
Part 1. IBM VideoCharger Extender for DB2 Universal Database

Chapter 1. IBM VideoCharger Extender for DB2 Universal Database	3
DB2 extender concepts	3
User-defined types (UDTs)	3
User-defined functions (UDFs)	4
UDF and UDT names	4
Function path	4
Overloaded functions	5
Triggers	5
Administrative support tables	5
Security and recovery	5
Installing the VideoCharger Extender	6
Prerequisites	6
Installing the extender on AIX	6
Installing the extender on Windows NT	7
Enabling and disabling data objects for the VideoCharger Extender	7
Enabling databases	8
Enabling tables and columns	8
Disabling a database	9
Enabling your VideoCharger server for the VideoCharger Extender	10
Managing audio and video objects with the VideoCharger Extender	10
Storing an object	11
Retrieving information about an object	11
Retrieving the size of an object	11
Retrieving the status of an object	12
Retrieving an object	12
Deleting an object	12
UDTs for the VideoCharger Extender	13
UDFs for the VideoCharger Extender	13
vcGetObjMetaData	14
Syntax	14
Parameters	14
Return value	14
Example	14
vcGetObjSize	14
Syntax	14
Parameters	14
Return value	15
Example	15
vcGetObjStatus	15
Syntax	15
Parameters	15
Return value	15
Example	15
vcInsertObjRef	15
Syntax	15
Parameters	15
Return value	16
Example	16
Synchronization utility	16
Syntax	17
Parameters	17
Messages issued by the VideoCharger Extender	18

Chapter 2. Notices	23
Trademarks.	24

Chapter 1. IBM VideoCharger Extender for DB2 Universal Database

The IBM VideoCharger Extender for DB2 Universal Database (DB2 UDB) enables you to manage your video and audio objects through your DB2 database while using your VideoCharger server for storage and retrieval of the objects. The extender provides you with the full power of DB2 for managing your data, and the advantages of scalability and data streaming provided by VideoCharger.

The VideoCharger Extender operates with DB2 for Windows NT and DB2 for AIX. For storing objects on the VideoCharger server and querying the database, your client can reside on any platform supported by DB2. For playing objects stored on the VideoCharger server, your client must be on a platform supported by the VideoCharger Player—either Windows 95, Windows 98, or Windows NT. Your DB2 server must be on AIX or Windows NT.

The information presented in this document assumes that you are familiar with IBM VideoCharger and IBM DB2 UDB.

For additional information about VideoCharger, see:

IBM VideoCharger Administrator's Guide and Reference, SC26-9499

IBM VideoCharger Programmer's Reference, available from the link to the VideoCharger documentation on your VideoCharger home page (created when you installed VideoCharger)

For additional information about DB2 UDB, see:

DB2 Universal Database Version 5 Quick Beginnings, S10J-8149 (Windows) or S10J-8148 (AIX)

DB2 Universal Database Version 5 Administration Guide, S10J-8157

DB2 Universal Database Version 5 Embedded SQL Programming Guide, S10J-8158

DB2 extender concepts

DB2 extenders extend the capability of DB2 by defining distinct data types and special functions for manipulating those types. By doing this, the extenders save you the time and effort of defining these data types and functions in your applications. The data types and functions are available through SQL, giving your applications a single point of access to the data supported by the extender as well as traditional numeric and character data.

For additional information about DB2 extenders, see:

DB2 Universal Database Image, Audio, and Video Extenders Administration and Programming, SC26-9107

DB2 Universal Database Text Extender Administration and Programming, SC26-9108

User-defined types (UDTs)

A *user-defined type* (UDT) provides a way to differentiate data used by the VideoCharger Extender. You can use UDTs in the same way as SQL built-in types to describe the data stored in columns of tables.

Extender concepts

The VideoCharger Extender provides the following UDTs, described in “UDTs for the VideoCharger Extender” on page 13:

vcobjfilename

The fully qualified name of a workstation file

vcobjmetadata

Metadata that describes an object stored on the VideoCharger server

vcobjref

Object reference information that identifies an object stored on the VideoCharger server

vcobjsize

The size of an object stored on the VideoCharger server

vcobjstatus

The status of an object stored on the VideoCharger server

User-defined functions (UDFs)

A *user-defined function* (UDF) is a way to create SQL functions, adding to the set of built-in functions supplied with DB2. You can use UDFs in an SQL statement in the same way that you use built-in functions.

The VideoCharger Extender provides the following UDFs, described in “UDFs for the VideoCharger Extender” on page 13:

vcGetObjMetaData

Returns the metadata for an object on the VideoCharger server

vcGetObjSize

Returns the size of an object stored on the VideoCharger server

vcGetObjStatus

Returns the status of an object stored on the VideoCharger server

vcInsertObjRef

Loads an object onto the VideoCharger server and stores information about it in the database

UDF and UDT names

The full name of a DB2 function is *schema-name.function-name*, where *schema-name* is an identifier that provides a logical grouping for SQL objects. The schema name for the VideoCharger Extender UDFs is `vcbadm`. The `vcbadm` schema name is also the qualifier for the VideoCharger Extender UDTs.

You can use the full name anywhere you refer to a UDF or a UDT. For example, `vcbadm.vcInsertObjRef` identifies the UDF whose schema name is `vcbadm` and whose function name is `vcInsertObjRef`. `vcbadm.vcobjref` identifies a UDT whose schema name is `vcbadm` and whose distinct-type name is `vcobjref`. You can also omit the schema name when you refer to a UDF or UDT; in this case, DB2 uses the function path to determine the function or distinct data type that you want.

Function path

The function path is an ordered list of schema names. DB2 uses the order of schema names in the list to resolve references to functions and distinct data types.

You can specify the function path by specifying the SQL statement SET CURRENT FUNCTION PATH. This sets the function path in the CURRENT FUNCTION PATH special register.

Recommendation: If you do not log on as vcdbadm, add the vcdbadm schema to the function path. This allows you to enter VideoCharger Extender UDF and UDT names without having to prefix them with vcdbadm. Use the following command to add the vcdbadm schema to the function path:

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, vcdbadm
```

Overloaded functions

Function names can be overloaded. *Overloaded* means that multiple UDFs, even in the same schema, can have the same name. However, two functions cannot have the same signature. A *signature* is the qualified function name concatenated with the defined data types of all the function parameters.

Triggers

A *trigger* defines a set of actions that are activated by a change to a table. The VideoCharger Extender uses a trigger to remove an object from the VideoCharger server when a row containing a reference to that object is deleted.

Administrative support tables

Administrative support tables contain the information that an extender needs to process user requests. Some administrative support tables identify tables that are enabled for an extender. Other administrative support tables contain attribute information about objects in enabled tables.

The VideoCharger Extender creates the following tables for its own use:

vc_server

Contains information on the VideoCharger servers where objects referenced by the tables in the database are stored

vc_object

Contains information on the objects stored on the VideoCharger server and information on the tables in the database that reference those objects

vc_objview

Contains all views of the vc_object table, however, users can access only those views for which they are authorized

Security and recovery

The information stored in the database tables enabled for the VideoCharger Extender and in the administrative support tables are afforded the same security and recovery protection as traditional numeric and character data. Users must have the required privilege to select, insert, or update objects.

The objects stored on the VideoCharger server are protected by VideoCharger.

The tables enabled for the VideoCharger Extender and the administrative support tables can be backed up and recovered in the same way as other data in DB2. The synchronization utility can be used to synchronize the objects stored on the

Extender concepts

VideoCharger server with the references to those objects by the tables in the database. See “Synchronization utility” on page 16.

Installing the VideoCharger Extender

Install the VideoCharger Extender on your DB2 server as described in the following sections.

Prerequisites

You must have the following software installed:

- On the workstation that is to be your DB2 server:
 - DB2 Universal Database Enterprise Edition version 5.0 or higher
 - DB2 Universal Database Workgroup Edition version 5.0 or higher
- On the workstation that is to be your VideoCharger server: IBM VideoCharger Version 2.0 with the PTF for level 2.0.0.2 applied

Important: For the VideoCharger server to access your databases, the Media Manager subsystem of VideoCharger must run in the DB2 instance environment. The DB2 instance is determined by the \$INSTHOME and \$DB2INSTANCE environment variables set by a db2profile script. When the fileset avs.mm 2.0.0.2 is applied to your VideoCharger server on AIX, a shell script called /usr/bin/mm.sh is created. This script invokes a db2profile script when the Media Manager is started to ensure that the Media Manager runs in the DB2 instance environment. If you executed a db2profile script prior to applying avs.mm 2.0.0.2, the script /usr/bin/mm.sh executes that db2profile script. If you have not executed a db2profile script prior to applying avs.mm 2.0.0.2, you must edit the script /usr/bin/mm.sh to invoke the proper db2profile script.

Installing the extender on AIX

You can get the VideoCharger Extender for AIX from the avs.db fileset on the external FTP site. This fileset includes:

- avs.db.db2, which contains the extender
- avs.db.sample, which contains sample code

To install the extender:

1. Execute the db2profile script provided by DB2, if it hasn't been executed previously. Enter:

```
. db2profile
```

This script defines the following environment variables used by the extender:

```
$INSTHOME  
$DB2INSTANCE
```

2. Enter smitty to install the VideoCharger Extender.

The extender is installed in the /usr/lpp/avs/database/db2 directory as db2vcxfn.o. If either the \$INSTHOME or \$DB2INSTANCE environment variable is defined, a link is created in the DB2 Instance home directory under the sqllib/function directories. If you have multiple DB2 instances that you want to enable for the VideoCharger Extender, you need to create the link manually for the additional instances.

To create a link manually, enter:

```
link -sf /usr/lpp/avs/database/db2/db2vcxfn.o $INSTHOME/sql1lib/function/db2vcxfn.o
```

See “Enabling tables and columns” on page 8 for additional information on creating a link.

3. Create a password for the administrator ID created during the installation of the extender:
 - a. Enter: `pwdadm vcdbadm` to create the password.
 - b. Enter: `pwdadm -c vcdbadm` to indicate that the password does not need to be changed at the next login.
4. If the `$DB2INSTANCE` variable is not set before you install the extender, you must manually add the administrator user ID, `vcdbadm`, to the DB2 instance group. Enter:

```
chgrpmem -m + vcdbadm db2iadm1
```

Installing the extender on Windows NT

Requirement: Use the user ID administrator to install the VideoCharger Extender. Other user IDs in the Administrators group do not have sufficient authority to ensure proper operation of the VideoCharger Extender.

To install the extender, run `avsexti` and respond to the prompts.

During installation, the administrator user ID, `vcdbadm`, is created for DB2 administration. The password for this user ID is `ibmvc`. This user ID is deleted when the VideoCharger Extender is uninstalled.

Enabling and disabling data objects for the VideoCharger Extender

You prepare databases, tables, and columns to hold extender data by enabling them. First enable the database. Then enable a table and any of its columns in the database.

When you no longer want extender data in your data objects, you can disable the objects.

The following sections describe how to:

- Enable your database
- Enable tables and columns in your database
- Disable your database

The scripts used to enable and disable data objects are located on your DB2 server in the following directories:

On AIX: `/usr/lpp/avs/database/db2`

On Windows NT: The `\bin` subdirectory of the directory where you installed the VideoCharger Extender, the default is `\ibm\videocharger\bin`

Tip: To allow users on client workstations to access the scripts, copy the scripts on a network drive.

Enabling and disabling data objects

Enabling databases

When you enable a database, the VideoCharger Extender:

- Creates the UDTs associated with the extender. The UDTs created are listed in “UDTs for the VideoCharger Extender” on page 13.
- Creates the UDFs associated with the extender. The UDFs created are listed in “UDFs for the VideoCharger Extender” on page 13.
- Creates the administrative support tables. These tables are used by the extender to manage extender data. Do not manually edit these tables.

Use the administration user ID `vcdbadm`, created when you installed the VideoCharger Extender, to enable your database for the VideoCharger Extender. To enable your database:

1. Create your database, if you don't already have one.

For example, the following command creates a database named `vcdb`. Enter the command from the command prompt:

```
db2 create database vcdb
```

2. Connect to your new database.

For example, the following command connects to the database named `vcdb`. Enter the command from the system prompt:

```
db2 connect to vcdb
```

3. Define the UDTs, UDFs, and administrative support tables using the `db2vcchr.dd1` script provided with the VideoCharger Extender for both AIX and Windows NT.

For example, the following command enables the database that you connected to in step 2. Enter the command from the command prompt:

```
db2 -tvf db2vcchr.dd1
```

Enabling tables and columns

Any user can create a table to contain VideoCharger objects. At least one column should be defined with the UDT `vcobjref`. The database must be enabled before you can enable a table within it.

To enable a table and its columns:

1. Create a table.

For example, the following command creates a table named `video_preview` with a column named `preview` that contains the reference information for VideoCharger objects. Enter the command from the command prompt:

```
db2 create table video_preview(  
    video_id char(5)      not null,  
    title   varchar(30)  not null,  
    preview vcobjref,  
    primary key(video_id)
```

2. Create the trigger for your table using the scripts provided with the VideoCharger Extender. The trigger is used when an object is deleted from the table.

On AIX, the script also creates the link needed by the VideoCharger Extender program `db2vcxfn.o` for the current DB2 instance. See “Installing the extender on AIX” on page 6 for additional information.

Enabling and disabling data objects

Tip: You must have the authority to delete rows from `vc_objview` to create a trigger for your table.

On AIX:

```
tblvccr.sh database-name userid table-name column-names
```

On Windows NT:

```
tblvccr table-name column-names
```

where:

database-name

The name of the database that contains the table that you want to enable for the extender.

On Windows NT, the database that you are connected to is used.

userid The user ID used to connect to the database.

On Windows NT, the user ID that you used to connect to *database-name* is used.

table-name

The name of the table that you want to enable for the extender.

column-names

The names of the columns in *table-name* that are defined with the type `vcobjref`. It is recommended that only one column per table be enabled.

For example, the following commands create the trigger for the column named `preview` in the table named `video_preview` in the database named `vcdb`. Enter the command from the command prompt:

On AIX: `tblvccr.sh VCDB VCDBADM VIDEO_PREVIEW PREVIEW`

On Windows NT: `tblvccr VIDEO_PREVIEW PREVIEW`

Disabling a database

To remove support for the VideoCharger Extender:

1. Delete all references to VideoCharger objects from the tables in the database. See “Deleting an object” on page 12.
2. Drop the trigger from each table enabled for the VideoCharger Extender using the scripts provided.

On AIX: `tblvcdl.sh database-name userid table-name`

On Windows NT: `tblvcdl table-name`

where:

database-name

The name of the database that contains the table that you want to enable for the extender.

On Windows NT, the database that you are connected to is used.

userid The user ID used to connect to the database.

On Windows NT, the user ID that you used to connect to *database-name* is used.

table-name

The name of the table that contains the trigger.

Enabling and disabling data objects

For example, the following command drops the trigger from the table named video_preview. Enter the command from the command prompt:

On AIX: `tblvcdcr.sh VCDB VCDBADM VIDEO_PREVIEW`

On Windows NT: `tblvcdcr VIDEO_PREVIEW`

3. Drop the administrative support tables using the db2vcdrp.dd1 script provided for both AIX and Windows NT.

For example:

```
db2 -tvf db2vcdrp.dd1
```

Important: You must disable support for tables enabled for the VideoCharger Extender before you drop a database that is enabled for the VideoCharger Extender.

Enabling your VideoCharger server for the VideoCharger Extender

For your VideoCharger server to access the database on your DB2 server, you must edit the db2cli.ini file on your VideoCharger server. The db2cli.ini file is located in the /home/\$DB2INSTANCE/sqlib/cfg directory on AIX or in the %DB2PATH% directory on Windows NT. Set the following parameters:

[data-source-name]

The name that denotes the section header in the db2cli.ini file. Typically, this is the database alias name. The name must be enclosed in square brackets.

DBALIAS

The name of your database

UID

The administrator user ID created for the VideoCharger Extender, vcdbadm

PWD

The password for vcdbadm, ibmvc

The following example shows an entry in the db2cli.ini file for the database named vcdb:

```
[VCDB]
DBALIAS=VCDB
UID=vcdbadm
PWD=ibmvc
```

Managing audio and video objects with the VideoCharger Extender

You can use the VideoCharger Extender to enable DB2 to manage the information about video and audio objects stored on your VideoCharger server. Descriptive information for an object (such as the title of a video, the director, the plot summary) can be stored in your table in columns defined with the standard numeric and character data types. The reference information used by the VideoCharger Extender is stored in the same table in the column defined with the UDT vcobjref. The object itself is stored on the VideoCharger server along with metadata needed by the VideoCharger server to access the object.

SQL statements containing references to VideoCharger Extender UDFs are used to manage the VideoCharger objects. These statements can be included in an application program or entered from the DB2 command line.

You can use standard SQL statements to query your database when searching for objects with similar attributes. For example, you can search for all video objects that

Managing audio and video objects

were directed by the same person. You can then use the SQL SELECT statement with the vcGetObjMetaData UDF to retrieve the information needed to play the video that you are interested in. Use the VideoCharger Player to view the video stored on the VideoCharger server.

The following sections describe how to use the UDFs to manage your audio and video objects.

A sample program, called vcsample.sqc, is provided with the VideoCharger Extender and shows how each of the UDFs can be used.

On AIX: The sample program and its makefile are in the /usr/samples/avs/database directory.

On Windows NT: The sample program and its makefile are in the \sample subdirectory of the directory where you installed the VideoCharger Extender, the default is \ibm\videocharger\sample.

The sample program uses a database named sample, which you must create before compiling the program. To use a different database, you must change the vcsample.sqc and vcsample.mak files to refer to your database. The database that you use for the sample must be enabled for the VideoCharger Extender. See “Enabling databases” on page 8.

Storing an object

Use the vcInsertObjRef UDF in an SQL INSERT statement to store a video or audio object. vcInsertObjRef performs the following actions:

- Stores the reference information and descriptive information in your database
- Updates the administrative support tables with the information about the VideoCharger server and the object being stored
- Transfers the object to your VideoCharger server using FTP

For example, the following statement stores information about a video in a table named video_preview and loads an object onto the VideoCharger server from a host named video1:

```
insert into video_preview values ('00001','Topgun',
    vcInsertObjRef('video1',          --ftp host name
                  'userid',          --ftp user ID
                  'password',        --ftp password
                  '9.111.22.333:23793', --VideoCharger server IP address
                  'AG',              --VideoCharger server asset group
                  'c:\topgun.mpg, c:\icing.mpg')); --file list
```

The VideoCharger server automatically creates and stores the metadata needed to access the object. The metadata is stored on the VideoCharger server.

Retrieving information about an object

You can use UDFs to retrieve the size or the status of an object. The size and status are stored on the VideoCharger server.

Retrieving the size of an object

Use the vcGetObjSize UDF in an SQL SELECT statement to retrieve the size of a video or audio object. The UDF retrieves the information from the VideoCharger server.

Managing audio and video objects

For example, the following statement retrieves the size of the object with the title *Topgun* in the table named `video_preview`:

```
select video_id,title,vcGetObjSize(preview)
  from video_preview
  where title='Topgun'
```

Retrieving the status of an object

Use the `vcGetObjStatus` UDF in an SQL `SELECT` statement to retrieve the status of a video or audio object. The status is either invalid, pending, or valid. The UDF retrieves the information from the VideoCharger server.

For example, the following statement retrieves the status of the object with the title *Topgun* in the table named `video_preview`:

```
select video_id,title,vcGetObjStatus(preview)
  from video_preview
  where title='Topgun'
```

Retrieving an object

Use the `vcGetObjMetaData` UDF in an SQL `SELECT` statement to retrieve the metadata for a video or audio object. The UDF retrieves the information from the VideoCharger server. You can store the metadata in a file for later use by specifying a file name when you call `vcGetObjMetaData`.

Use the metadata to invoke the VideoCharger Player to view the object. See the sample program, `vcsample.sqc`, for an example of invoking the VideoCharger Player.

The following statement retrieves the metadata for the object with the title *Topgun* in the table named `video_preview` and stores it in a file named `c:\vcobjmet.ivs`:

```
select video_id,title,vcGetObjMetaData(preview,'c:\vcobjmet.ivs')
  from video_preview
  where title='Topgun'
```

Deleting an object

Use an SQL `DELETE` statement to delete an object. Deleting the information from your table activates the trigger that removes the object from the VideoCharger server.

For example, the following command deletes a row from the table named `video_preview`:

```
delete from video_preview
  where video_id='00001'
```

Important: Before dropping a table that contains object references to objects on a VideoCharger server, you must delete all rows from the table. Deleting the row from the table triggers the VideoCharger server to delete the object from the VideoCharger server. If you drop a table without deleting the rows, the associated objects remain on the VideoCharger server and are inaccessible.

UDTs for the VideoCharger Extender

Table 1 describes the UDTs created by the VideoCharger Extender. The table also lists the DB2 source data type for each distinct data type.

Table 1. UDTs defined by the VideoCharger Extender

UDT	Source data type	Description
vcobjfilename	VARCHAR(128)	Fully qualified name of a workstation file
vcobjmetadata	VARCHAR(8196)	The metadata that describes an object The metadata is set by and stored on the VideoCharger server.
vcobjref	VARCHAR(128)	The object reference information stored in the database in the format: <i>server-ip-address:port-number/asset-group/object-name</i> where: <i>server-ip-address:port-number</i> The IP address and port number for the VideoCharger server If you don't provide a port number, the default port number for the VideoCharger Media Manager, 23793, is used. <i>asset-group</i> The VideoCharger asset group <i>object-name</i> The name of the object stored on the VideoCharger server
vcobjsize	DOUBLE	The size of the object in kilobytes (KBs) The size is stored on the VideoCharger server.
vcobjstatus	CHAR(1)	The status of the object The status can be: I Invalid. The store failed and the object cannot be used. P Pending. The object is being loaded onto the VideoCharger server. V Valid. The object is available for use. The status is stored in the vc_object table in your database.

UDFs for the VideoCharger Extender

This section provides reference information for the UDFs created by the VideoCharger Extender. The UDFs are listed in alphabetical order.

UDFs

vcGetObjMetaData

Returns the metadata for the object. The metadata is retrieved from the VideoCharger server. If a file name is specified, the metadata is stored in the file.

Syntax

Retrieve reference information

```
▶▶—vcGetObjMetaData—(—object-reference—)—————▶▶
```

Retrieve reference information to a file

```
▶▶—vcGetObjMetaData—(—object-reference—,—file-name—)—————▶▶
```

Parameters

object-reference

The reference information used to identify the object on the VideoCharger server. The data type for this parameter is vcobjref.

file-name

The name of the file where the returned metadata is stored. The data type for this parameter is vcobjfilename.

Return value

The metadata for the object. The data type for this value is vcobjmetadata.

The value returned contains the content-type information used to identify the object to a browser. When you pass this value to a browser, the browser launches the VideoCharger Player. The content-type information is not stored when a file name is provided.

Example

See the sample program, vcsample.sqc, provided with the VideoCharger Extender for an example of using this UDF.

vcGetObjSize

Returns the size of the object. The size is retrieved from the VideoCharger server.

Syntax

```
▶▶—vcGetObjSize—(—object-reference—)—————▶▶
```

Parameters

object-reference

The reference information used to identify the object on the VideoCharger server. The data type for this parameter is vcobjref.

Return value

The size of the object. The data type for this value is `vcobjsize`.

Example

See the sample program, `vcsample.sqc`, provided with the VideoCharger Extender for an example of using this UDF.

vcGetObjStatus

Returns the status of the object. The status is retrieved from the VideoCharger server.

Syntax

```
►►—vcGetObjStatus—(—object-reference—)—————►
```

Parameters

object-reference

The reference information used to identify the object on the VideoCharger server. The data type for this parameter is `vcobjref`.

Return value

The status of the object. The data type for this value is `vcobjstatus`.

Example

See the sample program, `vcsample.sqc`, provided with the VideoCharger Extender for an example of using this UDF.

vcInsertObjRef

Loads an object onto the VideoCharger server and stores the reference information in the database table.

Syntax

```
►►—vcInsertObjRef—(—ftp-host—,—ftp-userid—,—ftp-password—,—————►  
  
►—server-name—,—asset-group—,—file-list—)—————►
```

Parameters

ftp-host

Host name of the server where the object is located. The data type for this parameter is `VARCHAR(64)`.

ftp-userid

User ID used to access the object using FTP. The data type for this parameter is `VARCHAR(18)`.

UDFs

ftp-password

Password associated with *ftp-userid*. The data type for this parameter is VARCHAR(18).

server-name

The IP address and port number for the VideoCharger server. The data type for this parameter is VARCHAR(64).

If you don't provide a port number, the default port number for the VideoCharger Media Manager, 23793, is used.

asset-group

The VideoCharger server asset group. The data type for this parameter is VARCHAR(64).

If a null string is entered (""), the default asset group is used.

file-list

The list of files to be loaded. File names must be separated by a comma (,). If more than one file is listed, the files are concatenated in the specified order and a single file is stored on the VideoCharger server. The data type for this parameter is VARCHAR(255).

Return value

The reference information used to identify the object on the VideoCharger server. The data type for this value is vcobjref.

Example

The following statement stores information about a video in a table named `video_preview` and loads an object onto the VideoCharger server from a host named `video1`:

```
EXEC SQL insert into video_preview values ('00001','Topgun',
      vcInsertObjRef('video1',          --ftp host name
                    'userid',          --ftp user ID
                    'password',        --ftp password
                    '9.111.22.333:23793', --VideoCharger server IP address
                    'AG',              --VideoCharger server asset group
                    'c:\topgun.mpg, c:\icing.mpg')); --file list
```

Synchronization utility

Use the synchronization utility to synchronize the objects on your VideoCharger server with the tables that reference those objects. This utility uses the information in the administrative support tables to:

- Identify objects on the VideoCharger server that are no longer referenced by your tables
- Identify objects that were not successfully loaded onto the VideoCharger server
- Remove invalid objects from the VideoCharger server and delete the corresponding row from your table
- Remove pending objects from the VideoCharger server and delete the corresponding row from your table
- Remove objects from the VideoCharger server that are no longer referenced

Synchronization utility

Run the synchronization utility from a command prompt on your VideoCharger server. Objects found to be out of synchronization with the tables are displayed. Depending on the options specified, the objects are also removed.

You should run the utility for each table that is enabled for the VideoCharger Extender.

Restriction: For tables that have more than one column defined with the type `vcobjref`, the utility only displays objects with the status of pending or invalid. The flags are ignored and no objects are deleted. You must determine the appropriate action to take to correct the tables.

Syntax

On AIX:

```
▶▶ db2vcsyn.sh database userid password schema table
```

```
▶ [ -i ] [ -p ] [ -o ]
```

On Windows NT:

```
▶▶ db2vcsyn database userid password schema table
```

```
▶ [ -i ] [ -p ] [ -o ]
```

Parameters

database

The name of your database.

userid The administrator user ID used to connect to the database.

password

The password for *userid*.

schema

The schema of the database table used for synchronization. The value must be entered in uppercase.

table The name of the database table used for synchronization. The value must be entered in uppercase.

-i Removes invalid objects from the VideoCharger server and deletes the corresponding rows from the table

If this option is omitted, candidates for removal are displayed, but not removed.

-p Removes pending objects from the VideoCharger server and deletes the corresponding rows from the table

Synchronization utility

If this option is omitted, candidates for removal are displayed, but not removed.

Important: Objects that are in the pending state might be in the process of loading at the time this utility is run. Determine the state of the object on the VideoCharger server before specifying that these objects should be removed from the table.

- o Removes objects from the VideoCharger server that are not referenced

If this option is omitted, candidates for removal are displayed, but not removed.

Messages issued by the VideoCharger Extender

The VideoCharger Extender UDFs return a code in the SQLSTATE field of the SQLCA structure. Some messages are also logged to the log directory on the DB2 server:

On AIX: The log directory is in the /var/adm/ras directory

On Windows NT: The log directory is in the registry

Messages in the log can be mapped to a return code by replacing the prefix AVS with the number three (3). For example, message AVS8705E maps to return code 38705.

The following codes are returned by the UDFs:

38705

Problem: The VideoCharger media manager was unable to perform database operation on video-type tables.

User Action: Verify that the VideoCharger system is properly configured for database operation.

User Action: The VideoCharger media manager installation is incorrect, re-install the media manager. If the problem persists, contact your IBM Service representative for further assistance.

38706

Problem: An error condition was detected while performing the operation, the operation could not be completed.

User Action: Verify that the VideoCharger system is properly configured. See your IBM Service representative for further assistance.

38709

Problem: Insert operation failed, no file was specified to load.

User Action: Correct the command and retry the operation. If the problem persists, contact your IBM Service representative for further assistance.

38707

Problem: Received a corrupt order from a VideoCharger UDF.

User Action: Retry the operation. If the problem persists, contact your IBM Service representative for further assistance.

38727

Problem: The VideoCharger media manager was unable to perform database operation on video-type tables.

User Action: Verify that the VideoCharger system is properly configured for database operation.

38708

Problem: Internal error in VideoCharger media manager.

38728

Problem: The VideoCharger UDF is unable to communicate with the VideoCharger subsystem.

User Action: Verify that the VideoCharger system is properly configured and running.

38729

Problem: An error occurred while attempting to receive a response from the VideoCharger subsystem.

User Action: Verify that the VideoCharger system is properly configured and running.

38730

Problem: An insert operation failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38770

Problem: Delete object failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38771

Problem: Delete object failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38772

Problem: Get object metadata failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38773

Problem: Get object metadata failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38774

Problem: Get object metadata failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38775

Problem: Get object metadata failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38776

Problem: Get object reference failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38777

Problem: Get object size failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38778

Problem: Get object size failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38779

Problem: Get object status failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38780

Problem: Get object status failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38781

Problem: An invalid parameter was detected.

User Action: A null parameter was specified where non-null should be used.

38782

Problem: An insert operation failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38783

Problem: An insert operation failed.

User Action: Verify that the VideoCharger system is properly configured and running.

38805

Problem: Internal system error.

User Action: See message number 2100-010 in the *IBM VideoCharger Administrator's Guide and Reference*.

38806

Problem: Initialization is not done.

User Action: See message number 2100-011 in the *IBM VideoCharger Administrator's Guide and Reference*.

Messages

38808

Problem: Invalid number of files.

User Action: See message number 2100-230 in the *IBM VideoCharger Administrator's Guide and Reference*.

38809

Problem: Service not available.

User Action: See message number 2100-231 in the *IBM VideoCharger Administrator's Guide and Reference*.

38812

Problem: Resource is temporarily unavailable.

User Action: See message number 2100-012 in the *IBM VideoCharger Administrator's Guide and Reference*.

38813

Problem: Invalid event type.

User Action: See message number 2100-078 in the *IBM VideoCharger Administrator's Guide and Reference*.

38816

Problem: Bad flag.

User Action: See message number 2100-160 in the *IBM VideoCharger Administrator's Guide and Reference*.

38818

Problem: Bad argument.

User Action: See message number 2100-016 in the *IBM VideoCharger Administrator's Guide and Reference*.

38819

Problem: Invalid asset group.

User Action: See message number 2100-091 in the *IBM VideoCharger Administrator's Guide and Reference*.

38820

Problem: Invalid mode.

User Action: See message number 2100-003 in the *IBM VideoCharger Administrator's Guide and Reference*.

38821

Problem: Invalid SMPTE time.

User Action: See message number 2100-017 in the *IBM VideoCharger Administrator's Guide and Reference*.

38822

Problem: Session Limit exceeded.

User Action: See message number 2100-026 in the *IBM VideoCharger Administrator's Guide and Reference*.

38825

Problem: RPC system error.

User Action: See message number 2100-075 in the *IBM VideoCharger Administrator's Guide and Reference*.

38829

Problem: Invalid bit rate.

User Action: See message number 2100-092 in the *IBM VideoCharger Administrator's Guide and Reference*.

38830

Problem: Inconsistent resource.

User Action: See message number 2100-094 in the *IBM VideoCharger Administrator's Guide and Reference*.

38839

Problem: Asset bit rate is required.

User Action: See message number 2100-175 in the *IBM VideoCharger Administrator's Guide and Reference*.

38885

Problem: Bad name existed.

User Action: See message number 2100-163 in the *IBM VideoCharger Administrator's Guide and Reference*.

38898

Problem: Video file already exists.

User Action: See message number 2100-174 in the *IBM VideoCharger Administrator's Guide and Reference*.

38899

Problem: Name already exists.

User Action: See message number 2100-163 in the *IBM VideoCharger Administrator's Guide and Reference*.

38901

Problem: Name not found.

User Action: See message number 2100-164 in the *IBM VideoCharger Administrator's Guide and Reference*.

38902

Problem: Number of locations/files is < 1.

User Action: See message number 2100-177 in the *IBM VideoCharger Administrator's Guide and Reference*.

38903

Problem: More info/entries available.

User Action: See message number 2100-178 in the *IBM VideoCharger Administrator's Guide and Reference*.

38905

Problem: Insufficient resources.

User Action: See message number 2100-009 in the *IBM VideoCharger Administrator's Guide and Reference*.

38907

Problem: Insufficient disk space.

User Action: See message number 2100-096 in the *IBM VideoCharger Administrator's Guide and Reference*.

38910

Problem: Max users > 0 is required.

User Action: See message number 2100-176 in the *IBM VideoCharger Administrator's Guide and Reference*.

38913

Problem: Asset is already in use.

User Action: See message number 2100-165 in the *IBM VideoCharger Administrator's Guide and Reference*.

38915

Problem: Asset exists in asset group.

User Action: See message number 2100-193 in the *IBM VideoCharger Administrator's Guide and Reference*.

38916

Problem: Unable to open catalog.

User Action: See message number 2100-194 in the *IBM VideoCharger Administrator's Guide and Reference*.

38993

Problem: No asset replica available.

User Action: See message number 2100-159 in the *IBM VideoCharger Administrator's Guide and Reference*.

Messages

Chapter 2. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM	AIX
DB2	DB2 Universal Database
VideoCharger	

Microsoft, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks or service marks of others.