

SQL Connectivity Kits Version 6.2.0.42

The family of SQL Connectivity Kits (including Microsoft SQL Server, IBM DB2 and ODBC) are published as a set. The changes and fixes listed in this document comprise all changes made to that set. Some changes are specific to particular CKs and others are generic to the set.

Recent Changes and Fixes (since Build 4.0.0.35)

Build 6.2.0.42

All DataFlex SQL Drivers:

Removed DF_File_Record_Identity attribute from Studio driver configuration files.

Fix to make Studio driver configuration files work with DataFlex Studio before 19.0.

Changed files:

MSSQLDRV_DriverDef.xml

DB2_DRV_DriverDef.xml

ODBC_DRV_Driverdef.xml.

Build 6.2.0.41

All DataFlex SQL Drivers:

Delete_Index of a non-temporary index outside structure_start/structure_end caused access violation. (BT 7318)

All DataFlex SQL Drivers:

Recnum tables with RECNUM column bigint could not be restructured. Fixed.

All DataFlex SQL Drivers:

System table with real recnum column on SQL and no primary_index 0 in the INT file, would not hide the recnum column.

MSSQLDRV:

Changed MSSQLDRV_DriverDef.xml. MSSQLDRV_DriverDef.xml would only work with DataFlex Studio 19.0, but not with DataFlex Studio's before 19.0.

Build 6.2.0.40

MSSQLDRV: A date or datetime column with a default value 'getdate()', would not be filled with the current datetime on save of a new record. MSSQLDRV will now use the OUTPUT clause to retrieve the current datetime after save of a new record.

DB2_DRV and ODBC_DRV:

A date or datetime column with a default value 'CURRENT_TIMESTAMP', would not be filled with the current datetime on save of a new record. The default value to generate a timestamp can vary per database backend.

DB2_DRV will recognize 'CURRENT_TIMESTAMP' and 'CURRENT TIMESTAMP'

ODBC_DRV will recognize:

CURRENT_TIMESTAMP	(ANSI standard, Oracle)
CURRENT_DATE	(PostgreSQL)
NOW()	(PostgreSQL, MySQL)
DATE()	(MySQL)

Build 6.2.0.39

All DataFlex SQL Drivers: Set_attribute DF_INDEX_NAME for a temporary index caused exception.

Changed cli.pkg DF_INDEX_TEMPORARY had wrong value.

Build 6.2.0.38

MSSQLDRV: In some very rare situations a system table can unintentionally end up with more than 1 record. On SQL Server this would generate very hard to track-down errors: "Connection is busy with results for another command".

Made a change that prevents the connection busy error. Instead, a "System Table has more than one record." error will be generated on open.

Build 6.2.0.37

All DataFlex SQL Drivers: Changed naming in driver dll properties:

mssqldrv.dll	DataFlex SQL Server Driver
db2_drv.dll	DataFlex DB2 Driver
odbc_drv.dll	DataFlex ODBC driver

Build 6.2.0.36

MSSQLDRV: When using GUID (uniqueidentifier) columns with default value newid() or newsequentialid(), the OUTPUT clause is used to return generated GUID values in the buffer. When the table also has a trigger defined, this caused an error about invalid use of the OUTPUT clause. Changed to use OUTPUT INTO clause.

Build 6.2.0.35

All CLI Connectivity Kits: Updated Studio driver configuration files:

ODBC_DRV_DriverDef.xml.
df_index_sql_primary_key and df_index_clustered were missing.
Changed all attribute names to lowercase for better display in properties panel.
Changed description of df_field_default_value (square bracket use)

MSSQLDRV_DriverDef.xml
Changed description of df_field_default_value (square bracket use)

DB2_DRV_DriverDef.xml
Added df_file_identity_cache_size attribute
Changed description of df_field_default_value (square bracket use)

DB2_DRV: Do not write out IDENTITY_CACHE_SIZE to INT file, if the value is 0 or 1 (NO_CACHE).

Build 6.2.0.34

All CLI Connectivity Kits: Changed index creation order during restructure/conversion: Primary Key indexes will always be created first. This fixes a problem in DB2 where in some specific situation (Non-clustered index, PK name not alphabetically first), an incorrect index would be used for the Primary Key.

DB2_DRV: When (re-)creating a primary key during restructure, the callback (output panel in Studio) would sometimes incorrectly report "Failed to create primary key."

All CLI Connectivity Kits Improved context information in Open errors. Added Table= and Source= context.

Build 6.2.0.33

DB2_DRV: Calling a DB2 stored procedure with a CLOB argument. When the length of the passed parameter in SQLSetArgument was larger than 16K, the parameter would be truncated to 16K. Now fixed to pass any length, provided the argument_size is large enough.

Build 6.2.0.32

All CLI Connectivity Kits: Changed some error texts for open errors.

Build 6.2.0.31

All CLI Connectivity Kits: Open of a system table in database explorer 19.0 raised "Index number out of range" error. Caused by Get_attribute DF_INDEX_NAME of a system table raising "Index number out of range" error. Changed to not generate error but return empty string for DF_INDEX_NAME.

Build 6.2.0.30

All CLI Connectivity Kits: When setting or getting a SQL filter no oem/ansi conversion was done. This caused SQL filters not to work properly with accented characters. Changed set_attribute/get_attribute DF_FILE_SQL_FILTER to perform oem/ansi conversion when table_character_format is ANSI.

MSSQLDRV: Added SQL Server client name in error text on minimum client version.

Build 6.2.0.29

All CLI Connectivity Kits: Changed cli.pkg. Added constants for missing cli error codes (CLIERR_)

All CLI Connectivity Kits: Added version check for minimum database client version for DB2_DRV and ODBC_DRV. (This was already added for MSSQLDRV in build 6.2.0.28)

New driver configuration file (mssqldrv.int, db2_drv.int, odbc_drv.int) keyword: MINIMUM_CLIENT_VERSION
New attribute: DF_DRIVER_MINIMUM_CLIENT_VERSION

For DB2_DRV and ODBC_DRV the default value for minimum client version is 0. For MSSQLDRV see notes build 6.2.0.28

DB2_DRV: Made a change to enable DSN-less connections for DB2.

All CLI Connectivity Kits: Changed driver configuration files (mssqldrv.int, db2_drv.int, odbc_drv.int) Added new settings for Connectivity Kits 6.2: Minimum_Client_Version, Login_On_Open and Allowed_structure_changes

Build 6.2.0.28

All CLI Connectivity Kits: All searches for connectstring keywords (like SERVER, DSN) made case insensitive.

MSSQLDRV: Added version check for minimum database client version.
(Currently only for mssqldrv)

New driver configuration file (mssqldrv.int) keyword:
MINIMUM_CLIENT_VERSION
New attribute:
DF_DRIVER_MINIMUM_CLIENT_VERSION

For SQL Server the minimum client version can be set to:

- | | |
|----|--|
| 8 | SQL Server (SQL Server 2000 client) |
| 9 | SQL Native Client (SQL Server 2005 client) |
| 10 | SQL Server Native Client 10.0 (SQL Server 2008 client) |
| 11 | SQL Server Native Client 11.0 (SQL Server 2012 client) |
| 12 | ODBC Driver 11 for SQL Server (SQL Server 2014 client) |
| 13 | ODBC Driver 13 for SQL Server (SQL Server 2016 client) |

The default value for minimum client version is 10 SQL Server Native Client 10.0 (SQL Server 2008 client)
The version check will be performed during login. Note the first open in a program may also do a login.
Notes:

In general it is best to use matching client and server versions. For this the match_client_server setting can be used.

Using older client with newer server versions may sometimes cause unpredictable results.
A known issue is when using SQL Server date type. This type was introduced in SQL Server 2008 and is not recognized by older clients. An older client will return SQL dates as string (df_ascii), which can cause incorrect date values in the database.

Changed cli.pkg
Added DF_DRIVER_MINIMUM_CLIENT_VERSION attribute

Changed error file dferr003.dat. Added:
12344 Structure change not allowed
12345 Minimum database client not found

MSSQLDRV:

Changed mssqldrv.pkg
Added:
Function SqlServerClientVersionName Integer iClientVersion Returns String

ODBC_DRV:

Changed ODBC_DRV_DriverDef.xml
New attributes for DataFlex 19.0 / Connectivity Kit 6.2 added to odbc_drv_driverdef.xml:
df_file_restructure
df_file_restructure_int_only
df_file_allowed_structure_changes
df_index_sql_type
df_index_number

These attributes were already added in mssqldrv_driverdef.xml and db2_drv_driverdef.xml.

Build 6.2.0.27

Internal build

Build 6.2.0.26

DB2_DRV:

ODBC_DRV:

Enumeration of tables with function EnumerateTables would not list all tables when one or more tables with comments longer than 128 characters were found.

Build 6.2.0.25

All CLI Connectivity Kits:

Reference to index.0 was not written to the INT file in case of an index-only restructure. (A restructure that only involves creating/dropping of indexes).

All CLI Connectivity Kits:

If no SERVER_NAME was present in the INT file, would cause an exception. The Connectivity Kit will now raise an appropriate error.

Build 6.2.0.24

All CLI Connectivity Kits:

cli.pkg Removed double ifdef

All CLI Connectivity Kits:

After a failed login with invalid connection id, a connection handle was not released.

Build 6.2.0.22

MSSQLDRV:

Changed mssqldrv.pkg:

Added constants for "ODBC Driver 11 for SQL Server" (Client for SQL Server 2014) and "ODBC Driver 13 for SQL Server". (Client for SQL Server 2016) in mssqldrv.pkg.

Added new function EnumerateServersLocal. This function returns all SQL server instances on the local machine.

Existing function EnumerateServers has not changed and will still return all SQL Server instances in the network. Note this function sometimes takes a long time.

Build 6.2.0.20

MSSQLDRV:

Added support for "ODBC Driver 11 for SQL Server" (Client for SQL Server 2014) and "ODBC Driver 13 for SQL Server". (Client for SQL Server 2016).

The MSSQLDRV connectivity Kit will now recognize the following clients. The highest installed client version on a workstation will be used:

```
#define SQLSERVER2000CLIENT 8 /* "SQL Server" */
#define SQLSERVER2005CLIENT 9 /* "SQL Native Client" */
#define SQLSERVER2008CLIENT 10 /* "SQL Server Native Client 10.0" */
#define SQLSERVER2012CLIENT 11 /* "SQL Server Native Client 11.0" */
#define SQLSERVER2014CLIENT 12 /* "ODBC Driver 11 for SQL Server" */
#define SQLSERVER2016CLIENT 13 /* "ODBC Driver 13 for SQL Server" */
```

All CLI Connectivity Kits:

Bug in RedirectIdConnectString caused memory error in logout.

All CLI Connectivity Kits:

Setting a constrain with accented characters did no oem/ansi translation.
(Merge from build 6.1.0.32)

MSSQLDRV:

Changed default map schema for SQL Server (DEFAULT_MAP_DF_TO_SQL_TYPE_SCHEMA) to MAP_DF_TO_SQL_TYPE_SQL2012.

The map schema controls which SQL type will be used when creating new columns. The default map schema was MAP_DF_TO_SQL_TYPE_SQL2000 but is now changed to MAP_DF_TO_SQL_TYPE_SQL2012.

The following table shows the available mapping schema's:

DataFlex Type	SQL2000	SQL2005	SQL2008	SQL2012
DF_ASCII	Char	Char	Char	Char
DF_DATE	Datetime	Datetime	Date	Date
DF_DATETIME	Datetime	Datetime	Datetime2	Datetime2
DF_TEXT	Text	Varchar(max)	Varchar(max)	Varchar(max)
DF_BINARY	Binary/image	Varbinary(max)	Varbinary(max)	Varbinary(max)

Notes:

These type mappings only apply when adding/creating new columns. Existing columns will always keep the type they already have.

These mappings will apply when converting from embedded database to SQL Server, since this is effectively creating new columns.

Previous versions (before build 6.2.0.20) used SQL200 mappings by default. Build 6.2.0.20 and later will use SQL2012 mappings by default.

The SQL Server 2012 map schema can also be used for SQL Server 2014 and SQL Server 2016.

Changed mssqldrv.int to reflect the new defaults.

Build 6.2.0.19

All CLI Connectivity Kits:

A masked time value of 00:00:00 would be passed as " : : ", causing a "Time contains an invalid value" error. This will now be handled as time value 00:00:00.

(Merge from build 6.1.0.31)

Build 6.2.0.19

All CLI Connectivity Kits:

Restructure prevention changes.

Implemented a new table level attribute DF_FILE_ALLOWED_STRUCTURE_CHANGES

This can be set to following values:

ALL_TABLE_CHANGES_ALLOWED
ONLY_TABLE_INT_FILE_CHANGES_ALLOWED
NO_TABLE_CHANGES_ALLOWED

All set_attribute of DF_FILE_*, DF_FIELD_* and DF_INDEX_* attributes will check against the DF_FILE_ALLOWED_STRUCTURE_CHANGES and raise an error if a change is not allowed.

Changing properties of a table may result in changes in the INT file only (client side changes) or changes on the server (server side changes). In some situations it may be desirable to allow only client side changes. In that case this attribute can be set to ONLY_TABLE_INT_FILE_CHANGES_ALLOWED to protect against server side changes.

Example:

```
Set_Attribute DF_FILE_ALLOWED_STRUCTURE_CHANGES of hTable iField to  
ONLY_TABLE_INT_FILE_CHANGES_ALLOWED  
Structure_Start hTable "MSSQLDRV"  
Set_Attribute DF_FIELD_LENGTH of hTable iField to iNewLength  
Structure_End hTable DF_STRUCTEND_OPT_NONE
```

Build 6.2.0.18

All CLI Connectivity Kits:

Added new DF_DRIVER_CONFORMANCE level bit:
Bit 4 Driver supports handled connections (build 6.2.0.18)

DF_DRIVER_CONFORMANCE now indicates the following features

- bit 1 Driver supports rowid
- bit 2 Driver supports datetime type
- bit 3 Driver supports DD SQL filters (build 6.0.0.22)
- bit 4 Driver supports handled connections (build 6.2.0.18)

All CLI Connectivity Kits:

Restructure prevention changes.

Implemented a new setting ALLOWED_STRUCTURE_CHANGES

This can be set to following values:

- ALL_TABLE_CHANGES_ALLOWED
- ONLY_TABLE_INT_FILE_CHANGES_ALLOWED
- NO_TABLE_CHANGES_ALLOWED

ALLOWED_STRUCTURE_CHANGES can be configured in driver configuration file (mssqldrv.int, db2_drv.int, odbc_drv.int)

All set_attribute of DF_FILE_*, DF_FIELD_* and DF_INDEX_* attributes will check against the ALLOWED_STRUCTURE_CHANGES and raise an error if a change is not allowed.

Build 6.2.0.17

All CLI Connectivity Kits:

If an error occurs during open, the driver will now return error 75 DFERR_CANT_OPEN_DATA_FILE.

This will make any error during open a critical error that will abort the program.

The original error with the reason for the error will be returned in the additional error text.

Build 6.2.0.16

All CLI Connectivity Kits:

Bugfix: Setting the value of a SQL_TIME column, could cause memory overwrite.

This was side effect of earlier fix in 6.2.0.13.

RedirectIdConnectionString and RedirectIdConnectionStringKeepLoggedIn now combined to single function with bKeepLoggedIn parameter.

New RedirectIdConnectionString function.

This function can be used to redirect connections when using connection ids.

This function should be used instead of RedirectConnection when using connection ids

The function is defined in cCliHandler class in cli.pkg

This function can be used for switching between identical databases in multi-tenant situations.

Syntax:

Function RedirectIdConnectionString String sConnectionId String sConnectionString Boolean bKeepLoggedIn
Returns Integer

Description:

Redirect a connection of a connection ID

sConnectionId must contain a valid ConnectionId ("DFCONNID=MyID")

sConnectionString must contain a valid connectstring

("SERVER=MyServer;DATABASE=MyDatabase;UID=MyName;PWD=MyPassword")

RedirectIdConnectionString will redirect from the old connection (Server/Database the connectionId currently points to) to the new connection (Server/Database specified in sConnectionString).

All opened tables will be moved from old connection to new connection. (Tables must have identical structure in both connections)

If bKeepLoggedIn = false, the new connection will be logged in, the old connection will be logged out on each redirect.

If bKeepLoggedIn = true, connections will remain open and re-used on next redirectIdConnection

Returns:

- 0 when the redirection was successful.
- 1 if the connection ID does not exist or can not be found.
- 2 login into the new connection failed.

Example:

```
// We use the following connectionId:  
// MssqlOrderID=SERVER=(local);DATABASE=MssqlOrder190;Trusted_Connection=yes
```

```
String sConnectionId  
String sConnectionString
```

Open SalesP

```
Move "DFCONNID=MssqlOrderID" to sConnectionId  
Move "SERVER=(local);DATABASE=MssqlOrder182;Trusted_connection=yes" to sConnectionString
```

```
Set psDriverID of oCliHandler to "MSSQLDRV"  
Move True to bKeepLoggedIn  
Get RedirectIdConnectionString of oCliHandler sConnectionId sConnectionString bKeepLoggedIn to iRetVal
```

In this example before the RedirectIdConnectionString:

```
ConnectionId "DFCONNID=MssqlOrderID" will point to  
SERVER=(local);DATABASE=MssqlOrder190;Trusted_Connection=yes
```

Table SalesP will be open in SERVER=(local);DATABASE=MssqlOrder190

After RedirectIdConnectionString:

```
ConnectionId "DFCONNID=MssqlOrderID" will point to  
SERVER=(local);DATABASE=MssqlOrder182;Trusted_Connection=yes
```

Table SalesP will be open in SERVER=(local);DATABASE=MssqlOrder182

Notes on RedirectIdConnectionString

- Can switch between different databases on different servers
- Tables must have identical structures in both databases. Differences in structures will give unpredictable results
- If bKeepLoggedIn = false: Switching the database will involve login to the new connection and logout from the old connection on every redirect.
- If bKeepLoggedIn = true, connections will remain open and re-used on next redirectIdConnection

Build 6.2.0.15

New RedirectIdConnectionStringKeepLoggedIn function.

Amended version of RedirectIdConnectionString function. Syntax and functionality is same.

Difference:

- RedirectIdConnectionString will logout of old connection and login to old connection on every redirect.
- RedirectIdConnectionStringKeepLoggedIn will not logout/login on redirect but keep connections open.

Build 6.2.0.14

Added new attribute DF_DATABASE_DEFAULT_DATABASE.

This attribute returns or changes the default database of a database connection.

The default database is the database where the connectivity kit sends its SQL commands. It can be compared to changing the Use <dbname> that is used in SQL scripts.

The default database is initially set to the database specified in the connectstring.

Example:

```
MssqlOrderID=SERVER=(local); DATABASE=MssqlOrder190;Trusted_Connection=yes
```

If logged in with this connection id the default database will MssqlOrder190

Function GetDefaultDatabase Returns String

```
Integer iDriverIndex  
Handle hConnectionHandle  
String sDefaultDatabase
```

```
Get LoadedDriverIndex of ghoConnection "MSSQLDRV" to iDriverIndex  
If (iDriverIndex) Begin  
    Get CLIConnectionHandle iDriverIndex "DFCONNID=MssqlOrderID" to hConnectionHandle  
    If (hConnectionHandle) Begin  
        Get_Attribute DF_DATABASE_DEFAULT_DATABASE of iDriverIndex hConnectionHandle to sDefaultDatabase  
        //ShowIn (SFormat("DF_DATABASE_DEFAULT_DATABASE = %1", sValue))  
    End  
End  
Function_Return sDefaultDatabase
```

End_Function

The following function will change the default database.

Procedure SetDefaultDatabase String sDatabaseName

```
Integer iDriverIndex  
Handle hConnectionHandle  
String sValue
```

```
Get LoadedDriverIndex of ghoConnection "MSSQLDRV" to iDriverIndex  
If (iDriverIndex) Begin  
    Get CLIConnectionHandle iDriverIndex "DFCONNID=MssqlOrderID" to hConnectionHandle  
    If (hConnectionHandle) Begin  
        Set_Attribute DF_DATABASE_DEFAULT_DATABASE of iDriverIndex hConnectionHandle to  
sDatabaseName  
    End  
End
```

End_Procedure

Changing the default database can be used in multi-tenant applications where multiple database with identical structures are used.

When changing the default database, opened tables will remain open. After a database change the opened tables will point to a different database. It is expected that tables in both databases have identical structures.

Example:

```
// Lets assume this returns MssqlOrder190  
Get GetDefaultDatabase to sOldDefaultDatabase  
// Open table SalesP in database MssqlOrder190  
Open Salesp  
  
// Find a record in table Salesp in database MssqlOrder190  
Clear Salesp  
Move "MM" to SalesP.Id  
Find Eq SalesP by Index.1  
  
// Switch to a different database  
Send SetDefaultDatabase "MssqlOrder182"  
  
// Table Salesp is still open, now points to database MssqlOrder182  
// Find will now find a record in table Salesp in database MssqlOrder182  
Clear Salesp  
Move "MM" to SalesP.Id  
Find Eq SalesP by Index.1
```

Notes on DF_DATABASE_DEFAULT_DATABASE

- Can only switch between databases on the same server

- Databases will be accessed with the same credentials
- Tables must have identical structures in both databases. Differences in structures will give unpredictable results

New RedirectIdConnectionString function.

This function can be used to redirect connections when using connection ids.
This function should be used instead of RedirectConnection when using connection ids

The function is defined in cCliHandler class in cli.pkg
This function can be used for switching between identical databases in multi-tenant situations.

Example:

```
// We use the following connectionId:
//  MssqlOrderID=SERVER=(local);DATABASE=MssqlOrder190;Trusted_Connection=yes

String sConnectionId
String sConnectionString

Move "DFCONNID=MssqlOrderID" to sConnectionId
Move "SERVER=(local);DATABASE=MssqlOrder182;Trusted_connection=yes" to sConnectionString

Set psDriverID of oCliHandler to "MSSQLDRV"
Get RedirectIdConnectionString of oCliHandler sConnectionId sConnectionString to iRetVal
```

This will switch from database MssqlOrder190 to MssqlOrder182

Notes on RedirectIdConnectionString

- Can switch between different databases on different servers
- Switching the database will involve login to the new connection and logout from the old connection.
- Tables must have identical structures in both databases. Differences in structures will give unpredictable results

Bugfix conversion

MSSQLDRV: CLI_BCP function.

Conversion to SQL Server.

When date columns allow nulls (Default_Nullable_Date setting in mssqldrv.int) and no dummy zero dates are used (Default_Use_Dummy_Zero_Date in mssqldrv.int) the conversion will now save 0 dates as NULL in SQL Server.

Build 6.2.0.13

Merge from build 6.1.0.30:

All CLI Connectivity Kits:

Setting the DD value of a text column, for example:

```
Set Field_Changed_Value of Customer_DD Field Customer.Comments to sText
Send Request_Save of Customer_DD
```

This code could cause memory error if:

- sText value was the same as the value already in Customer.comments
- An ansi/oem conversion was done.

All CLI Connectivity Kits:

Moving data to a df_ascii/sql_varchar column, where the data was larger than the field length, could cause a memory error.

Build 6.2.0.12

Added new embedded SQL function SQLConnectionConnect

Function SQLConnectionConnect String sDrvrlD String sConnectionString Returns Integer

SQLConnectionConnect establish an embedded SQL connection that uses an existing connection from an earlier login or open. Existing connections are identified by their connect string (can be obtained with DF_DRIVER_SERVER_NAME attribute)

If sConnectionString exist in the list of existing connections, that connection will be used for embedded SQL.

sConnectionString must be an exact match, but can be case insensitive

sConnectionString can contain a connection id. For example: "DFCONNID=MyConnectionID"

If no matching connectstring is found, this function will return 0.

This function will only work with CLI connectivity kits 6.2 or later.

Example:

Procedure TestSQLConnectionConnect

```
String sConnectionString  
String sDriverId
```

```
String[][] aResultSet
```

```
Handle hdbc  
Handle hstmt  
String sSQLQuery
```

```
Move "SERVER=(local);DATABASE=MssqlOrder190;Trusted_Connection=yes" to sConnectionString  
Move "MSSQLDRV" to sDriverId
```

```
// Login to a connection  
Login sConnectionString "" "" sDriverId
```

```
// Use the connection for embedded SQL  
Get SQLConnectionConnect of oSQLHandler sDriverId sConnectionString to hdbc
```

```
If (hdbc <> 0) Begin  
  Get SQLOpen of hdbc to hstmt  
  If (hstmt <> 0) Begin  
    Move "Select * from SalesP" to sSQLQuery
```

```
    Send SQLExecDirect of hstmt sSQLQuery  
    Get SQLFetchResultsetValues of hstmt to aResultset  
    // Send ProcessResultSet aResultSet
```

```
    Send SQLClose of hstmt  
  End  
  Send SQLDisconnect of hdbc
```

```
End  
Else Begin  
  ShowIn (SFormat("SQLConnectionConnect could not connect to %1", sConnectionString))  
End
```

End_Procedure

Build 6.2.0.11

All CLI Connectivity Kits:

Trying to open a table that does not exist in the SQL database, an "Invalid cursor state" error would be raised.
Now changed to "Table not in connection" error.

All CLI Connectivity Kits:

Added new driver level setting LOGIN_ON_OPEN that can be set to 0 (False) or 1 (True).

This setting controls whether during an open a login will be attempted if we are not already logged in on the connection.

LOGIN_ON_OPEN can be specified in connectivity kit configuration file (mssqldrv.int, db2_drv.int, odbc_drv.int) or at runtime with DF_DRIVER_LOGIN_ON_OPEN attribute.

New configuration file keyword: LOGIN_ON_OPEN

Possible values: 1 or 0

Default value: 1 (true)

New driver level attribute: DF_DRIVER_LOGIN_ON_OPEN

Possible values: 1 or 0

Example:

Get_Attribute DF_DRIVER_LOGIN_ON_OPEN of iDriverIndex to iLoginOnOpen

Set_Attribute DF_DRIVER_LOGIN_ON_OPEN of iDriverIndex to 0

LOGIN_ON_OPEN will be 1 (true) by default.

If LOGIN_ON_OPEN is 1

When opening a table and there is no connection, the connectivity kit will attempt an automatic login. This is the behaviour of earlier versions of the connectivity kit.

If LOGIN_ON_OPEN is 0

When opening a table and there is no connection, the connectivity kit will not attempt to login, but generated an error.

New cli.pkg with DF_LOGON_ON_OPEN attribute added.

Build 6.2.0.10

ODBC_DRV:

Oracle columns created as type INTEGER returned an incorrect length.

(Merge from build 6.1.0.29)

All CLI Connectivity Kits:

Changing DF_Length of integer type columns is now an int file only change (df_file_restructure_int_only). Only when length remains below max length for the integer types: bigint 14, int 10, smallint 5, tinyint 3.

All CLI Connectivity Kits:

When renumbering an index (DF_INDEX_NUMBER) that is the primary index, the df_file_primary_index must also be renumbered.

Build 6.2.0.9

Support for temporary indexes:

Temporary indexes are indexes that are created on the fly at runtime. They will only exist as long as the table is open. Temporary indexes will not be created on the server.

Temporary indexes are created like regular indexes, but they must be created outside structure_start/structure_end.

- Create_index can now be done outside structure_start/structure_end
- Setting DF_INDEX_* and DF_INDEX_SEGMENT_* attributes can now be done outside structure_start/structure_end
- An index created outside structure_start/structure_end will be a temporary index that only exists at runtime.
- Temporary index will have type DF_INDEX_TEMPORARY for DF_INDEX_SQL_TYPE.
- A temporary index can be deleted outside structure_start/structure_end with delete_index.

New cli.pkg

Added DF_INDEX_TEMPORARY as possible value for DF_INDEX_SQL_TYPE attribute.

Example:

Handle hTable
Integer iIndex
Open Customer

```

Move Customer.File_Number to htable
Create_Index hTable at iIndex

// Set Index Attributes: 2 segments;
Set_Attribute DF_INDEX_NUMBER_SEGMENTS of hTable iIndex to 2
Set_Attribute DF_INDEX_SEGMENT_FIELD of hTable iIndex 1 to 4 // Column 4 == City
Set_Attribute DF_INDEX_SEGMENT_FIELD of hTable iIndex 2 to 1 // Column 1 == Customer_Number

Clear Customer
Move "Dallas" to Customer.City
Vfind hTable iIndex GT

Send Clear of Customer_DD
Move "Dallas" to Customer.City
Send Find of Customer_DD GT iIndex

```

Build 6.2.0.9

Fixed a bug where create_index did not adjust last_index_number properly

Build 6.2.0.7

Bugfix: SQLConnect crashed when connecting with a not existing connection ID.

Bugfix: Opening table with no index definition in INT (system table !) crashed the driver.

DF_INDEX_SQL_TYPE of a recnum index (index.0) will not be set to DF_INDEX_SERVER_ONLY. Even when it is missing from the INT file will be set to DF_INDEX_SERVER.

Made changes to push (renumber) server only indexes always to the end of the index list:

When DF_INDEX_NUMBER is set to the number of a server only index.

When the DF_INDEX_SQL_TYPE of an index is set to or from DF_INDEX_SERVER_ONLY

Build 6.2.0.6

If a column appears as a segment in a server_only index, the field_index of the column will no longer be set during open.

When adding a new index (Create_Index) and there are server_only indexes present, the new index will get the number of the first server_only index. The server_only index(es) will get their index number incremented, so they remain at the end of the index list.

Added new attribute DF_INDEX_NUMBER that allows to renumber indexes.

syntax:

```
Set_Attribute DF_INDEX_NUMBER of hTable iOldIndexNum to iNewIndexNum
```

if iOldIndexNum and iNewIndexNum are both existing indexes, the indexes will be switched.

if iOldIndexNum exists and iNewIndexNum is a non-existing index, after restructure iNewIndexNum will exist and there will be no iOldIndexNum

Example:

```
Move 2 to iOldIndexNum
Move 3 to iNewIndexNum
```

```
Structure_Start hTable "MSSQLDRV"
Set_Attribute DF_INDEX_NUMBER of hTable iOldIndexNum to iNewIndexNum
Structure_End hTable DF_STRUCTEND_OPT_NONE
```

If index 2 and index 3 are both existing indexes, after restructure the indexes will be switched.

If index 2 exists and index 3 is a non-existing index, after restructure index 3 will exist and there will be no index 2.

New cli.pkg with DF_INDEX_NUMBER added

New MSSQLDRV_DriverDef.xml with DF_INDEX_NUMBER added

Build 6.2.0.5

New attribute DF_FILE_RESTRUCTURE_INT_ONLY added.

This attribute returns true if structural changes were made, that requires ONLY writing a new table INT file.
This attribute cannot be set.

The DF_FILE_RESTRUCTURE_INT_ONLY attribute provides additional information to the DF_FILE_RESTRUCTURE attribute.

If DF_FILE_RESTRUCTURE_INT_ONLY returns true, there will be no structural changes on the server; only the INT file will be changed. If DF_FILE_RESTRUCTURE_INT_ONLY returns false, the DF_FILE_RESTRUCTURE attribute will indicate if and what kind of restructure is needed.

Examples of changes that will require only rewrite the INT file:

- Adding/removing/changing client side indexes
- Change DF_Length of varchar(max) and varbinary(max) columns
- Change the DataFlex type of SQL datetime column.

Note: DF_FILE_RESTRUCTURE_INT_ONLY will only return true if there are **only** INT file changes. If there are additional changes that do require server side changes, DF_FILE_RESTRUCTURE_INT_ONLY will return false.

New cli.pkg with DF_FILE_RESTRUCTURE_INT_ONLY attribute added.

Build 6.2.0.4

New index attribute DF_INDEX_SQL_TYPE added. This attribute replaces DF_INDEX_ON_BACKEND attribute.
DF_INDEX_ON_BACKEND remains as a read-only attribute for backward compatibility, but should no longer be used.

The new DF_INDEX_SQL_TYPE attribute can have following values:

- DF_INDEX_CLIENT
- DF_INDEX_SERVER
- DF_INDEX_SERVER_ONLY

DF_INDEX_CLIENT

The index is a client side index. The index is not defined on the server. The index is defined in the table INT file.
The index definition in the INT file must define the index number, the index name and the index segments.

Example:

```
INDEX_NUMBER 3
INDEX_NAME Vendor003_CS
INDEX_NUMBER_SEGMENTS 2
INDEX_SEGMENT_FIELD 4
INDEX_SEGMENT_FIELD 1
INDEX_SEGMENT_DIRECTION DESCENDING
```

Index Vendor003_CS cannot be defined on the server.

DF_INDEX_SERVER

The index is a server side index. The index must be defined on the server. To map the SQL index name to a DataFlex index number, the index_number and index_name must also be specified in the table INT file.

Example

```
INDEX_NUMBER 2
INDEX_NAME Vendor002
```

DF_INDEX_SERVER_ONLY

The index is a server side index. The index must be defined on the server. The index is not defined in the table INT file.

New MSSQLDRV_DriverDef.xml file.

Added DF_INDEX_SQL_TYPE attribute.

New cli.pkg

Added DF_INDEX_SQL_TYPE attribute.

Build 6.2.0.3

All CLI Connectivity Kits:

During conversion recnum index would be created as client side index.

Build 6.2.0.2

All CLI Connectivity Kits:

The default for DF_INDEX_ON_BACKEND was false when creating new index on new table. This caused all indexes to be created as client side indexes during conversion.

Build 6.2.0.2 (merge from 6.1.0.28)

All CLI Connectivity Kits:

On a standard table with an identity column (df_file_identity = true), when inserting or deleting a column BEFORE the identity column, the identity would not be adjusted. This caused identity insert error during restructure.

Build 6.2.0.1

Improved support for client side indexes.

Attribute DF_INDEX_ON_BACKEND is now a read/write attribute. Default when creating new index is true. During restructure the index will move from client to server (or vice versa) if DF_INDEX_ON_BACKEND is changed. If DF_INDEX_ON_BACKEND not changed, restructure will keep the index were it was.

Changed MSSQLDRV_DriverDef.xml:

df_index_on_backend <readOnly> changed from true to false

Build 6.1.0.32

- All CLI Connectivity Kits: Setting a constrain with accented characters did no OEM/ANSI translation. This was side effect of an earlier fix in 6.1.0.30. Now that is fixed - the translation is happening and constraints should work as expected.

Build 6.1.0.31

- All CLI Connectivity Kits: Bugfix: Setting the value of a SQL_TIME column, could cause memory overwrite. This was side effect of earlier fix in 6.1.0.30 A masked time value of 00:00:00 would be passed as " : : ", causing a "Time contains an invalid value" error. This will now be handled as time value 00:00:00.

Build 6.1.0.30

- All CLI Connectivity Kits: Setting the DD value of a text column, for example:

Set Field_Changed_Value of Customer_DD Field Customer.Comments to sText

Send Request_Save of Customer_DD

This code could cause memory error if:

-sText value was the same as the value already in Customer.comments

-An ansi/oem conversion was done.

- All CLI Connectivity Kits: Moving data to a df_ascii/sql_varchar column, where the data was larger than the field length, could cause a memory error.

Build 6.1.0.29

- ODBC_DRV: Oracle columns created as type INTEGER returned an incorrect length.

Build 6.1.0.28

- All CLI Connectivity Kits: On a standard table with an identity column (df_file_identity = true), when inserting or deleting a column BEFORE the identity column, the identity would not be adjusted. This caused identity insert error during restructure.

Build 6.1.0.27

- All CLI Connectivity Kits: When calling a stored procedure (SQLCall) that generated informational messages, the SQLGetMessage function would not return those informational messages.

Build 6.1.0.26

- All CLI Connectivity Kits: Fixed memory leak in finding/saving. This leak was caused by a change in build 6.1.0.20.

Build 6.1.0.25

- All CLI Connectivity Kits: DF_ASCII columns where the underlying type is SQL_WVARCHAR will be trimmed when saved to the database. WVARCHAR are wide variable length types like NVarchar on SQL Server. This was already implemented earlier for DF_ASCII/SQL_VARCHAR types.

Build 6.1.0.24

- MSSQLDRV: When calling a stored procedure with a varchar(max) parameter, the varchar(max) parameter was not passed.

Build 6.1.0.23

- All CLI Connectivity Kits: Saving to datetime2 type columns would sometimes save incorrect values. (PM values would be saved as AM values). This could happen when regional settings were set to use 12-hour clock (AM/PM notation).

Build 6.1.0.22

- All CLI Connectivity Kits: When changing DF_FILE_LOGIN from normal connect string to DFCONNID connect string that points to the same database, the restructure process would hang.
- All CLI Connectivity Kits: Studio would crash at breakpoint if a table had ASC column with length above 255.

Build 6.1.0.21

- All CLI Connectivity Kits: On a standard table, having numeric columns with decimals in the primary index, the rowid after a save could be different from the rowid after a find. This could cause incorrect behavior in grids, for example the grid would be cleared after save of a new record.

Build 6.1.0.20

- All CLI Connectivity Kits: Using a SQL filter with a length above 64K caused memory overwrite.

Build 6.1.0.19

- ODBC_DRV: Some Oracle data types were not properly recognized and would be mapped as DF_ASCII. The Oracle TIMESTAMP type will now be mapped as a DF_DATETIME type. The Oracle RAW type will now be mapped as a BINARY type.

Build 6.1.0.18

- MSSQLDRV: When trying to save a duplicate record on a table that has a SQL Primary key defined, an unhandled SQL error was generated. The connectivity kit will now generate a handled error (Duplicate records not allowed (error 28)) in this situation.

Build 6.1.0.17

- ODBC_DRV: Descending index segments on Oracle were not properly recognized.
- All CLI Connectivity Kits: In some circumstances the int file for a table could be overwritten with the contents of a cch file. This could happen if the df_open_path was changed at runtime and the int file or the cch file could not be located in df_open_path.
- All CLI Connectivity Kits: Increased the maximum possible connection string length, as it was too short for some DSN-less connections.

Build 6.1.0.16

- All CLI Connectivity Kits: A SQL syntax error was generated on a find with following characteristics:
 - o Find LT on multi segment index
 - o First index segment nullable and seeded with value NULL
 - o SQL Filter defined
- ODBC_DRV: Changes in MySQL text and blob type handling:
 - o The MySQL longtext and longblob types returned an invalid length -1.
 - o All MySQL text types (text/mediumtext/longtext/tinytext) and blob types (blob/mediumblob/longblob/tinyblob) will now preserve their type in a restructure

Build 6.1.0.15

- ODBC_DRV: The Oracle ODBC driver did not return default values for columns. Build a workaround to retrieve column default values when connected to Oracle.

Build 6.1.0.13

- All CLI Connectivity Kits: DF_ASCII columns where the underlying type is SQL_VARCHAR will be trimmed when saved to the database. Previous DF_ASCII/SQL_VARCHAR columns would not be trimmed. This could cause find issues on Oracle since Oracle compares trimmed strings as not equal to space padded strings.
- ODBC_DRV: On Oracle, finding on an index with one or more nullable segments, would generate incorrect sort order.

Build 6.1.0.12

- MSSQLDRV: Fixed an issue with finding on cleared GUID column in a multi segment index.
- ODBC_DRV: Improved support for primary keys, foreign keys and auto_increment columns on MySQL.

Build 6.1.0.11

- DB2_DRV: On a standard table with an identity column, after save of a new record, the assigned identity value was not returned in the buffer.
- DB2_DRV: Removed length limitation of 18 characters on DB2 index names. Index names on DB2 can be 128 long.
- ODBC_DRV: On PostgreSQL, restructure of an existing table would cause syntax errors.
- All CLI Connectivity Kits: Fixed a problem with overlap handling.

Build 6.1.0.10

- All CLI CK's: Changed the default for Find_Cache_Timeout to 100 (milliseconds).

Build 6.1.0.9

- MSSQLDRV: Finding on empty GUID column would in some situations generate casting errors.

Build 6.1.0.8

MSSQLDRV:

- Table with both identity column and GUID column with default newid() , would generate error on save.
- Generate better error when saving GUID column with invalid value.
- Fixed problem with sort order of GUID columns

ODBC_DRV:

- Fix for Oracle Date/Datetime handling: The Oracle DATE type can contain both DATE and DATETIME values. By default an Oracle DATE type will be mapped to DataFlex DF_DATE type. To map Oracle DATE to DataFlex DF_DATETIME the following line: FIELD_TYPE DATETIME should be specified in the <table>.int file.
- The ODBC connectivity kit did not detect this keyword and would not map the Oracle DATE to DATETIME.
- Changed Oracle.int file: Added default_default_datetime setting.

Build 6.1.0.7

- MSSQLDRV: Improved support for GUID (UniqueIdentifier) columns.

Columns of type uniqueidentifier can have a default value of newid() or newsequentialid(). On insert (save of a new record) the database will assign a new GUID value to the column. The Connectivity Kit will retrieve the new assigned GUID value in the file buffer after save of a new record.

Find logic on uniqueidentifier columns has been improved to avoid casting errors on find.

When creating new uniqueidentifier columns in the Studio, the default value for the column can be set in the properties panel with the df_field_default_value attribute.

Note: Values provided for df_field_default_value, will normally be treated as a string. Single quotes will be added by the connectivity kit.

To enter database functions like *newid()* or *newsequentialid()* as the default value, the value must be entered with square brackets: **[newid()]** or **[newsequentialid()]**

- MSSQLDRV: Changed handling of identity columns on standard tables during save.

The connectivity kit generates a SQL INSERT statement when saving a new record and a SQL UPDATE statement when saving an existing record.

For a standard table with an identity (auto increment) column, the connectivity kit will no longer include the identity column in the generated INSERT or UPDATE statement.

This has the following consequences:

- When saving a new record a new value for the identity column will always be assigned by the database backend.
- When saving an existing record the value for the identity column will remain unchanged.
- When saving a record (either new or existing) any value that was put in the field buffer will be ignored.

Example:

Move 123 To TheTable.IdentityColumn

Move 'SomeValue' To TheTable.SomeOtherColumn

SaveRecord TheTable

With this code, the value moved to the identity column will be ignored during save.

If this saves a new record a new identity value will be assigned by the database.

If this saves an existing record, the value of the column will remain as it was.

Build 6.1.0.6

- All CLI Connectivity Kits: When creating a primary key, the name of the primary key will have _PK added. (For example: **Customer001_PK**). Note: Existing index or primary key names will not change during a restructure; they can be changed manually with the Df_Index_Name attribute.
- All CLI Connectivity Kits: After changing an index name (Df_Index_Name), the connectivity kit would generate errors in restructure. This was caused by the wrong index name being used.

Build 6.1.0.5

- MSSQLDRV: Made changes to enable conversion to and restructure on SQL Azure. SQL Azure requires each table to have a clustered index. During conversion and restructure all indexes are dropped and re-created for performance reasons. For SQL Azure clustered indexes will no longer be dropped and re-created during restructure and conversion.
- MSSQLDRV: Changed DROP INDEX syntax for SQL Server to: DROP INDEX <IndexName> on <schema>.<TableName> This syntax is required for SQL Azure.
- MSSQLDRV: After a restructure database triggers will be preserved. Earlier versions of the Connectivity Kit would drop triggers during a restructure. For the moment only implemented for SQL Server Connectivity Kit.

Build 6.1.0.4

- All CLI Connectivity Kits:

System table changes

To fully support system tables on SQL Azure changes have been made in the handling of system tables.

Before Connectivity Kit version 6.1 system tables could not have any indexes. Since SQL Azure requires a clustered index on all tables, changes have been made to allow indexes on system tables.

In Connectivity Kit version 6.1 two types of system tables are supported:

- Default system tables
- Real recnum system tables

Default system tables.

- Have no recnum column on the SQL side
- Usually have no indexes, but it is possible to define indexes.
- On the DataFlex side system tables are always recnum tables.
- The Connectivity Kit maintains an internal virtual recnum.
- The primary index is always 0 (Recnum index)

Real recnum system tables.

- These are legacy system tables from older connectivity kit versions. They are supported for backward compatibility.
- Have a Recnum column as the first column in the table.
- Have an index on the recnum column.
- Is always a recnum table
- Primary index is always 0

For SQL Azure it is required to have a clustered index on any table. To use system tables on SQL Azure a clustered index should be defined on a column in the system table.

The column can be any column already in the table or a new column can be added.

The only purpose of adding an index is to meet the SQL Azure requirement of having a clustered index on any table. It is not necessary to make program changes.

- All CLI Connectivity Kits: Changing a recnum table to a standard table by setting Df_File_Recnum_Table attribute to false, would generate errors. Improved handling to convert recnum tables to standard tables and vice versa.
- DB2_DRV: Added support for identity columns on standard tables for DB2.
- DB2_DRV: Preserve foreign keys on restructure for DB2.

6.0.0.32

- DB2_DRV: Connecting to a DB2 table with the DB2 Connection Wizard would fail with 'Cannot open table on backend' error. The earlier fix in build 6.0.0.31, did not cover all situations.

6.0.0.31

- MSSQLDRV: Dbbuilder conversion to SQL Server (CLI_BCP): Converting DF_ASCII columns to varchar or nvarchar would create the (n)varchar columns padded with spaces. DF_ASCII columns will now be trimmed when converting to varchar or nvarchar.
- DB2_DRV: Connecting to a DB2 table with the DB2 Connection Wizard would fail with 'Cannot open table on backend' error. This was caused by the connectivity kit incorrectly determining the table type (standard or recnum table) and the record_identity_method.

Build 6.0.0.30a

- Sql.pkg (Embedded SQL): When executing 2 embedded SQL queries in succession and the first result set is empty, invalid statement handle errors could occur when fetching from the second result set. Solved by clearing giLastSQLhstmt in SQLClose and giLastSQLhdb in SQLDisconnect
- Marked the following properties as private:
paSQLColumns

psDummyZeroDate
psDummyZeroDateMssqlDatetime

6.0.0.30

- MSSQLDRV: Fix for DF_FIELD_INDEX, DF_FIELD_RELATED_FILE and DF_FIELD_RELATED_FIELD attributes sometimes returning incorrect values. This happened in following situation:
Processing large number of tables, querying all table attributes. (StSync tool).
Hidden columns present (NEXT_COLUMN_HIDDEN in INT file)

6.0.0.28c

- Sql.pkg (Embedded SQL): After an embedded SQL call that returned multiple result sets, the statement properties (like piColumnCount) were only stored for the first result set. SQLNextResultSet will now store the properties for each result set.

6.0.0.28

- ALL CLI CK's: (Embedded SQL): The column number can now be passed with FUNC_SQLGETDATA. This gives the possibility to reduce the number of call_driver calls when calling SQLGetData.

The new functions SQLFetchRowValues and SQLFetchResultsetValues will now also return variable data.

Changes to embedded sql (sql.pkg).

- Removed character mode (DF32) syntax:
 - Replaced all Class Array by Class cObject
 - Removed all local declarations
 - Replaced all Current_Object by self
 - Removed Use case.mac
- Marked private properties and methods with { Visibility=Private } so they do not longer end up in the documentation.
- Changed some of the private property / method names. Public interface is unchanged.
- Syntax changes (integer -> handle/ create/destroy object)
- Changes in DFDateToSQLDate and SQLDateToDFDate functions
 - No longer use cCliHandler
 - Only get dummy zero date value from driver for ODBC_DRV. Mssqldrv and db2_drv have fixed dummy zero date.
 - Handle 2 dummy zero dates for SQL Server: 1753-01-01 for sql_type_timestamp, 0001-01-01 for sql_type_date

PLEASE NOTE:

Important notes about Embedded SQL (sql.pkg)

This version of sql.pkg will only work with CLI Connectivity Kits 6.0.0.28 or later. For earlier Connectivity Kit versions use sqlold.pkg.

This version of sql.pkg will no longer function with DataFlex character mode (DataFlex 3.2). To use embedded SQL with DataFlex character mode, use sqlold.pkg.

This version of sql.pkg no longer contains the embedded SQL command interface. The command interface is still available in sqlold.pkg

6.0.0.27

- ALL CLI CK's: (Embedded SQL): The column number can now be passed with FUNC_SQLCOLUMNVALUE. This gives the possibility to reduce the number of call_driver calls when calling SQLColumnValue.

New sql.pkg for embedded SQL:

- Several performances improvements to speed up getting a value with SQLColumnValue
- New functions:
 - Function SQLFetchRowValues Returns String[]
Fetches a row and returns all columns in an array.
 - Function SQLFetchResultsetValues Returns String[][]
Fetches all rows of a result Set and Returns the result set as a 2-dimensial array.
- Removed command interface. (Still available in sqlold.pkg)

The new sql.pkg can only be used with connectivity kits version 6.0.0.27 and later.

The previous version of sql.pkg is available as sqlOld.pkg and can be used with connectivity kits version before 6.0.0.27.

6.0.0.26

- MSSQLDRV: On a sql server datetime column: changing the DF_Type from DF_Date to DF_Datetime would sometimes set the sql type to "unknown type"
- ALL CLI CK's: Setting Native_Type to "unknown type" generated an error that the type is not supported by the backend.

6.0.0.24

- MSSQLDRV: Setting DF_FIELD_NATIVE_TYPE to "datetime2" an incorrect length/precision would be set.
- ALL CLI CK's: When setting DF_FIELD_NATIVE_TYPE, the Connectivity Kit will check if the type is supported by the database server and raise an error if the type is not available. This might be the case for example for the **date** and **datetime2** type not being available on SQL Server 2005 and earlier.

6.0.0.23

- ALL CLI CK's: When DF_DATE_FORMAT was set to DF_DATE_MILITARY an error 4523 *DateTime contains an invalid value*, could be generated:
By the FillField command.
On constrained Finds where there is at least one date in the index.

6.0.0.22

- ALL CLI CK's: Changed the DF_DRIVER_CONFORMANCE level. This attribute will now return the value 7. The DF_DRIVER_CONFORMANCE attribute must be handled as a bitwise integer value, where each bit indicates whether the Connectivity Kit supports a certain feature.

DF_DRIVER_CONFORMANCE now indicates the following features:

- bit 1 Driver supports rowid
- bit 2 Driver supports datetime type
- bit 3 Driver supports DD SQL filters (added in build 6.0.0.22)

- DB2_DRV: Made changes for setting the native type for DB2. Handling of the following DB2 types was changed GRAPHIC, VARGRAPHIC, LONGVARGRAPHIC, TIME, DBCLOB, BLOB New DB2_DRV_DriverDef.xml file
- MSSQLDRV: New MSSQLDRV_DriverDef.xml file

6.0.0.21

- ODBC_DRV: The Oracle DATE type will now be mapped to DF_DATE by default. It can be mapped to DF_DATETIME by specifying FIELD_TYPE DATETIME in the <table.int> file. Made changes to Oracle.int configuration file to reflect latest Oracle ODBC driver.
- DB2_DRV: The DB2 xml type does not allow a default value. DB2 xml columns will now be created as nullable, without a default value. DB2 xml type will always be mapped to DF_TEXT.
- MSSQLDRV: SQL Server xml type will always be mapped to DF_TEXT.

6.0.0.20

- MSSQLDRV: Improved support for setting/getting native types. Added support for uniqueidentifier and time types. Changed various char/varchar to DF type mappings.

6.0.0.19

- MSSQLDRV: Improved support for setting/getting native types. Added new Df_Field_Native_Type values for SQL Server (var...(max), money and datetime types).
- ALL CLI CK's: The DF_length for integer and small integer type columns was too short. Changed maximum/default df_length for SQL_Integer and SQL_SmallInt columns: SQL_Integer from NUM(9.0) to NUM(10.0) SQL_SmallInt from NUM(4.0) to NUM(5.0) This change will also allow recnum columns to store max value 2.147.483.647 (10 digits) instead of max 999.999.999 (9 digits).
- ALL CLI CK's: In a situation where a date column is part of an overlap:
Move <somevalue> to TheTable.Overlap
This would sometimes put an incorrect date value in the overlapped date column. On save this would cause an 'invalid character value for cast specification' error.

6.0.0.18

- ALL: Added new attribute DF_FILE_SQL_FILTER_EQ. When set to true SQL Filters will be applied to all Find EQ. The default value is false. Earlier versions of the Connectivity would never apply SQL Filters to Find EQ. This change includes an updated cli.pkg with the DF_FILE_SQL_FILTER_EQ attribute added. When a SQL Filter changes between 2 consecutive Finds and there was no find cache timeout: the second find would return the next record from the find cache, instead of applying the changed SQL Filter.

6.0.0.16

- ODBC_DRV: Fixed a memory overwrite when connected to Oracle.

6.0.0.15

- ALL: SQLBindFile generated error 4523 *DateTime contains an invalid value* on DF_DATE columns. SQLBindfile did not convert DF_DATETIME columns to the correct format and returned incorrect values in the file buffer.

6.0.0.14

- ALL: Fill_Field for DF_DATE type columns did not work in DF32 console mode. Earlier changes (build 10 and 11) did not fix it. The logic to determine if the Connectivity Kit is running on a DF32 runtime was changed.
- ALL: Only on DF32, with date4_state set to true: Fill_Field DF_LOW for DF_DATE field would fill the field with 2001/01/01, instead of 0. This also caused constrained finds on indexes with a date column to fail.

6.0.0.12

- ALL: All connectivity kits version 6 now need a valid Connectivity Kit version 6.x registration code. Note: No separate connectivity kit license is needed when using a "Connectivity Enabled" Visual DataFlex license. Note that all Visual DataFlex Studio licenses (12.1 or newer) are "Connectivity Enabled".

6.0.0.11

- DB2_DRV for Linux: Fill_Field for DF_DATE type columns did not work in DF32 for Linux.

6.0.0.10

- ALL: Fill_Field for DF_DATE type columns did not work in DF32 console mode.
- DB2_DRV for Linux: Db2_drv for Linux did not load on some Linux systems.
- DB2_DRV: Changed DB2_DRV_DriverDef.xml. Added the following tablespace attributes:
DF_FILE_TABLE_TABLESPACE
DF_FILE_INDEX_TABLESPACE
DF_FILE_LONG_TABLESPACE
- Packages: Removed "// DOC tags" in Db2_drv.pkg and Odbc_drv.pkg

6.0.0.9

- ODBC_DRV: Made changes to improve performance on Oracle.

6.0.0.8

- ODBC_DRV: Microsoft Access has only a datetime type and no date type. Handling is now the same as for SQL Server datetime type.
MS Access datetime is DF_Date by default.
MS Access datetime is DF_Datetime if FIELD_TYPE DATETIME is specified in the table int file.

6.0.0.7

- SQL Server: A begin_transaction after the last transaction was aborted, would not try to reconnect.
- DB2: Added SQL_State "08007" as a connection loss state for DB2.
- Fixed typo's in mssqldrv.int, odbcdrv.int, db2_drv.int and What's New document.

6.0.0.6

- Improvements for reconnect in begin_transaction.

6.0.0.5

- Bug fix in get_attribute Df_Field_Native type.

6.0.0.4

- Removed the MAP_DF_TO_SQL_TYPE_CK5 and MAP_DF_TO_SQL_TYPE_CK6 value for DF_DRIVER_DEFAULT_MAP_DF_TO_SQL_TYPE_SCHEMA attribute.
This attribute can now only be used for SQL Server:
Possible values: MAP_DF_TO_SQL_TYPE_SQL2xxx
For DB2 and ODBC no MAP_DF_TO_SQL_TYPE_SCHEMA can be set.
- Made connection id's cases insensitive.

6.0.0.3

- Changes for handling setting of native type.
- Changed MSSQLDRV_DriverDef.xml and DB2_DRV_DriverDef.xml
With these changes it is possible to change native type of a column in the Studio.
The MSSQLDRV_DriverDef.xml and DB2_DRV_DriverDef.xml now only contain the types that are supported by MSSQL and DB2.
- Changed cli_drv.pkg, mssqldrv.pkg, db2_drv.pkg. Defines all supported native types for mssql and db2
- Changed What's New document. Migrating existing databases to new types section

6.0.0.2

- Includes db2_drv Linux build. Some minor changes for Linux.
- All: When using embedded SQL with SQLBindfile on a recnum table, the recnum value would not be returned in the buffer. This change was also made in 5.1.0.101

6.0.0.1

Initial build. Functionally equivalent to 5.1.0.100

5.1.0.101

- ALL: When using embedded SQL with SQLBindfile on a recnum table, the recnum value would not be returned in the buffer.

5.1.0.100

- ODBC_DRV: During restructure of a MySQL recnum table the auto_increment setting of the recnum column would be lost. Made a change to preserve the auto_increment on MySQL recnum tables during restructure.

5.1.0.99

- MSSQLDRV: Added support for SQL Server time columns.

5.1.0.98

- ODBC_DRV: Made changes to better support MySQL 5.5.

The handling of MySQLTableType setting/attribute has been changed:

MySQLTableType is a clause that must be added to the MySQL Create Table statement This used to be TYPE = <typename> in earlier MySQL versions, but in MySQL 5.5, it can only be ENGINE = <typename>

Also the possible type names have changed. Dependent on MySQL version some types no longer exist and new ones have been added.

Earlier versions of the ODBC Connectivity Kit had only a fixed list of types to choose from. As of build 5.1.0.98 the setup has been made more flexible.

The MySQLTableType clause should be specified in the MySQL.int configuration file, or with the DF_DATABASE_MYSQLTABLETYPE attribute.

The full clause, as it will be added to the Create Table statement should be specified.

MySQL.int file example:

MySQLTableType ENGINE = InnoDB
(add no quotes, spaces are allowed)

Set Attribute example:

Set_Attribute DF_DATABASE_MYSQLTABLETYPE of iDriverId hDatabase
to "ENGINE = InnoDB"

Default setting:

ENGINE = InnoDB

Specifying an empty string will turn the setting off.

The fixed named types can still be used for backward compatibility reasons.

Changed MySQL.int file.

- Changed comments/example for the MYSQLTABLETYPE keyword.
- Changed the settings for DF_Text and DF_Binary columns to nullable with no default, since MySQL does not allow defaults on BLOB and TEXT columns.

5.1.0.97

- MSSQLDRV: When using a trigger on a recnum table, in some situations an error on GetScopeIdentity.Fetch could be generated. This would happen if the trigger code generated additional result messages (x row(s) affected).

5.1.0.96

- MSSQLDRV: The function EnumerateServers in mssqldrv.pkg always used the SQL Server 2000 client odbc driver. This has now been changed to use the 'highest' installed client odbc driver. Changes were made in mssqldrv.pkg and cli.pkg.

- ALL: The statement Move (NullDateTime()) to DatetimeColumn (where DatetimeColumn is a nullable column), would generate an "Invalid character value for cast specification" error. This has now been fixed. The value will be stored as a NULL value.

5.1.0.95

- ALL: Lifted 'Invalid statement handle' errors after end_transaction or unlock inside embedded SQL loop.

Versions of the connectivity kit before 5.1.0.95 would generate 'Invalid statement handle' errors when inside an embedded SQL loop a transaction commit (end_transaction or unlock) or transaction abort (abort_transaction) occurred.

This happened because the Connectivity Kit would close all statement handles on all connections (including embedded SQL connections) after a transaction commit.

As of build 5.1.0.95 the Connectivity Kit will no longer close the statements on embedded SQL connections that are opened with the SQLConnect function.

Statements on shared embedded SQL connections (opened through SQLFileConnect function) will still be closed and will generate 'Invalid statement handle' errors after commit.

So if you want to perform transactions (table updates) inside an embedded SQL loop, you must use the SQLConnect function to create the embedded SQL connection.

5.1.0.94

- ALL: When the length of a binary column was specified in the INT files and it was larger than 32768, the allocated buffer would be too small. This could cause truncated data or cause memory overwrites.
- ALL: Changed the getrowid function to trim padding spaces from df_ascii fields when creating the rowid.
- Buffer values for df_ascii fields can sometimes be padded with spaces and sometimes trimmed with no trailing spaces. This would lead to different rowid values. The record would be found by both rowid's, but the IssameRowid function would return false.

5.1.0.93

- ODBC_DRV: When connected to a SQLBase database, the SQLBase ODBC driver returned too short lengths for maximum table name length, column name length, schema name length and cursor name length.
- Created new configuration file keywords to override these maximum lengths:

```
MAX_TABLE_NAME_LEN 30
MAX_COLUMN_NAME_LEN 30
MAX_SCHEMA_NAME_LEN 30
MAX_CURSOR_NAME_LEN 30
```

These keywords can be specified in a database specific configuration file like SQLBase.int or Oracle.int, etc.

- cli.pkg: Added SQLSERVER2012CLIENT value for DF_DRIVER_SQLSERVER_CLIENT_VERSION attribute.

5.1.0.92

- MSSQLDRV: The MSSQL Connectivity Kit will detect and use the SQL Server 2012 client (SQL Server Native Client 11.0) when it is installed on the client machine.

5.1.0.91

- ALL: Find last record on a table with Table_Character_Format set to OEM, would not always find the correct record. This was caused by an incorrect ansi/oem conversion in determining the highest collating value of the backend.

5.1.0.90

- ALL: An INT file containing a FIELD_NUMBER larger than the actual number of columns of the backend table, would cause a memory overwrite with unpredictable results. The Connectivity Kit will now raise an error on an invalid FIELD_NUMBER in the INT file.

5.1.0.89

- ALL: Changed the syntax of the open method introduced in build 5.1.0.88.

The open with connect string method now uses the following syntax:

```
<drivername>:[<schemaname>#]<table>@<connectstring>[[options[<name>=<value>]...]
```

Currently the following |options exist:

```
|intfile=<filepath>.int
```

Example:

```
MSSQLDRV:orderhea@SERVER=MYServer;Trusted_Connection=yes;DATABASE=Orders|options|intfile=Orderhea.int
```

The open string can now be followed by the 'options' keyword and one or more name=value pairs.

If the intfile option is not present the table will be opened without an int file. (This is unchanged from earlier builds.)

If the intfile option is provided:

- The int file will be used to open the table.
- The DRIVER_NAME, SERVER_NAME, DATABASE_NAME and SCHEMA_NAME keywords from the INT file will be skipped. Those values will be taken from the open string.
- Cache files will not be used.
- Restructure of the table is not possible.

New error message added (dferr003.dat)

DFODBCERR_INVALIDCONNECTSTRINGOPENOPTION /* 12342 */
Invalid connect string open option

- ALL: An INT file containing a FIELD_NUMBER larger than the actual number of columns of the backend table, would cause a memory overwrite with unpredictable results. The Connectivity Kit will now raise an error on an invalid FIELD_NUMBER in the INT file.

5.1.0.88

- ALL: Added INT file support for opening files using the direct open method (<CKId>:<Schema>#<Table>@<ConnectionString>|<Filename>.int")

Now the @<ConnectionString> can be followed by a pipe character '|' followed by the name of the INT file. Doing so, the driver will apply the information from the .INT file to the passed direct open string.

Note that when an INT file has been passed, it will skip the keyword DRIVER_NAME, SERVER_NAME, DATABASE_NAME and SCHEMA_NAME from the INT file because those are supposed to be supplied in the string already. For example:

"MSSQLDRV:dbo#customer@SERVER=.\\SQLEXPRESS;Trusted_Connection=yes;DATABASE=OrderEntrySQL|CUSTOMER.INT" AS Customer

5.1.0.87

- ALL: Getting the value from a binary column when using the driver with the DataFlex Console Mode runtime would not return the full length of the binary data.

5.1.0.86

- ALL: The way the FILL_FIELD function works has been changed for DATE and DATETIME based columns. When filling the column with DF_LOW, it will put a 0 value in the column instead of the lowest possible date or datetime value for the database. The connectivity kit will take care of converting the 0 value into the lowest possible value when actually accessing the database. This change makes the FILL_FIELD command work consistently across all connectivity kits.

5.1.0.85

- ALL: Using SQLBindFile on a table that has the TABLE_CHARACTER_FORMAT set to ANSI would use an OemToAnsi translation too many, causing characters in the higher regions of the ASCII table (>127) to be wrong.

5.1.0.84

- ALL: Calling the function GetRowId for exactly the same row could lead to row id's that were not exactly the same. Calling FindByRowID would always return the same row. Comparing the row id's using IsSameRowID could return a false condition.

5.1.0.83

- MSSQLDRV: Fixed a bug where converting a table to Microsoft SQL Server _with_ recnum support would assign a different value to a recnum field of the row when there recnums are not numbered sequentially. This results in rows getting assigned a different (too low) recnum value after a 'hole' in the recnum series.

5.1.0.82

- MSSQLDRV: The method for determining the actual used default collating sequence of a server has been changed to speed up the login process of the driver.

5.1.0.81

- ALL: Functions for CreateConnectionID and DeleteConnectionID have been made case-insensitive.

5.1.0.80

- ALL: Redirecting a connection using the function RedirectConnection did not change a connection when the old connection string was completely present in the new connection string.

- ALL: The function RedirectConnection now supports redirecting to the same connection string. This might seem strange, but it allows one to redefine an existing connection id and then let all tables using that connection id reconnect by executing the following code (assuming your connection id is named MyId):

```
Get DeleteConnectionID of hoCLI "MyId" -1 to iRetVal
Get CreateConnectionID of hoCLI "MyId" ;
"SERVER=(local);UID=sa;PWD=sa;DATABASE=MyDb" 1 to iRetVal
Get RedirectConnection "DFCONNID=MyId" "DFCONNID=MyId" to iRetVal
```

- ALL: An error will be triggered when a connection ID is created that already exists.
Error: 12341: DFODBCERR_CONNECTIONIDALREADYEXISTS

5.1.0.79

- ALL: Tables opened without INT file that have a primary index on a numeric column were assumed to get a value assigned for the column by the database when inserting new rows. Because of this, tables that have primary index on a numeric column without being 'autoincrement', would not get their value set.
This has been changed so that only if the primary index is on a column type that is equal to the database's identity type (e.g. 'int identity'), then it will be assumed that the database automatically assigns a value.

5.1.0.78

- ALL: Finding on an empty file buffer on an index that has uppercase segments could generate error messages. This was introduced in release 73.

5.1.0.77

- ALL: The SQLBindFile procedure would not work well if the executed SQL statement would return a mix of columns that could be mapped to a base table and columns that could not be mapped to a base table. More specifically, Microsoft SQL Server would return a base table for unlimited varchar columns, but not for other columns. This has been fixed by ignoring the base table names for unlimited varchar columns.

5.1.0.76

- ALL: The driver would truncate data when using embedded SQL and getting the value for a numeric column that was using all positions for that field. For example, for a 4.2 defined field with a value of 1234.56, the driver would return 1234.5. This is fixed. Please note that this also requires a change to the SQL.PKG.

5.1.0.75

- MSSQLDRV: Reading newly assigned value for tables with recnum support and which don't have the recnum column defined as the first column of the table now select an index will now select an index, preventing the driver to crash with a null pointer reference.

5.1.0.74

- ALL: CCH files are now created by default in the same location where the .INT file is, instead of using the first folder of the DFPATH. The new way is the way it was documented. Use the CACHE_PATH attribute to explicitly define where cache files should be put.

5.1.0.73

- ALL: Doing a FIND GE, GT, LE or LT on a clear buffer using an index that contains segments on nullable fields which are neither the first nor the last segment, could make the next segment using the wrong value for seeding the index.

5.1.0.72

- ALL: The selection process for mapping a DataFlex field type to a database column type has been altered for numeric fields that do not have decimals. The driver could select an integer size that could not completely fit a field's requested length. For example, a numeric field in DataFlex defined as 10 digits with no decimals, could be assigned to a 32-bit integer. The 32-bit integers can hold 10 digits, but only up to 4294967296 or 2147483648 depending on the sign. The CK will now scale the column type up to a NUMERIC or DECIMAL type.

5.1.0.71

- ALL: A clear now initializes a GUID type field to all 0's, otherwise performing a find EQ on a cleared buffer that has a GUID in the index would generate a SQL error message about the format of the GUID not being correct.

5.1.0.70

- MSSQLDRV: The method for getting an assigned rowid (recnum) after a record has been created has changed from IDENT_CURRENT into SCOPE_IDENTITY because the IDENT_CURRENT could return the assigned identity from a different row that was inserted between inserting the self inserted row and getting the identity for the self inserted row.

5.1.0.69

- ALL: Getting a binary value now respects the requested length if not all of the value is requested. Requesting less than the actual available data caused a buffer overrun. Problem was exposed by the Visual DataFlex debugger that requests the first 256 bytes of data from a binary column.

5.1.0.68

- MSSQLDRV: Finding on index that was based on a "unique identifier" data type would cause an error when the file buffer was empty or contains spaces. Field is now filled with the minimal GUID value.

5.1.0.67

- DB2_DRV: Introduced a new driver level attribute SKIP_IDENTITY_COLUMN_CHECK,. When set to true, it will no longer perform a check for the existence of an identity column when a table gets opened. It was reported by Marco Kuipers that this statement would be executed for each open statement. When running in a character mode environment where lots of FLX are 'CHAIN'ed, this may lead to an overhead.

5.1.0.66

- DB2_DRV: DF_Binary columns will now be created as "VARCHAR FOR BIT DATA" on DB2. Previously "CHARACTER FOR BIT DATA" was used. With this change DF_Binary columns will no longer be padded with spaces, but with binary zeroes.

5.1.0.65

- Find's on index segments with accented characters would sometimes fail if:
 - Ignore_Ucase_Support true
 - The table still having uppercased columns.

5.0.0.64

- DB2_DRV: The OPTIMIZE FOR 1 ROW option will no longer be added to embedded SQL select statements. Earlier change in build 5.0.0.63 was not correct.

5.0.0.63

- Changed default for Ignore_Ucase_Support to true.

Warning: This change may have consequences for existing databases/applications!!

The Ignore_Ucase_Support flag controls whether special uppercased columns (U_ columns) are created.

If Ignore_Ucase_Support is set to true (the new default), the Connectivity Kit will **not** create uppercased (U_) columns. Also, for existing tables having uppercase columns, the uppercased columns will be removed on restructure of the table.

The main reasons for changing the default to no longer create uppercase columns:

- The uppercase columns are only useful in some very specific circumstances. Especially: When a Case Insensitive collating sequence is used on the database backend, the uppercase columns are not needed.
- Use of uppercase columns has bad influence on performance.
- Uppercase columns can be confusing to users of external tools (e.g Crystal Reports).

Uppercase columns

Uppercased (U_) columns were introduced to be fully compatible with the DataFlex embedded database.

In the DataFlex embedded database, we know the concept of uppercased index segments. This is different from other (SQL based) databases where this concept does not exist.

In the DataFlex embedded database columns appearing in an index can be marked uppercased (or case insensitive) or not. The same column can be uppercased when it is used as a segment in one index and non-uppercased when it is used as a segment in another index.

In other databases case sensitivity is defined on the column. (Not on the index segment). It is defined by the used collating sequence that usually can be Case Sensitive or Case Insensitive.

Although the handling of case (in)sensitivity is not exactly the same in the DataFlex embedded database and SQL based databases, in most situations this does not require the existence of special uppercase (U_) columns.

Specifically, when the database backend uses a **case insensitive** collating sequence (the default on most backends), the uppercase columns are not needed. During find operations the same records will be found. It makes no difference if these finds are based on the original columns or on the uppercased (U_) columns.

When using a **case sensitive** collating sequence on the backend, not having U_ columns may cause differences in behavior for existing applications. This will only be the case if the Ignore_Case flag for an index segment is on. (Df_Index_Segment_Case = Df_Case_Ignore).

Find operation may find the record with U_ columns, but not find it without U_ Columns:

Example: If a table has a row with 'AA':

Find eq 'aa'

will find the row if table has U_ columns, but will not find it if the table has no U_ columns.

Note this behavior difference may also influence relates and constraints.

Consequences for existing databases

The change to no longer generate uppercased (U_) columns may have consequences for existing applications or databases.

If you have an existing database that has tables with uppercased (U_) columns, and you are upgrading from an earlier Connectivity Kit version: be aware of following consequences:

- The Connectivity Kit will no longer create uppercase (U_) columns during conversion or restructure.
- If existing tables with U_ columns are restructured, the U_ columns will be removed.
- Existing tables still having U_ columns will behave as before in Open, Find and Save operations. The U_ columns will be used as before in the SQL statements generated by these operations.
- The Ignore_case setting (Df_Index_Segment_Case attribute) will always be off. The flag has no meaning anymore. Case sensitivity is determined by the used collating sequence of the backend.
- If tables are used by external tools, removal of U_ columns may require changes. For example: In Crystal Reports a Verify Database is necessary after removing columns.
- When already using Case Insensitive collating sequence on the database backend, removal of U_ Columns will not cause existing programs to behave different.
- When using Case Sensitive collating sequence, existing applications may behave different after removal of U_ columns.

Keep using uppercase columns

If you need to keep using uppercase columns, the Ignore_Ucase_Support can be set to false in the driver configuration file. Or the Df_Driver_Ignore_UCase_Support or Df_Database_Ignore_UCase_Support attributes can be set to false at runtime.

If Ignore_Ucase_Support is set to false, the Connectivity Kit will behave the same as earlier versions.

- The SQLConnect function will now set the transaction isolation level to Read_Uncommitted.

So far, SQLConnect used the default isolation level of the database backend. For SQL Server and DB2 the default is Read_Committed.

For table access the Connectivity Kit uses Read_Uncommitted transaction isolation level.

To get the same isolation level for table access and Embedded SQL, SQLConnect will now set the isolation level to Read_Uncommitted.

- DB2_DRV: The OPTIMIZE FOR 1 ROW option will no longer be added to embedded SQL select statements.
- DB2_DRV for Linux: After restructure with db2_drv for Linux, index.0 would have incorrect name. TABLE0 instead of TABLE00.

5.0.0.62

- Support for SQL Server 2008

The SQL Server Connectivity Kit will now use the SQL Server 2008 client ("SQL Server Native Client 10.0") if it is installed on the client machine.

The SQL Server Connectivity Kit checks which clients are installed on the client machine and uses the 'highest' available.

- "SQL Server Native Client 10.0" (SQL Server 2008 Client)
- "SQL Native Client" (SQL Server 2005 Client)
- "SQL Server" (SQL Server 2000 Client (MDAC))

The used client version can be queried by the new attribute `DF_DRIVER_SQLSERVER_CLIENT_VERSION`

This attribute can return following values:

```
//*** Possible DF_DRIVER_SQLSERVER_CLIENT_VERSION values
#REPLACE SQLSERVERUNKNOWNCLIENT 0
#REPLACE SQLSERVER2000CLIENT 8 // "SQL Server"
#REPLACE SQLSERVER2005CLIENT 9 // "SQL Native Client"
#REPLACE SQLSERVER2008CLIENT 10 // "SQL Server Native Client 10.0"
```

This attribute cannot be set.

The connectivity kit can be configured to raise an error if the client and server version do not match. (See `Match_Client_Server_Version`)

- The SQL Server Connectivity Kit supports the new date and time data types for SQL Server 2008.

SQL Server 2008 introduces new data types for storing date (`SQL_DATE`) and time (`SQL_TIME`) variables. SQL server versions before SQL Server 2008 did not have a `SQL_Date` type, but only a `SQL_Datetime` type.

When creating new columns the Connectivity Kit would map DataFlex `DF_Date` type columns as `SQL_Datetime` type.

This has been left unchanged. When creating new columns (during conversion or restructure)

`DF_Date` maps to `SQL_Datetime`
`DF_Datetime` maps to `SQL_Datetime`

When connecting to existing SQL Server tables with the Connect Wizard:

`SQL_Date` maps to `DF_Date`
`SQL_Datetime` maps to `DF_Datetime`

Note: The new SQL Server 2008 types will only be recognized if the "SQL Server Native Client 10.0" (SQL 2008 client) is used. When using an older client the new types will be returned as `SQL_Char`/`DF_Ascii` types.

- `Match_Client_Server_Version` (All cli connectivity kits)

This setting (`MATCH_CLIENT_SERVER_VERSION`) can be set in the connectivity kit configuration file (`mssqldriv.int`, `db2_drv.int`, `odbc_drv.int`) or through the `DF_DRIVER_MATCH_CLIENT_SERVER_VERSION` attribute.

If this setting is set to true, the Connectivity Kit will check if the version of the database client and server software match. If the client version is less than the server version, an error will be raised when connecting to the server.

If the setting is set to false, versions will not be checked.

The default is false.

5.0.0.61

- When using Embedded SQL (`SQLConnect` function) the `Silent_Login` setting was not used.
- `Set_Attribute` did not check if the Connectivity Kit had initialized itself. If the first call made to the Connectivity Kit was a `Set_Attribute` of a `DF_DRIVER_` or `DF_DATABASE_` attribute, this would cause an access violation.
- Fixed misspelled attributes in `cli.pkg`:

`DF_DRIVER_DEFAULT_TABLE_CHARCATER_FORMAT`
`DF_DRIVER_APPLICATION_CHARCATER_FORMAT`

5.0.0.60

- SQL Server Connectivity Kit: Descending index segments were not reported as descending when using MS SQL Server 2005 without any Service Pack. This was a side effect of a change in build 5.0.0.53. The Connectivity Kit will now use a workaround when the server version is before SQL Server 2005 SP2.
- Changed default value for DB2 datetime type to include time portion. Default is now '0001-01-01 00:00:00'

5.0.0.57

- The Connectivity Kit for DB2 for Linux raised an error when processing the DFCONNECTIONID keyword in the connectivity kit configuration file (db2_drv.int)

5.0.0.56

- Some of the DRV*_ attributes returned incorrect values when used with the Cli_Get_Driver_Attribute command.

5.0.0.55

- Changed DF_HIGH character for DB2.
- Get_Attribute did not check if the Connectivity Kit had initialized itself. If the first call made to the Connectivity Kit was an Get_Attribute, this would cause an access violation.

5.0.0.54

- Get_Attribute DF_FIELD_NATIVE_TYPE inside structure_start/structure_end caused unhandled exception.

5.0.0.53

- Multiple SQLExecute calls generating informational messages would cause a memory violation error.
- If the full path name of a cache file (.cch) exceeds 64 characters, the cache file would not be found.
- ODBC Connectivity Kit: When connecting to a FoxPro table a date type field would be mapped to a datetime type field. This caused error: "Bad attribute value. Only columns that have a native datetime type can be set."
- When a SQL Filter is active, a Find GE would in some situations not respect the SQL Filter.
- After a find on a multisegment index, followed by a restructure, the structure_end would raise a memory overwrite error.
- SQL Server Connectivity Kit: If an error occurred during "Copy records on backend" in a restructure operation, the restructure rollback operation that followed would cause a GPF in dbBuilder.
- When reading the connectivity kit configuration file, a line after a 'line with no spaces' would not be processed.

Added new error to error file DFERR003.DAT.
 DFODBCERR_INVALID_CONFIGURATION_VALUE /* 12339 */

Changed Connectivity Kit configuration files:
 (Added explanation and example for DFCONNECTIONID)
 MSSQLDRV.INT
 DB2_DRV.INT
 ODBC_DRV.INT

- When changing the column name of a DB2 table in dbBuilder, the column name would be uppercased.
- Added support for setting nullable and default values for DateTime type columns:

Added Connectivity Kit configuration file keywords:

DEFAULT_NULLABLE_DATETIME
 DEFAULT_DEFAULT_DATETIME

Added new attributes:

DF_DATABASE_DEFAULT_DEFAULT_DATETIME
 DF_DATABASE_DEFAULT_NULLABLE_DATETIME
 DF_DRIVER_DEFAULT_DEFAULT_DATETIME
 DF_DRIVER_DEFAULT_NULLABLE_DATETIME

Added new attributes to Cli.Pkg

Changed Connectivity Kit configuration files
 MSSQLDRV.INT
 DB2_DRV.INT
 ODBC_DRV.INT

- After conversion the ascending/descending flag of index segments was sometimes set incorrect. This could happen if the name of one segment was a substring of another segment.

Example:

VENDORID (DESC), ID (ASC)

In this case the ID segment would be set to descending, since ID is a substring of VENDORID.

5.0.0.52

- Using an Express Connectivity license and connecting to DB2 Express-C Edition 9.5 or later would create a "Not an Express edition" error. This was caused by a change in the product identifier in DB2 Express-C 9.5.
- Set beta expiration date for db2_drv for Linux to September 1st 2008.

5.0.0.51

- A large DF_OPEN_PATH could cause a memory overwrite. This could sometimes cause a crash in the Studio. This happened when many libraries and/or long workspaces paths were used.
- ODBC Connectivity Kit: After conversion to Microsoft Access, length/precision of columns with type 'currency' would be incorrect. For example: A DataFlex Num 6.2 would be Num 4.4 on Access after conversion. This was caused by Access always using 4 decimals for currency type.

5.0.0.50

- SQL Server Connectivity Kit: When connected to SQL Server 2005, locking did not work properly. In some specific situations this could lead to deadlock errors or other locking related problems.

5.0.0.48

- Changing the find direction when finding inside a transaction would in some situations find the wrong record. For example:

```
Begin_Transaction
Find Ge
Find Lt
End_Transaction
```

Find Lt would not change the find direction and act like a Find Gt.

5.0.0.47

- Updating the single record of a system file would create an additional record in the system file in the following situation:

```
Move 1 To Sysfile.Recnum
Saverecord Sysfile
```

If Sysfile.Recnum already contained the value 1, the file_status would be set to inactive, causing the save to generate a new record, instead of updating the existing record. This bug was found in DF32 console mode dfBrowse program, where editing a system file record would always result in writing an extra record to the system file.

- The following code:

```
Clear Sysfile
Find Ge Sysfile By Recnum
```

would not find the single record of a system file

- Finding on non-unique indexes would not find all (duplicate) records.
- When getting the value of a datetime field, and using the AM/PM notation for the time part, the returned AM/PM value would sometimes be wrong. This was caused by a CK internal datetime to string conversion, where the AM/PM part would be truncated off the resulting string.

5.0.0.46

- Set beta expiration date for db2_drv for linux to June 1st 2008.

5.0.0.45

- Build a workaround for Firebird ODBC driver from Easysoft. (Uppercase segment handling)

5.0.0.44

- Changed user counting logic for use under Windows Vista.

5.0.0.43

- Changed the logic to determine if JIT binding should be used. JIT binding will now only be used if the native length of a column is larger than the JIT threshold.

5.0.0.42

- Getting the value of a JIT binded column would fail in the following situation:
 - Standard table
 - JIT column (text or binary field) present.
 - The primary index has one or more uppercased segments.

5.0.0.41

- Made some changes to let the odbc connectivity kit function with Firebird database.

5.0.0.40

- When using SQL Server 2000 the following errors could sometimes be generated during a save:
 - HY000 Connection is busy with results for another hstmt**
This error is caused by a known bug in SQL Server 2000.
A workaround has been implemented to avoid situations where this error can occur.
- After restructure, the field length for numeric and date fields was sometimes not right. This was caused by not writing FIELD_LENGTH to the <table>.int file.
- When using embedded SQL the installed MS SQLServer client(s) were not properly determined.

5.0.0.39

- Fixed a problem with SQLGetdata. An error was generated when executing SQLGetData on a column of type Varchar(Max). The Varchar(Max) type has an unlimited length. This unlimited length is returned as length = 0 by SQL Server. This could not be handled by the CK.
- DB2 Connectivity Kit - Turned on OPTIMIZEFORNROW option.
- Added support for CK Bundle license.
- DB2 Connectivity Kit (both Windows and Linux) - Find First_Record on an index with a date segment would cause a DB2 "invalid datetime syntax" error.
- A SQL Server 2005 column of type varchar(max), was presented as a DF_ASCII field, instead of a DF_TEXT field. Changed handling of varchar(max) and varchar(binary) types. These types were new in SQL Server 2005.

5.0.0.34

- Beta expiration date set to December 31st, 2007.

5.0.0.33

- Minor changes to Mssqldriv.pkg, Db2_drv.pkg and Odbc_drv.pkg - Some REPLACE lines had a comma at the end. Commas removed. Added comment line "Last updated: May 2, 2007"
- When accessing a Pervasive.SQL database through ODBC, deleting or updating existing records would generate errors.
- A Find Last_Record on index with last segment is recnum and constrain on first segment would not find a record.
- A Find Last_Record on an index containing a segment of type date, would give an status 16 error (Please enter a valid date). This happened only on VDF12.0 or earlier runtime.
- The following code:


```
Move "" To Table.Column
Save Table
```

 If Table.Column uses JIT Binding (e.g. a text field) , and the column already contained data, the empty string would not be saved. The column would keep the old value after a save.
- Opening a system table with a text field caused GPF. This would happen if the text field used JIT binding.
- After adding a new TEXT field, the length would not be written to the INT file. When opening the table after adding the TEXT field, an incorrect length (the default 16383) would be assumed
- Moving a string with a DateTime value to a DateTime field, for example - Move "31/12/2006 11:11:11.111" to Orderhea.Order_Date - would not save the milliseconds, or would cause "Invalid character value for cast specification" error on save. Type of error depends on the regional setting for the decimal separator and/or date format.
- When a DF_Collate.Cfg (other than English) was used by the runtime, the highest collating character was not determined correct. This caused a Find Last_Record to find a wrong record (or no record at all)

5.0.0.32

- See the What's New in Version 5 document for all changes since the 4.1 Connectivity Kits.

4.1.0.30

- Converting system files would crash Database Builder.
- The ODBC Connectivity Kit was not picking up the max_active_statement configuration set in the ODBC_DRV.int for ESQL connections.
- Fixed "Invalid SQL statement identifier" error when using MyOB driver.
In order to establish if relationships exist, the ODBC function SQLForeignKeys is called. This function was causing problems. The ODBC CK was adjusted so it no longer calls this function when connecting to an OpenRDA database.
- When the CK tried to establish if relationships existed, the MyOB ODBC driver would return an error. The ODBC Connectivity Kit now works around this problem.

4.1.0.21

- In some cases moving a value to a column that was already there was considered a change, fixed.
- Fixed a bug in the dummy update logic.
- Fixed DB2 lock issues.
- Changed RowId logic to support relative fetching.
- Fixed a problem in getting index definition from FoxPro database.
- Fixed a number of ANSI/OEM issues with table and index names.
- Added new configuration keyword USEDLOCKERROR. When set the error number generated for a deadlock or timeout will be the DataFlex time out error (4106) instead of the CK error. This will enable the automatic retry mechanism build into DataFlex.
- In SQL server binary data was filled out with binary zeroes. The CK would return the data in this way leading to misinterpretations of binary data. Trailing binary zeroes are now truncated.
- Fixed a problem in getting primary key information. In DBIsam it is possible to define a primary key column that is not part of the table definition (record id is an internal column). The CK could not handle that, it assumed the column in the primary key would be visible.

4.1.0.17

- Fixed a bug in moving a value to a field that was already in that field. This should not be considered a change, but sometimes it was.
- The FindByRowID logic for a standard table now also uses relative fetching. This makes it faster when doing repeated find equal operations on the same record for the primary index. This happens when finding parent records for example.
- Fixed a problem with cursor names in the DB2 CK when saving a record.

4.1.0.14

- Fixed a number of problems with converting to MySQL using the ODBC driver.

4.1.0.11

- Fixed a problem when converting system tables back to DataFlex.

4.1.0.8

- This was the build used in the release of Visual DataFlex 11.1.
- SQL Server CK could generate a "Connection is busy with result for another hstmt" error when creating a new record. Fixed.

4.1.0.7

- Changed default of DF_FILE_RECNUM_TABLE to true for new tables. This will fix the use of a 4.1 CK in pre-VDF11 versions.
- Converting a table with a column that has length 0 to SQL Server would hang, fixed.

4.1.0.5

- Finding on uppercased index segments of Oracle tables would not work correctly when the segment contains an empty string. Fixed.

- Ending a transaction can cause the database server to close open cursors, this happens for a normal transaction end (commit) and an abnormal transaction end (rollback). The rollback logic was not taking all possible behaviors into account. Fixed.
- Reusing a connection global statement handle for updating records caused locks to stay on longer then needed. Fixed.
- Locking a record in MS Access would cause "resource errors" or simply crash. Fixed.
- When using an ODBC data source and the Use_Identity_Type database configuration file keywaord was set, the recnum column would not have the identity attribute. Fixed.

4.1.0.1

- Improved error reporting if recreation of indexes fails.
- Setting a default on a column that did not have a default yet would crash, fixed.
- Added support for full (negative) range of numeric columns.
- Added support for System tables with no indices.