

Delta Snap

Version 4.2

Copyright © 2008, The Australian Software Company

This document refers to Delta Snap version 4.2 and higher

Introduction

Delta Snap can be run without the need for user intervention by passing a structured xml file in the command line. This file must contain a valid set of commands allowing Delta Snap to connect to a database server to perform known tasks.

The command line process does not output any prompts or notices to the user interface and all errors are sent to CommandLineErrors.txt. Feedback can be sent to a user through email which can be defined within the command script.

You can run a basic GUI version by not including a project name in the command line however this application is intended to be simple with a limited user interface.

Delta Snap was formally called SQL Schema.

Run Delta Snap with a Script file

Add the command line file name to the end of the executable command.

DeltaSnap script filename

For example

"C:\program files\Delta Snap\DeltaSnap.exe" base.sdp

Alternatively run DeltaSnap and then choose the project file from the application.

How Does Delta Snap Work

Delta Snap reads the script file and opens a database using the <Connection> details. Delta Snap then outputs a snapshot of the specified database schema. The snapshot is an XML file using SQL Delta format, not SQL Server's format.

The snapshot file can then be read by SQL Delta to compare the effectively offline database with a local, usually active database.

Typically Delta Snap is run in as batch file on a scheduled basis. For example you can run Delta Snap on SQL Servers that cannot be physically accessed, and have Delta Snap email the snapshot to you. This is useful for logging changes to a database.

Base Script Details

Each command line script is an xml style file using the definitions of W3C for XML. Each script must contain a single <PROJECT> element starting with <PROJECT> and ending with </PROJECT>. The xml version and encoding definition is not strictly required by the script processor however should be retained for consistency with the standard. The script processor can process Unicode or normal ASCII and does not refer to the encoding definition to determine the text style.

All commands are case insensitive and empty elements, that is elements with no text between the open and close are ignored. For example <CONNECT></CONNECT> is ignored.

The command elements are processed by Delta Snap in the order required by Delta Snap and not the order presented in the xml script file.

IMPORTANT

Delta Snap script file is not the same as the SQL Delta project file (script file). Only a single connection is required and an output file Schema File must be defined. You may find it convenient to copy elements of a SQL Delta project into your Delta Snap script.

Basic Delta Snap Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Project>
  <Connection>
    <ServerType>2</ServerType>
    <ServerName>MyServer</ServerName>
    <User>sa</User>
    <Password Clear="true">test</Password>
    <Database>AdventureWorks</Database>
    <SchemaFile>Adv-Snap.xml</SchemaFile>
  </Connection>
  <LogFile>Adv-Snap-Log.txt</LogFile>
  <Emails>
    <EItem>
      <Host>MailServer.yourworld.com</Host>
      <User>hello@yourworld.com</User>
      <Password>clearpass</Password>
      <To>snap@yourworld.com</To>
      <From>hello@yourworld.com</From>
      <Subject>Delta Snap File</Subject>
      <Body>Here it is...</Body>
      <Files><FItem><Name>Adv-Snap.xml</Name></FItem></Files>
      <NotifyAlways>True</NotifyAlways>
    </EItem>
  </Emails>
</Project>
```

The options have been ignored and the default is to load all objects. File name paths have not been included however they are recommended especially when using Windows Vista. Server Type is 2 means SQL Server 2005.

Command List

CONNECTION - mandatory

The following Optional Elements provide additional control of Delta Snap and the script process.

EMAILS
OPTIONS
VERSION
COMMENT
LOGFILE
SOUND
CFGRANDOMFILE
RANDOMFILE

CONNECTION: mandatory <CONNECTION>.. elements ..</CONNECTION>

Defines the connection details for the database.

```
<CONNECTION>
  <SERVERTYPE>1</SERVERTYPE>
  <SERVERNAME>TestServer</SERVERNAME>
  <PORT>0</PORT>
  <USER>sa</USER>
  <PASSWORD Clear="true">no</PASSWORD>
  <DATABASE>Test</DATABASE>
  <SCHEMAFILE>c:\test.xml</SCHEMAFILE>
</CONNECTION>
```

CONNECTION OPTIONS

SERVERTYPE: mandatory <SERVERTYPE>1 or 2</SERVERTYPE>

SERVERTYPE defines the type of server the database is hosted on. 1 is for SQL Server 2000 and 2 is for SQL Server 2005. Delta Snap will recalculate the server type if this value is incorrect however this process can take a few seconds so setting the correct value will save a little time.

SERVERNAME: mandatory <SERVERNAME>text</SERVERNAME>

SERVERNAME defines the database server. The text can be IP Address, URI or local registered server name.

PORT <PORT>0</PORT>

PORT has been included for future database server requirements when other platforms of the product are released. Setting this port does not set the SQL Server port. For a non standard SQL Server port use the syntax ServerName, Port in SERVERNAME.

USER Syntax: <USER>text</USER>

USER identifies the user name required for connection to the database server. Omit if connecting using Windows Authentication.

PASSWORD [Clear=true/false] <PASSWORD>text</PASSWORD>

PASSWORD identifies the password required for connection to the database server. Omit if connecting using Windows Authentication. By default or when Clear='False' the password should be encrypted and this can be done using the provided Password Encryption program. Otherwise set the attribute Clear='True' to force Delta Snap to treat the password as clear text.

DATABASE: mandatory <DATABASE>text</DATABASE>

DATABASE identifies the database within the database server.

SCHEMAFILE: mandatory <SCHEMAFILE>filename</SCHEMAFILE>

SCHEMAFILE specifies the path and filename for the database schema file. If the filename exists then it will be overwritten with the new database schema.

eg. <SCHEMAFILE>c:\test.xml</SCHEMAFILE>

See the RANDOMFILE option for information about creating file names that are unlikely to be overwritten and can be unique.

Optional Elements

These elements can be in any order and provide further functionality to Delta Snap.

EMAILS <EMAILS>.. elements ..</EMAILS>

Allows the user the ability to send one or more emails based on events that occur during Delta Snap operation. This feature is an ideal mechanism for providing user feedback since there is no visual output during operation of the command line process.

Within <EMAILS> there must be one or more EITEM elements with each EITEM holding specific email control detail. It is the responsibility of the user to correctly declare and confirm the SMTP details and Delta Snap processes the information as-is without verification or validation.

EITEM <EITEM>.. elements ..</EITEM>

Each email event must have an EITEM.

```
<Emails>
  <EItem>
    <Host>127.0.0.1</Host>
    <User>Admin</User>
    <Password>Pass</Password>
    <To>john@test</To>
    <From>me@test</From>
    <Subject>Delta Snap Info</Subject>
    <Body>See Attachment </Body>
    <NotifyAlways>True</NotifyAlways>
    <Files><FItem><Name>C:\result.xml</Name></FItem></Files>
  </EItem>
</Emails>
```

HOST <HOST>.. text ../HOST>

The name of the SMTP server to send the email. Either enter a URI or IP Address.

PORT <PORT>.. text ../PORT>

The port, by default 25, for access to the SMTP server. Leave the port blank if it's standard.

USER <USER>.. text ../USER>

If the SMTP server requires a login then USER is the user name. By default this is empty and is not required.

PASSWORD <PASSWORD>.. text ../PASSWORD>

If the SMTP server requires a login then PASSWORD is the password corresponding to the user name. By default this is empty and is not required.

TO <TO>.. text ../TO>

CC <CC>.. text ../CC>

BCC <BCC>.. text ../BCC>

A semi-comma separated list of email addresses to whom the email shall be sent, CC'd and BBC'd respectively.

FROM <FROM>.. text ../FROM>

The email address of the sender

SUBJECT <SUBJECT>.. text ../SUBJECT>

Subject text for the email. This can be empty and Delta Snap will add a subject.

BODY <BODY>.. text ../BODY>

Body of the email

NOTIFYALWAYS <NOTIFYALWAYS>.. true/false ../NOTIFYALWAYS>

Identifies when the email shall be sent, in this case if true an email will be sent upon completion of the comparison regardless of result. If an error occurs then the error log will be automatically added to the attachment list. The other notify options are not required if NotifyAlways is included.

NOTIFYSCHEMA <NOTIFYSCRIPT>.. true/false ../NOTIFYSCRIPT>

Identifies when the email shall be sent, in this case if true an email will be sent upon completion if the schema file is successfully created.

NOTIFYERRORS <NOTIFYERRORS>.. true/false ../NOTIFYERRORS>

Identifies when the email shall be sent, in this case if true an email will be sent only if an error occurs. The error log file will be automatically added to the attachment list if an error exists.

SHOWERRORS <SHOWERRORS>.. true/false ../SHOWERRORS>

Adds an error message to the email if the attachment files cannot be found. This is true by default.

FILES <FILES>.. elements ../FILES>

Indicates that file attachments are to be sent with the email. Sending attachments may be limited or prevented by the SMTP server or local anti-virus software. Within FILES there must be one or more FITEMS for each file.

FITEM <FITEM>.. elements ../FITEM>

Element enclosing the file name.

NAME <NAME>.. text ../NAME>

The filename including path of the attachment to send. The Log file should be added to this list if you wish to receive the log file.

OPTIONS <OPTIONS>.. elements ../OPTIONS>

Options element provides a method of configuring Delta Snap prior to the execution of the database load. Each option has the syntax:

<OPTIONELEMENT>true/false/text*</OPTIONELEMENT>

* Some options are integer or text values and are shown below

If an option is omitted then the existing default for the omitted option is used.

The options are:

LoadTables	(True) load or (False) don't load
LoadViews	“ “ “
LoadProcedures	“ “ “
LoadTriggers	“ “ “
LoadFunctions	“ “ “
LoadDefaults	“ “ “
LoadRules	“ “ “
LoadUDTs	“ “ “
LoadUsers	“ “ “
LoadRoles	“ “ “
LoadCatalogs	“ “ “
LoadDiagrams	“ “ “
LoadPermissions	“ “ “
LoadProperties	“ “ “
LoadSynonyms	“ “ “
LoadAssemblies	“ “ “
LoadXMLSchemaCollections	“ “ “
LoadMessageTypes	“ “ “
LoadContracts	“ “ “
LoadQueues	“ “ “

LoadServices	“ “ “
LoadRoutes	“ “ “
LoadRemoteServiceBindings	“ “ “
LoadSchemas	“ “ “
LoadPartitionFunctions	“ “ “
LoadPartitionSchemes	“ “ “
LoadCertificates	“ “ “
LoadAsymmetricKeys	“ “ “
LoadSymmetricKeys	“ “ “
LoadEventNotifications	“ “ “
LoadSystemDatabasesAndObjects	“ “ “
ConnectionTimeout	Numeric value indicating number of milliseconds to wait before issuing a timeout message. A value of 1000 equates to 1 second
CommandTimeout	Numeric value indicating number of milliseconds to wait before issuing a timeout message. A value of 1000 equates to 1 second

```

<Options>
  <LoadTables>true</LoadTables>
  <LoadViews>true</LoadViews>
  <LoadProcedures>true</LoadProcedures>
  <LoadTriggers>true</LoadTriggers>
  <LoadFunctions>true</LoadFunctions>
  <LoadDefaults>true</LoadDefaults>
  <LoadRules>true</LoadRules>
  <LoadUDTs>true</LoadUDTs>
  <LoadUsers>true</LoadUsers>
  <LoadRoles>true</LoadRoles>
  <LoadCatalogs>true</LoadCatalogs>
  <LoadDiagrams>true</LoadDiagrams>
  <LoadPermissions>true</LoadPermissions>
  <LoadProperties>true</LoadProperties>
  <LoadSynonyms>true</LoadSynonyms>
  <LoadAssemblies>true</LoadAssemblies>
  <LoadXMLSchemaCollections>true</LoadXMLSchemaCollections>
  <LoadSystemDatabasesAndObjects>>false</LoadSystemDatabasesAndObjects>
  <ConnectionTimeout>60</ConnectionTimeout>
  <CommandTimeout>600</CommandTimeout>
</Options>

```

VERSION <VERSION>.. text ..</VERSION>

The version element is reserved for future use and allows the script processor to determine the type and format of the script. By including a version within a script there is a greater likelihood that the script can be supported by future Delta Snap versions.

COMMENT <COMMENT>.. text ..</COMMENT>

Allows a description of the schema command line file to be stored in the script.

LOGFILE <LOGFILE>.. filename ../</LOGFILE>

LOGFILE specifies the path and filename for the Delta Snap log file. If the filename exists then it will be overwritten with the new log file. To receive this file by email you must add this file name to the files list in the email section of the script.

See the RANDOMFILE option for information about creating file names that are unlikely to be overwritten and can be unique.

SOUND <SOUND>.. File name/System name ../</SOUND >

The SOUND element plays either a system sound or wave file. The filename can be a predefined Windows sound such as SystemAsterisk, SystemExclamation, SystemHand, or a .wav filename. When the process is complete this sound will be played. If the sound cannot be located then no sound is made.

File Name Generation

File names can be standard Windows/DOS formatted names using a path and filename. Additionally Delta Snap allows you to create a unique or random filename to ensure each time Delta Snap runs you are saving the snapshot (or log file) into a new file.

Random file names require two elements, the configuration and then the usage element. CfgRandomFile defines how to create the filename and RandomFile use that configuration.

CFGRANDOMFILE Attributes: ID=integer (mandatory), DIR=string (optional)
<CFGRANDOMFILE ID="9" DIR="filepath">format</CFGRANDOMFILE>

CFGRANDOMFILE element provides a mechanism to configure the format of the RANDOMFILE element. Use the ID attribute as the link CfgRandomFile and RandomFile. The DIR element allows the user to configure a path and this path is checked before the file is created. The format text allows the user to format the filename using various date and time options to generate a reasonably random filename. There is no validation on the filename created and the user must ensure a valid format structure.

Format variables: c, d, dd, ddd, dddd, m, mm, mmm, mmmm, yy, yyyy, h, hh, n, nn, s, ss, z, zz, t, tt, a/p, am/pm, quoted text.

c : output filename using the current short date global format

d, dd, ddd, dddd : output filename using date day number (d, dd) or day name

m, mm, mmm, mmmm: output filename using date month number (m, mm) or month name.

yy or yyyy: output filename using date year in 2 or 4 digits.

t, tt : output filename using time in short or long format

h, hh : output filename using time hour.

n, nn : output filename using time minutes.

s, ss : output filename using time seconds.

z, zzz : output filename using time milli-seconds.

a/p, am/pm : output filename using am/pm symbols.

Quoted text (") : output literal text within the quotes (must conform to XML specifications).

Examples: mmmddhhnss"schema.xml" = apr26113001schema.xml
mmdyyzzz".xml" = 042605034.xml

RANDOMFILE <RANDOMFILE>..ID ..</RANDOMFILE>

RANDOMFILE element provides a way of creating a semi-random file name based on the current date and time using the formatting provided through CFGRANDOMFILE element. The ID indicates the filename buffer to store the name with possible values of 0 through to 9.

The first time the RANDOMFILE element for a specific ID is used a new file name will be created and stored in memory. If a CFGRANDOMFILE ID does not exist then a default file name of mmdyyhhnss"schema.xml" (see above definition) will be used.

Each subsequent time the RANDOMFILE element for a specific ID is used the stored filename will be re-used. This allows the user to create a date based filename preventing past scripts from being overwritten and also allowing the script to them email the file back to the user.

More Complex Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Project>
  <Connection>
    <ServerType>2</ServerType>
    <ServerName>MyServer</ServerName>
    <User>sa</User>
    <Password Clear="true">test</Password>
    <Database>AdventureWorks</Database>
    <SchemaFile><RandomFile>1</RandomFile></SchemaFile>
  </Connection>
  <LogFile><RandomFile>2</RandomFile></LogFile>
  <CfgRandomFile ID=1 DIR="C:\temp">
    &quot;Snap-&quot;mmdyyzzz&quot;.xml&quot;</CfgRandomFile>
  <CfgRandomFile ID=2 DIR="C:\temp">
    &quot;Log-&quot;mmdyyzzz&quot;.xml&quot;</CfgRandomFile>
  <Emails>
    <EItem>
      <Host>MailServer.yourworld.com</Host>
      <User>hello@yourworld.com</User>
      <Password>clearpass</Password>
      <To>snap@yourworld.com</To>
      <From>hello@yourworld.com</From>
      <Subject>Delta Snap File</Subject>
      <Body>Here it is...</Body>
      <Files><FItem><Name>
        <RandomFile>1</RandomFile>
      </Name></FItem></Files>
      <NotifyAlways>True</NotifyAlways>
    </EItem>
  </Emails>
  <Sound> SystemAsterisk</Sound>
  <Options>
    <LoadTables>true</LoadTables>
    <LoadViews>true</LoadViews>
    <LoadProcedures>true</LoadProcedures>
    <LoadTriggers>true</LoadTriggers>
    <LoadFunctions>true</LoadFunctions>
    <LoadDefaults>true</LoadDefaults>
    <LoadRules>true</LoadRules>
    <LoadUDTs>true</LoadUDTs>
    <LoadUsers>true</LoadUsers>
    <LoadRoles>true</LoadRoles>
    <LoadCatalogs>true</LoadCatalogs>
    <LoadDiagrams>true</LoadDiagrams>
```

```
<LoadPermissions>true</LoadPermissions>
<LoadProperties>true</LoadProperties>
<LoadSynonyms>true</LoadSynonyms>
<LoadAssemblies>true</LoadAssemblies>
<LoadXMLSchemaCollections>true</LoadXMLSchemaCollections>
<LoadMessageTypes>true</LoadMessageTypes>
<LoadContracts>true</LoadContracts>
<LoadQueues>true</LoadQueues>
<LoadServices>true</LoadServices>
<LoadRoutes>true</LoadRoutes>
<LoadRemoteServiceBindings>true</LoadRemoteServiceBindings>
<LoadSchemas>true</LoadSchemas>
<LoadPartitionFunctions>true</LoadPartitionFunctions>
<LoadPartitionSchemes>true</LoadPartitionSchemes>
<LoadCertificates>true</LoadCertificates>
<LoadAsymmetricKeys>true</LoadAsymmetricKeys>
<LoadSymmetricKeys>true</LoadSymmetricKeys>
<LoadEventNotifications>true</LoadEventNotifications>
<LoadSystemDatabasesAndObjects>false</LoadSystemDatabasesAndObjects>
  <ConnectionTimeout>60</ConnectionTimeout>
  <CommandTimeout>600</CommandTimeout>
</Options>
</Project>
```