o cualquier otro acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se determinaron en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse efectuado en sistemas a nivel de desarrollo, y no existe ninguna garantía de que dichas mediciones sean las mismas en sistemas de disponibilidad general. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relacionada con productos no de IBM se ha obtenido de los suministradores de dichos productos, de sus anuncios publicados o de otras fuentes de información pública disponibles. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con productos que no son de IBM. Las consultas acerca de las prestaciones de los productos que no son de IBM deben dirigirse a los suministradores de tales productos.

Todas las declaraciones relativas a la dirección o intención futura de IBM están sujetas a cambios o anulación sin previo aviso y representan únicamente metas y objetivos.

Esta información se suministra sólo con fines de planificación. La presente información esta sujeta a cambios antes de que los productos que en ella se describen estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en las operaciones de negocios diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es totalmente casual.

#### LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en idioma de origen, que ilustra las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma, sin pagar a IBM, con la finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Estos ejemplos no se han probado bajo todas las condiciones posibles. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, capacidad de servicio o funcionamiento de esos programas. Los programas de ejemplo se suministran "TAL CUAL", sin garantía de ninguna clase. IBM no se hará responsable de los daños que puedan derivarse del uso de los programas de ejemplo.

Cada copia, parcial o completa, de estos programas de ejemplo o cualquier trabajo derivado, debe incluir un aviso de copyright como el siguiente:

© (el nombre de su empresa) (año). Partes de este código provienen de programas de ejemplo de IBM Corp. © Copyright IBM Corp. \_escriba el año o años\_. Reservados todos los derechos.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

### Consideraciones sobre la política de privacidad

Los productos de software de IBM, incluidas las soluciones de software como servicio, ("Ofertas de software"), pueden utilizar cookies u otras tecnologías para recopilar información sobre el uso de productos, para ayudar a mejorar la experiencia del usuario final, para personalizar las interacciones con el usuario final o para otros fines. En muchos casos, las Ofertas de software no recopilan información de identificación personal. Algunas de nuestras Ofertas de software pueden ayudarle a recopilar información de identificación personal. Si esta Oferta de software utiliza cookies para recopilar información de identificación personal, la información específica sobre el uso de cookies por parte de esta oferta se expone más abajo.

Dependiendo de las configuraciones desplegadas, esta Oferta de software puede utilizar cookies de sesión o persistentes. Si un producto o componente no está en la lista, ese producto o componente no utiliza cookies.

*Tabla 7. Uso de cookies de los productos y componentes de InfoSphere Information Server*

Módulo de producto	Componente o característica	Tipo de cookie que se utiliza	Recopilar estos datos	Finalidad de los datos	Inhabilitación de las cookies
Cualquiera (parte de la instalación de InfoSphere Information Server)	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	Nombre de usuario	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> </ul>	No se pueden inhabilitar

Tabla 7. Uso de cookies de los productos y componentes de InfoSphere Information Server (continuación)

Módulo de producto	Componente o característica	Tipo de cookie que se utiliza	Recopilar estos datos	Finalidad de los datos	Inhabilitación de las cookies
Cualquiera (parte de la instalación de InfoSphere Information Server)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	Ninguna información de identificación personal	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> <li>Usabilidad de usuario mejorada</li> <li>Configuración de inicio de sesión único</li> </ul>	No se pueden inhabilitar
InfoSphere DataStage	Etapas Big Data File	<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	<ul style="list-style-type: none"> <li>Nombre de usuario</li> <li>Firma digital</li> <li>ID de sesión</li> </ul>	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> <li>Configuración de inicio de sesión único</li> </ul>	No se pueden inhabilitar
InfoSphere DataStage	Etapas XML	Sesión	Identificadores internos	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> </ul>	No se pueden inhabilitar
InfoSphere DataStage	Consola de operaciones de IBM InfoSphere DataStage and QualityStage	Sesión	Ninguna información de identificación personal	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> </ul>	No se pueden inhabilitar
InfoSphere Data Click	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	Nombre de usuario	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> </ul>	No se pueden inhabilitar
InfoSphere Data Quality Console		Sesión	Ninguna información de identificación personal	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> <li>Configuración de inicio de sesión único</li> </ul>	No se pueden inhabilitar
InfoSphere QualityStage Standardization Rules Designer	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	Nombre de usuario	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> </ul>	No se pueden inhabilitar
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> <li>Sesión</li> <li>Persistente</li> </ul>	<ul style="list-style-type: none"> <li>Nombre de usuario</li> <li>Identificadores internos</li> <li>Estado del árbol</li> </ul>	<ul style="list-style-type: none"> <li>Gestión de sesiones</li> <li>Autenticación</li> <li>Configuración de inicio de sesión único</li> </ul>	No se pueden inhabilitar
InfoSphere Information Analyzer	Etapas Reglas de datos en el cliente del Diseñador de InfoSphere DataStage and QualityStage	Sesión	ID de sesión	Gestión de sesiones	No se pueden inhabilitar

Si las configuraciones desplegadas para esta Oferta de software le ofrecen como cliente la posibilidad de recopilar información de identificación personal de los usuarios finales mediante cookies y otras tecnologías, debe buscar asesoramiento jurídico sobre la legislación aplicable a dicha recopilación de datos, incluidos los requisitos de notificación y consentimiento.

Para obtener más información sobre el uso de diversas tecnologías, incluidas las cookies, para estos fines, consulte la Política de privacidad de IBM en <http://www.ibm.com/privacy>, la sección “Cookies, balizas web y otras tecnologías” de la Declaración de privacidad en línea de IBM en <http://www.ibm.com/privacy/details> y la “Declaración de privacidad de productos de software y software como servicio de IBM” (en inglés) en <http://www.ibm.com/software/info/product-privacy>.

## **Marcas registradas**

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas comerciales o marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actual de las marcas registradas de IBM en el sitio web [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Los términos siguientes son marcas comerciales o marcas registradas de otras empresas:

Adobe es una marca registrada de Adobe Systems Incorporated en los Estados Unidos y/o en otros países.

Intel e Itanium son marcas comerciales o marcas registradas de Intel Corporation o sus filiales en los Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/ en otros países.

Microsoft, Windows y Windows NT son marcas registradas de Microsoft Corporation en los Estados Unidos y/o en otros países.

UNIX es una marca registrada de The Open Group en los Estados Unidos y en otros países.

Java<sup>™</sup> y todas las marcas registradas y logotipos basados en Java son marcas comerciales o marcas registradas de Oracle y/o sus filiales.

El Servicio de correos de Estados Unidos (United States Postal Service) es propietario de las siguientes marcas registradas: CASS, CASS Certified, DPV, LACS<sup>Link</sup>, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS y United States Postal Service. IBM Corporation es un licenciatario no exclusivo de DPV y LACS<sup>Link</sup> del Servicio de correos de Estados Unidos.

Otros nombres de empresas, productos y servicios pueden ser marcas comerciales o marcas de servicio de terceros.





---

# Índice

## Caracteres Especiales

\_STDC\_ token 38

## A

accesibilidad de los productos  
    accesibilidad 139  
administración  
    funciones de la API de C 41  
API, consulte la API de InfoSphere  
    DataStage 37  
API de DataStage  
    archivo de cabecera 37  
    códigos de error 96  
    compilación de aplicaciones 39  
    ejemplo de lógica de programación 38  
    funciones 40  
    redistribución de programas 40  
archivo .dsx 23  
    importación desde 23  
    listado  
        archivos dsx 27  
archivo de cabecera dsapi.h  
    descripción 37  
    incluyendo 39  
archivos de biblioteca 39  
archivos redistribuibles 40  
avisos 12  
avisos legales 149

## B

biblioteca vmdsapi.lib 39

## C

caracteres especiales  
    sintaxis de la línea de mandatos 141  
cifrado de contraseña  
    mandatos dsadmin 15  
    mandatos dsjob 2  
cifrar credenciales  
    mandatos dsadmin 15  
    mandatos dsjob 2  
cláusula de inicio de sesión 3, 16  
CLI, consultar CLI de InfoSphere  
    DataStage 1  
CLI de DataStage  
    cláusula de inicio de sesión 3, 16  
    códigos de finalización 1  
    utilización para la ejecución de trabajos 4  
    visión general 1  
contraseñas, establecimiento 78  
credenciales de usuario, cifrar  
    mandatos dsadmin 15

credenciales de usuario, cifrar  
    (continuación)  
    mandatos dsjob 2

## D

DataStage Development Kit 37  
    códigos de error 96  
    estructuras de datos 80  
    funciones de la API 40  
    grabación de programas de API 38  
    mandato dsjob 4  
datos de resultado  
    almacenamiento 38  
    reutilización 38  
descriptor de contexto de trabajo 71  
directorio dsdk 39  
DLL 40  
documentación del producto  
    acceder 145  
DSXImportService 23

## E

enlaces  
    funciones utilizadas para el  
        acceso 41  
        listado 7  
    recuperación de información 53  
    visualización de la información 10  
entradas de registro  
    añadir 11, 68  
    aviso 12  
    búsqueda de más reciente 13  
    control de proceso por lotes 12  
    error muy grave 12  
    filas rechazadas 12  
    funciones utilizadas para el  
        acceso 41  
        nuevas líneas en 69  
    recuperación 46, 48, 56  
    recuperación específica 13, 54  
    tipos 46  
    trabajo iniciado 12  
    trabajo restablecido 12  
entradas de registro de errores muy graves 12  
entradas de registro de información 12  
entradas de registro de proceso por lotes 12  
errores  
    API de DataStage 96  
    funciones utilizadas para el  
        manejo 42  
    recuperación de texto del mensaje 52  
    recuperación de valores 52  
estructura de datos DSCUSTINFO 81  
estructura de datos DSJOBINFO 82  
    y DSGetJobInfo 51  
    y DSGetLinkInfo 54

estructura de datos DSLINKINFO 85  
estructura de datos DSLOGDETAIL 86  
estructura de datos DSLOGEVENT 88  
estructura de datos DSPARAM 88  
estructura de datos DSPARAMINFO 90  
    y DSGetParamInfo 58  
estructura de datos  
    DSPROJECTINFO 91  
    y DSGetProjectInfo 59  
estructura de datos DSREPOSINFO 92  
estructura de datos DSREPOSUSAGE 93  
estructura de datos DSSTAGEINFO 94  
    y DSGetStageInfo 64  
estructura de datos DSVARINFO 95  
estructuras de datos  
    cómo se utilizan 38  
    descripción 82  
    resumen de uso 80  
etapas  
    funciones utilizadas para el  
        acceso 41  
    listado 7  
    recuperación de información 63  
    visualización de la información 10

## F

filas rechazadas 12  
función DSCloseProject 44  
función DSFindFirstLogEntry 46  
función DSFindNextLogEntry 46, 48  
función DSGetCustInfo 49  
función DSGetJobInfo 50, 107  
    y trabajos controlados 51  
función DSGetLastError 52  
función DSGetLastErrorMsg 52  
función DSGetLinkInfo 53, 110  
función DSGetLogEntry 54, 111  
función DSGetLogEventIds 56, 113  
función DSGetLogSummary 114  
función DSGetNewestLogId 115  
función DSGetParamInfo 58, 115  
función DSGetProjectInfo 59, 117  
función DSGetProjectList 60  
función DSGetStageInfo 63, 118  
función DSGetVarInfo 65, 120  
función DSLockJob 68  
función DSLogEvent 68, 120  
función DSLogFatal 121  
función DSMakeJobReport 69  
función DSOpenJob 70  
función DSOpenProject 71  
función DSRunJob 72  
función DSSetDisableJobHandler 127  
función DSSetDisableProjectHandler 127  
función DSSetGenerateOpMetaData 74, 128  
función DSSetJobLimit 74, 128  
función DSSetParam 75  
función DSSetServerParams 78  
función DSStopJob 79, 131

función DSTransformError 131  
función DSUnlockJob 79  
función DSWaitForJob 80  
funciones, API de DataStage 40

## H

hebras  
    utilización de múltiples 38  
    y almacenamiento de errores 38  
    y datos de resultado 38  
    y DSFindFirstLogEntry 48  
    y DSFindNextLogEntry 48  
    y DSGetLastErrorMsg 52  
    y entradas de registro 48  
    y errores 52

## I

importar objetos 23  
interfaz de control de trabajos 37  
interfaz de línea de mandatos 1

## L

límites 74  
límites de aviso 4, 74  
límites de fila 4, 74  
listado de archivos dsx 27

## M

macro DSHostName 134  
macro DSJobController 134  
macro DSJobInvocationId 134  
macro DSJobInvocations 134  
macro DSJobName 134  
macro DSJobStartDate 134  
macro DSJobStartTime 134  
macro DSJobStatus 134  
macro DSJobWaveNo 134  
macro DSLinkLastErr 134  
macro DSLinkName 134  
macro DSLinkRowCount 134  
macro DSProjectName 134  
macro DSStageInRowNum 134  
macro DSStageLastErr 134  
macro DSStageName 134  
macro DSStageType 134  
macro DSStageVarList 134  
macros, estado de trabajo 1  
macros de estado de trabajo 1  
mandato dsadmin 14  
mandato dsjob  
    descripción 1  
mandato DSXImportService 23, 27  
mandato SyncProject 27  
mandatos  
    dsadmin 14  
    dsjob 1  
    sintaxis 141  
    SyncProject 27  
mandatos dsadmin  
    credenciales cifradas 15  
    solicitud de credenciales 15

mandatos dsjob  
    credenciales cifradas 2  
    solicitud de credenciales 2  
    valores de parámetro cifrados 2  
marcas registradas  
    lista de 149  
motor del servidor de DataStage 3, 16

## N

nombres de motor, establecimiento 78  
nombres de usuario, establecimiento 78  
nuevas líneas en entradas de registro 69

## O

opción authfile  
    utilizar en mandatos dsadmin 15  
    utilizar en mandatos dsjob 2  
opción file  
    utilizar en mandatos dsadmin 15  
    utilizar en mandatos dsjob 2

## P

Parámetro de tipo de suceso 46  
parámetros, consulte parámetros del trabajo 58  
parámetros del trabajo  
    establecimiento 75  
    funciones utilizadas para el acceso 41  
    listado 8  
    recuperación de información 58  
    visualización de la información 11  
proyecto  
    crear una copia de seguridad 35  
    incoherencias 29  
    reconstruir 34  
    reparar incoherencias 31  
    restaurar 36  
proyectos  
    apertura 71  
    cierre 44  
    funciones utilizadas para el acceso 40  
    listado 7, 60

## S

señal \_cplusplus 38  
señal WIN32 38  
señales  
    \_cplusplus 38  
    \_STDC\_ 38  
    WIN32 38  
servicios de software  
    contactar 143  
sintaxis  
    línea de mandatos 141  
sintaxis de línea de mandatos  
    convenios 141  
soporte  
    cliente 143

soporte al cliente  
    contactar 143  
subrutina DSSetUserStatus 130  
SyncProject 31, 33  
    comprobación de incoherencias en los proyectos 29  
    creación de una copia de seguridad de un proyecto 35  
    parámetros de autenticación 28  
    reconstruir un proyecto 34  
    restaurar un proyecto 36

## T

trabajos  
    apertura 70  
    bloqueo 68  
    desbloqueo 79  
    detención 6, 79  
    ejecución 4, 72  
    en espera de finalización 80  
funciones utilizadas para el  
    acceso 40  
listado 7, 59  
recuperación de estado 50  
reinicio 72  
restablecimiento 4, 72  
validación 4, 72  
visualización de la información 9

## V

vmdsapi.dll 40





Impreso en España

SC43-1221-00



Spine information:

IBM InfoSphere DataStage    **Versión 11 Release 3**

**Guía del programador**

