

V12 Database Engine

Release Notes for version 3.1.1

February 2nd, 2000

What's New?

- ▶ The V12 Database Engine Xtra is now Verisign-approved. It can be automatically downloaded and installed on the client Mac or PC. See Auto-installation below.
- ▶ `V12Download()` enables the download of a V12 database from a remote URL. It can be used from both Projectors and Shockwave movies. It is secure: It downloads only V12 databases; it replaces a local file only if it is a valid V12 database and only if its password is supplied. See V12Download below.
- ▶ `V12DownloadInfo()` lets query the size of a remote V12 database without actually downloading it. See V12DownloadInfo below.
- ▶ V12 databases can now be open in Shared ReadOnly mode over a LAN. This is much faster than the Shared ReadWrite mode. Multiple accesses are granted to users only if they all open the shared database in Shared ReadWrite OR Shared ReadOnly mode (hybrid access is not permitted). See Shared ReadOnly and Shared ReadWrite: Rules and Exceptions below.

Fixed Issues

- ▶ `mGetUnique` did not return data from the master field if `mOrderBy` was used on a non-indexed field. Fixed.
- ▶ Characters were not properly mapped when importing DBF files on Mac. Fixed.
- ▶ Data formatting: spaces in a data format caused the retrieved data to be truncated. Fixed.
- ▶ `mOrderBy()` on a non-indexed field followed by `mSelectAll()` returned 0 records in the selection. Fixed.
- ▶ Styled text was not properly saved in Shared ReadWrite mode. Fixed.
- ▶ When ASCII #13 (carriage return) characters were used as record delimiters and when Text Qualifiers were present in the imported text, carriage returns that were part of the actual field data were ignored. Fixed.
- ▶ Opening a file in Shared ReadWrite mode from a locked volume caused errors on the Mac. It now opens the file in Shared ReadOnly mode.

Shared ReadOnly and Shared ReadWrite: Rules and Exceptions

V12 Database Engine relies on Windows' and MacOS' file and network managers to access shared files on a LAN. Thus, the following exceptions apply:

- ▶ If a V12 database is open in Shared ReadWrite mode by one or more V12-DBE clients, new clients can open it only in Shared ReadWrite mode. Shared ReadOnly opening is refused until the last client has dismissed its V12-DBE Xtra instances. If a V12 database is open by one or more V12-DBE clients in Shared ReadWrite mode, any Mac client will also be able to open it in Shared ReadOnly mode.
- ▶ If a V12 database is open in Shared ReadOnly mode by one or more V12-DBE clients, both Shared ReadOnly and Shared ReadWrite access are permitted. However, if a client opens the database in Shared ReadWrite mode, all Shared ReadOnly operations are suspended until the Shared ReadWrite instances are dismissed.

As before, only Microsoft's and Apple's networking protocols are supported. Third party networking software such as Dave, CopsTalks, etc. are not supported.

Auto-installation

Whether you deliver your project as a Shockwave movie or a Projector, V12 Database Engine version 3.1.1 Xtra can be automatically installed to your end-users computers via the web. This feature is supported as of Director 7 only. To instruct your movie to auto-install the V12 Database Engine Xtra when needed:

- 1 Make sure (or add) the following lines are in the "xtrainfo.txt" file located in your Director folder (watch out: the text between brackets must not be interrupted by Carriage Returns):

```
; V12-DBE for Director by Integration New Media, Inc.  
; http://www.integration.qc.ca/  
[#namePPC:"V12-DBE for Director.Xtr", #nameW32:"V12-DBE for Director.X32",  
#package:"http://www.integration.qc.ca/products/v12director/verisign/v12",  
#info:"http://www.integration.qc.ca/products/v12director"]
```
- 2 Launch Director and open your movie.
- 3 Choose Modify > Movie > Xtras.
- 4 If V12-DBE for Director is not in the list, click Add. Choose V12-DBE for Director.X32 (Win) or V12-DBE for Director.XTR (Mac) and click OK.
- 5 Click V12-DBE for Director.X32 (Win) or V12-DBE for Director.XTR (Mac) in the Xtras list.
- 6 Click Download if Needed.

At this point, Director tries to validate V12 Database Engine Xtra's URL and reports errors, if any.

Users who access Shockwave movies that automatically download V12-DBE get a Verisign certificate with the Xtra's credentials. Verisign is a standard means of downloading software from secure sources. Users have the choice of allowing or refusing the installation.

V12Download

Syntax `V12Download(url, local_file, password, completion_handler, status_handler, ref)`

Parameters `url`: URL of file to download. E.g.: `http://www.integration.qc.ca/test.v12`.

`local_file`: full pathname of the destination file. If the movie is playing in Director's authoring mode, as a projector, or as a local shockwave movie (specified by a local file name), the V12 database is downloaded to the specified location.

If the movie is running as a remote shockwave movie (a shockwave movie specified by its URL), only the file name is used. The full path, if specified, is ignored and the local file is created in the Temp folder. In that case, the downloaded database is erased when the movie is closed.

If `local_file` exists and it is not a V12 database or if its password is not supplied, `V12Download` fails and signals an error code.

`local_file` must be closed when `V12Download` is invoked.

`password`: password of the database to download. If `local_file` exists, `password` must also be able to unlock it.

`completion_handler`: Lingo handler to call when download is complete. If `ref` is zero, `completion_handler` accepts a single parameter: `V12Download`'s error code. If `ref` is an actual reference to a parent script or behavior, that reference is also returned to

```

completion_handler. E.g.
    on whenDownloadComplete err    -- global handler
    ... (err = error code)
    end whenDownloadComplete
OR
    on whenDownloadComplete me, err    -- from parent script or behavior
    ... (err = error code)
    ... (me = ref to parent script or behavior)
    end whenDownloadComplete

status_handler: Lingo handler to call repeatedly as the download progresses. If ref is zero,
status_handler accepts a single parameter: the number of bytes downloaded so far. If ref is
an actual reference to a parent script or behavior, that reference is also returned to
status_handler. E.g.
    on asDownloadProgresses b    -- global handler
    ... (b = number of bytes downloaded thus far)
    end asDownloadProgresses
OR
    on asDownloadProgresses me, b -- from parent script or behavior
    ... (b = number of bytes downloaded thus far)
    ... (me = ref to parent script or behavior)
    end asDownloadProgresses

ref: reference of parent script or behavior, or zero if none is used.

```

Description Download the remote V12 database specified by its URL to the local hard drive. If a local V12 database of the same name already exists, it is replaced by the downloaded file only if it is a valid V12 database and its password is supplied.

V12download resumes and returns control to Director immediately after initiating the download query. It calls *status_handler* as frequently as possible during download (generally used to display the download status in the user interface) and ends-up calling *completion_handler* when the download is complete (generally used to open the database and start working with it).

Example

```

on doDownloadDB
    V12Download("http://www.tagada.com/myDB.V12", the moviePath&"myDB.V12",
    "passw", "whenDownloadComplete", "asDownloadProgresses", 0)
    -- will periodically call asDownloadProgresses
    -- will resume execution in whenDownloadComplete
    V12CheckError() -- if error occurred (wrong password, non-V12 local file...),
    download is aborted
end doDownloadDB

on asDownloadProgresses b
    global gDBsize -- already initialized with actual size of file to download
    if (gDBsize>0) then put (100*b/gDBsize) & "% complete" into field "Feedback"
end asDownloadProgresses

on whenDownloadComplete err
    if (err) then
        alert "Download error " & V12Error(err)
        exit
    end if
    -- start normal use of local V12 database
    set gDB = new(Xtra "V12dbe", the moviePath&"myDB.V12", "ReadOnly", "passw")
    V12CheckError()
    -- etc.
end whenDownloadComplete

```

V12DownloadInfo

Syntax `V12DownloadInfo(url, info_type, completion_handler, ref)`

Parameters `url`: URL of file to get information on. E.g.: `http://www.integration.qc.ca/test.v12`.

`info_type`: type of information required. Currently, only file size is supported. This parameter is therefore necessarily `#size`.

`completion_handler`: Lingo handler to call when the required information is received on the client side. If `ref` is zero, `completion_handler` accepts two parameters: V12DownloadInfo's error code and the required data (in this case, file size). If `ref` is an actual reference to a parent script or behavior, that reference is also returned to `completion_handler`. E.g.

```
on whenInfoComplete err, s    -- global handler
... (err = error code)
... (s = the required data)
end whenInfoComplete
```

OR

```
on whenInfoComplete me, err, s    -- from parent script or behavior
... (err = error code)
... (s = the required data)
... (me = ref to parent script or behavior)
end whenInfoComplete
```

`ref`: reference of parent script or behavior, or zero if none is used.

Description Get information on `url` before actually downloading it. At the present time, only `url`'s file size can be returned, although other attributes (such as creation date, modification date, etc.) may be supported in a future release. This feature largely depends upon the HTTP server's ability to provide the required information.

V12downloadInfo resumes and returns control to Director immediately after initiating the network query. It eventually calls `completion_handler` when the required information is received.

Example

```
on GetRemoteDBsize
    V12DownloadInfo("http://www.tagada.com/myDB.V12", #size, "whenInfoGotten", 0)
    -- will resume execution in whenInfoReceived
end GetRemoteDBsize

on whenInfoGotten err, s -- called back from getRemoteDBInfo
    global gDBsize
    if (err) then
        alert "Network error " & V12Error(err) & ". Please make sure that " & ~
            "your network connection is up and that the requested URL exists."
        exit
    end if
    gDBsize = s
    -- can download file now with V12Dwnload
end whenInfoGotten
```