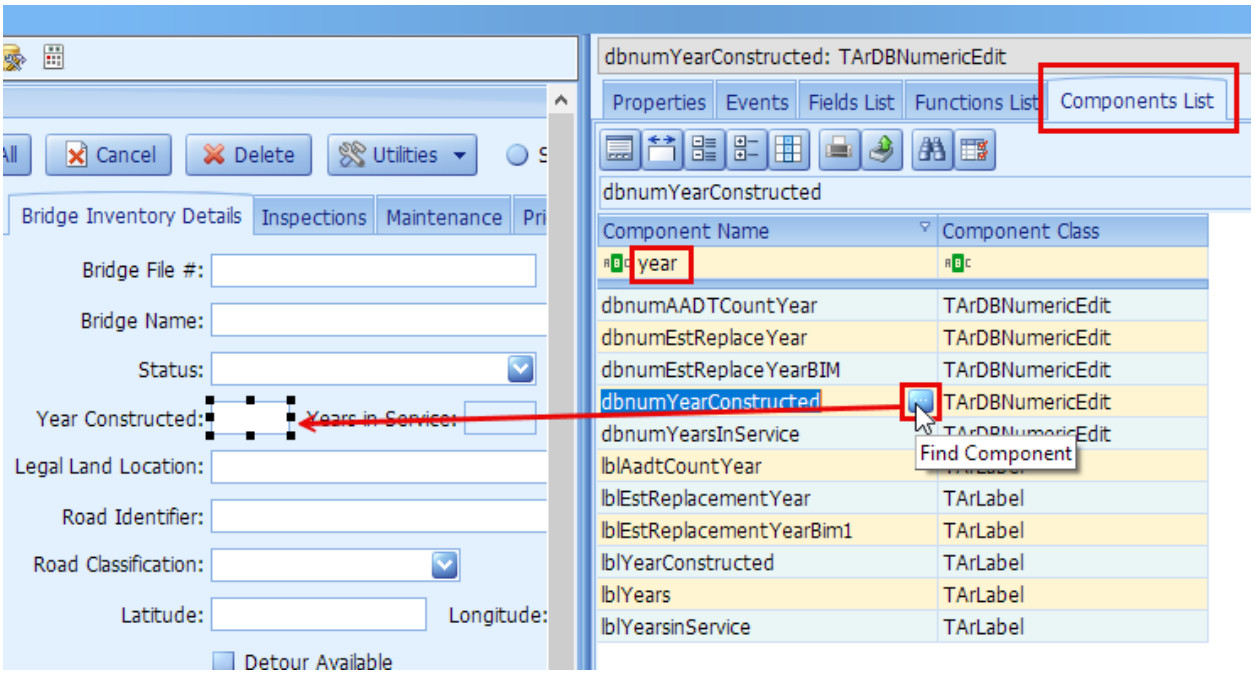
Release Notes for Data Manager and Data Manager Designer

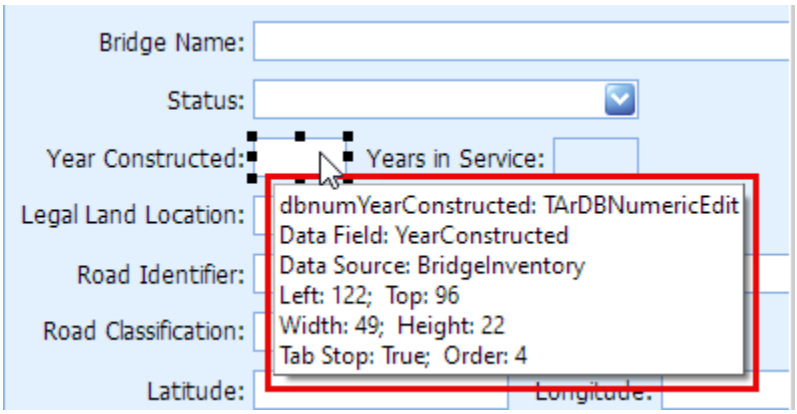
4.6 build 3

1. **New Feature:** In the Form Designer, we added the "Find Component" button in the "Component Name" column on the "Components List" tab to find and select the component on the form.

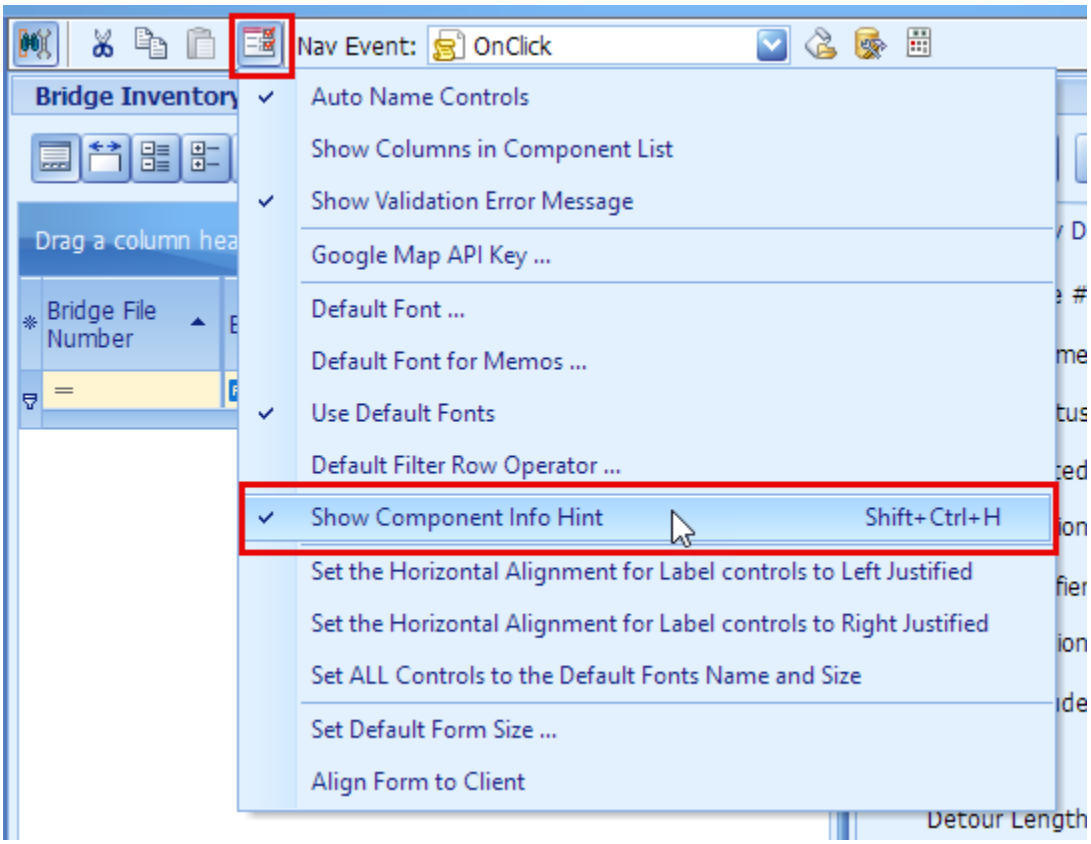
   Tip: You can filter the component list just like any other grid. This powerful feature helps you narrow down your search and find exactly what you're looking for. In the example below, the Component Name column is filtered to show anything that contains the word "year", reducing the list from 481 components to just 11. This demonstrates the effectiveness of this tool in helping you find the right component with ease.



2. **New Feature:** We have added a hint when hovering over the components in the Form Designer displaying pertinent information like the component's name, data field, and data source, as well as size and location in pixels. We call this the "Component Info Hint".



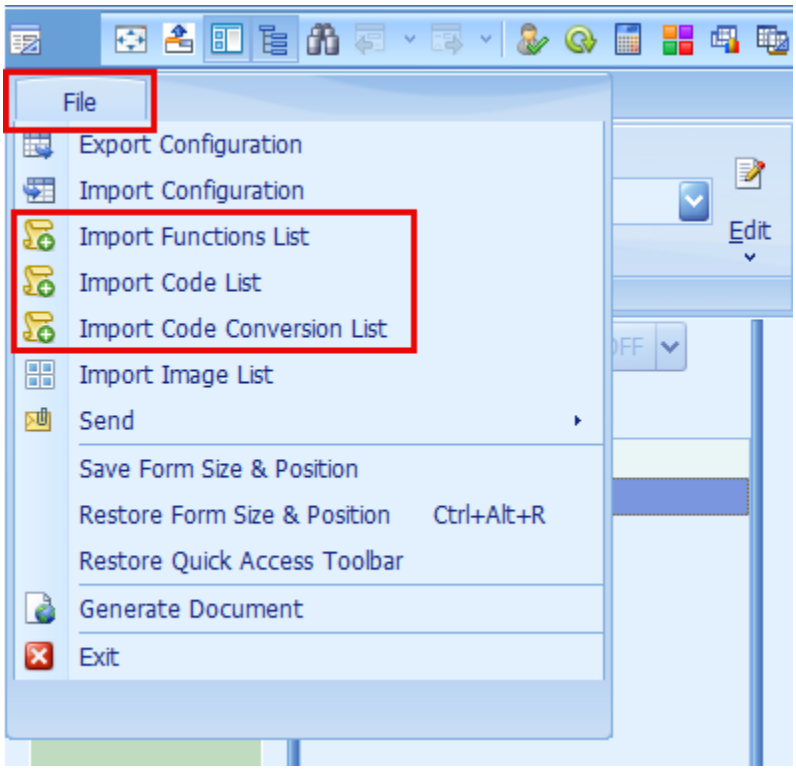
To turn on or off the hints, click the Options button, then click "Show Component Info Hint" or press Shift+Ctrl+H on your keyboard.



4.6 build 2

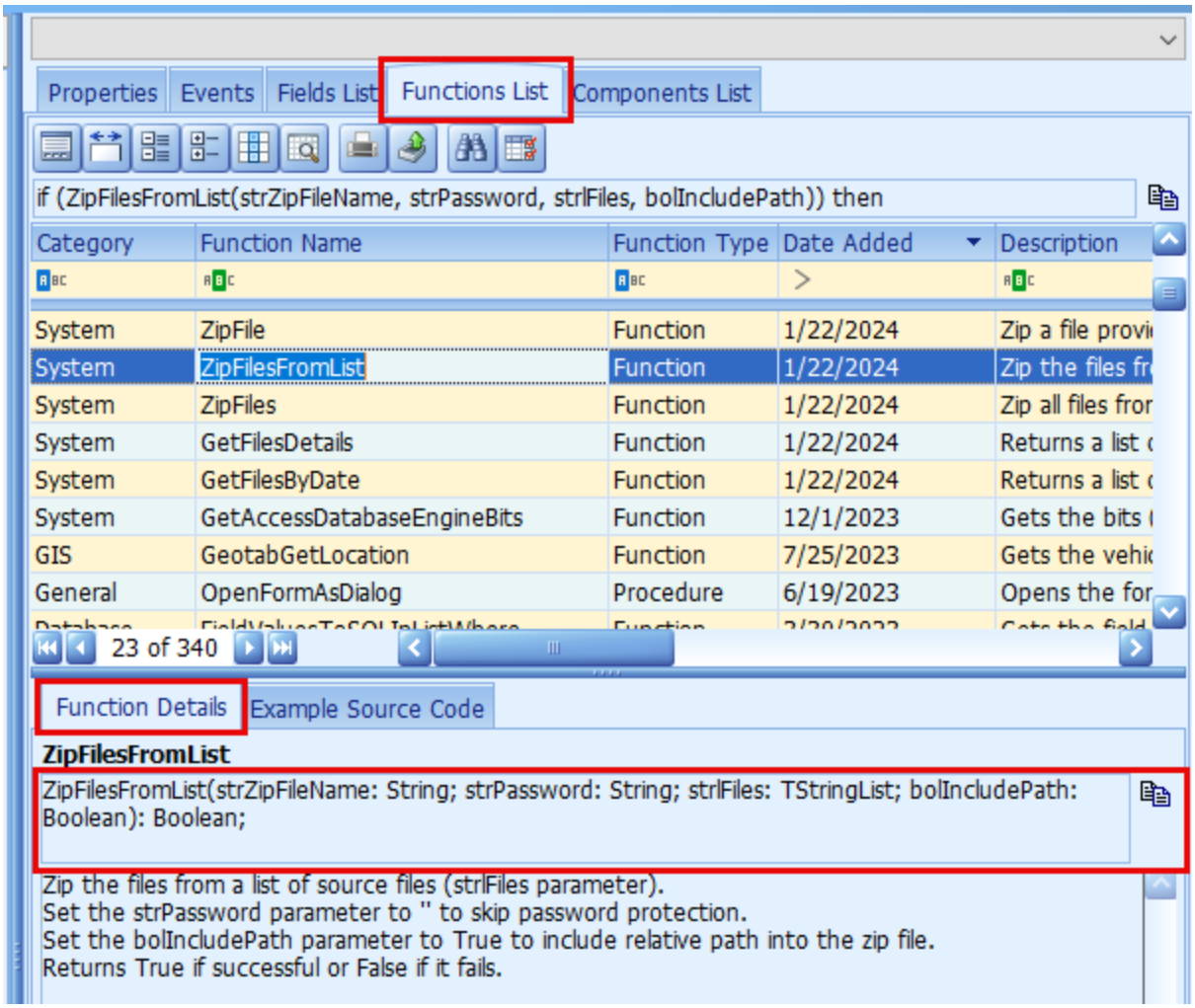
1. **Major Enhancement:** Added support for Microsoft SQL Server 2022.
   The supported versions of SQL Server are:
   - SQL Server 2022 ← This is the latest official release version as of January 2024.
   - SQL Server 2019
   - SQL Server 2017
   - SQL Server 2016
   - SQL Server 2014
   - SQL Server 2012

2. **New Feature:** Added the following import features to the File tab of Data Manager Designer. As we improve these lists (with better descriptions and examples), we will be able to send them to users of Data Manager Designer to import so the updated lists are available without the need to upgrade the software.

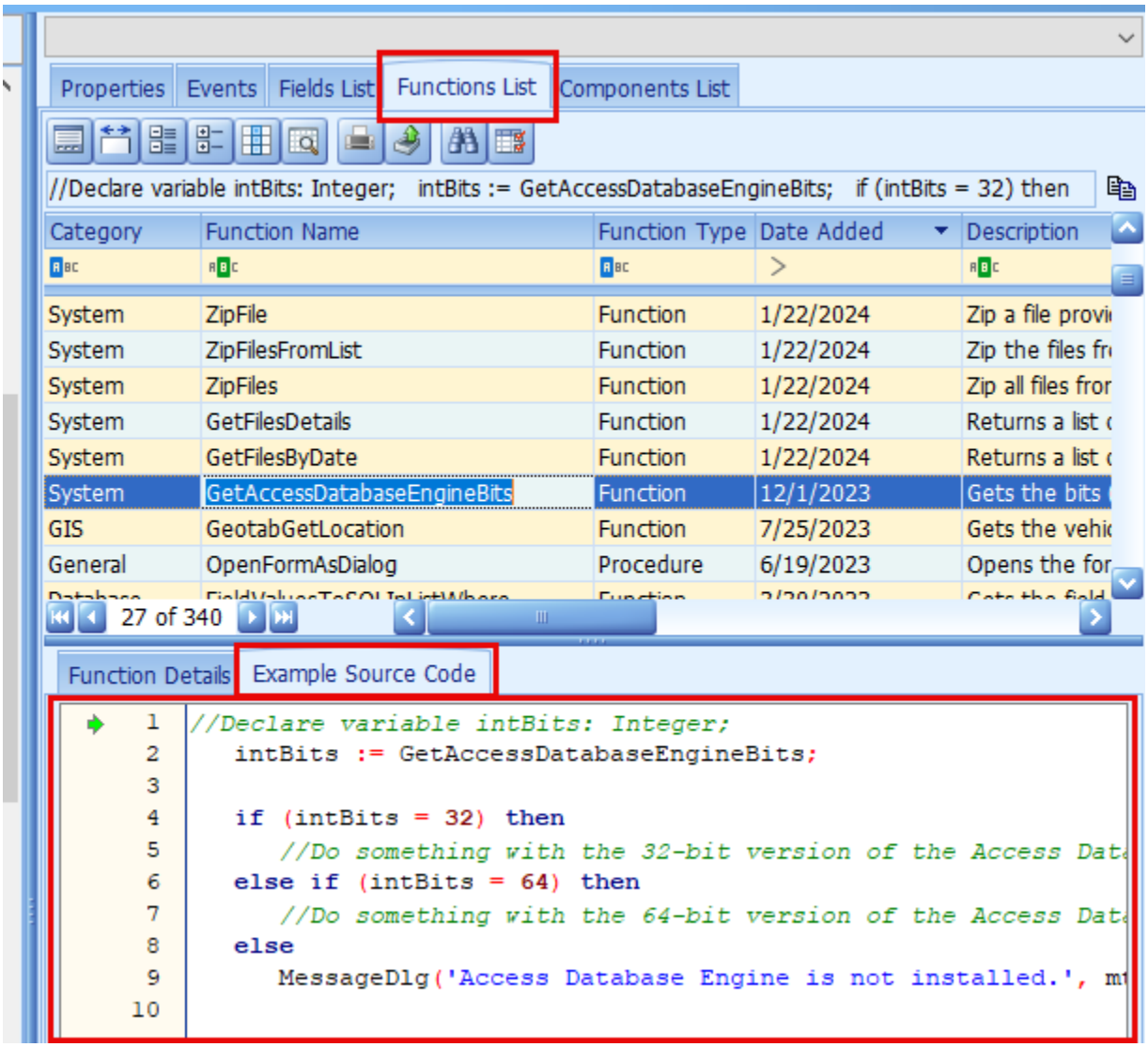


3. **New Feature:** The Functions List tab in the Form Designer now has the “Function Details” and “Example Source Code” tabs.

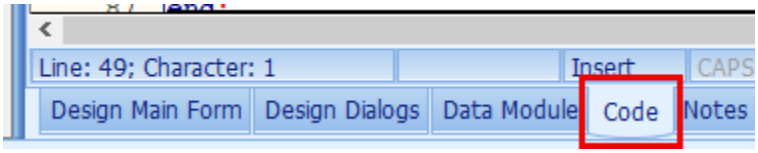
   Added a memo box on the “Function Details” tab that shows the Function Declaration field. This is helpful information when scripting.



The “Example Source Code” tab shows an example of how to use the selected function in scripting.

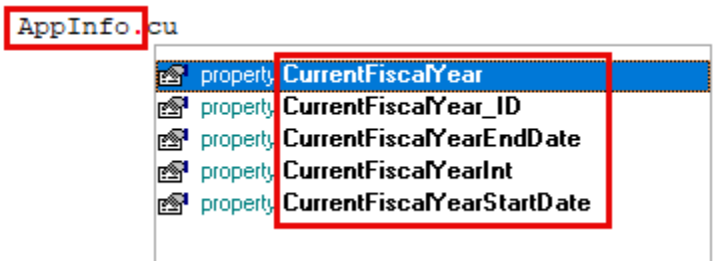


Note: The “Function Details” and “Example Source Code” tabs are only enabled when the “Code” tab is active. This is to avoid confusion when the Design tabs are active and the user uses the Ctrl+C hotkey to copy something.



4. **New Feature:** Added the following properties to the “AppInfo” object in scripting.
   This information is queried from the tblFiscalYears table in the System database.
   The CurrentFiscalYear… properties are based on the fiscal year record where the current system date is within the start and end date.
   For example, if the current system date is February 28, 2024, then based on the example data below, the CurrentFiscalYear would be “2023_24” and the CurrentFiscalYearInt would be 2023.

   AppInfo.CurrentFiscalYear – contains the value of the FiscalYear field.
   AppInfo.CurrentFiscalYear_ID – contains the value of the FiscalYear_ID field.
   AppInfo.CurrentFiscalYearEndDate – contains the value of the EndDate field.
   AppInfo.CurrentFiscalYearInt – contains the value of the FiscalYearInt field.
   AppInfo.CurrentFiscalYearStartDate – contains the value of the StartDate field.



   An example of the contents of the tblFiscalYears table in the System database for a client whose fiscal year starts on April 1 each year.

FiscalYear_ID	FiscalYear	FiscalYearInt	StartDate	EndDate
10000	2021_22	2021	2021-04-01	2022-03-31
20000	2022_23	2022	2022-04-01	2023-03-31
30000	2023_24	2023	2023-04-01	2024-03-31
40000	2024_25	2024	2024-04-01	2025-03-31
50000	2025_26	2025	2025-04-01	2026-03-31
60000	2026_27	2026	2026-04-01	2027-03-31
70000	2027_28	2027	2027-04-01	2028-03-31

5. **New Functions:** Added the following functions in scripting:

Function Name	Description	Date Added
ConvertCentimetersToFeetInches	Converts a distance in centimeters to feet and inches. Returns distance in feet and inches as a string.	1/30/2024
ConvertCentimetersToFeetInchesExt	Converts a distance in centimeters to feet and inches, like the ConvertCentimetersToFeetInches() function. However, this function also returns the Feet and Inches as variable parameters, so they can also be stored separately if needed. Returns distance in feet and inches as a string.	1/30/2024
ConvertFeetInchesToCentimeters	Converts a distance in feet and inches to centimeters. Returns distance in centimeters.	1/30/2024
ConvertFeetInchesToMeters	Converts a distance in feet and inches to meters. Returns distance in meters.	1/30/2024
ConvertMetersToFeetInches	Converts a distance in meters to feet and inches. Returns distance in feet and inches as a string.	1/30/2024
ConvertMetersToFeetInchesExt	Converts a distance in meters to feet and inches, like the ConvertMetersToFeetInches() function. However, this function also returns the Feet and Inches as variable parameters, so they can also be stored separately if needed. Returns distance in feet and inches as a string.	1/30/2024
DistCm2Feet	Converts a distance in centimeters to feet. Returns distance in feet.	1/31/2024
DistFeet2Cm	Converts a distance in feet to centimeters. Returns distance in centimeters.	1/31/2024
DistFeet2M	Converts a distance in feet to meters. Returns distance in meters.	1/31/2024
DistKm2Miles	Converts a distance in kilometers to statute miles. Returns distance in statute miles.	1/31/2024
DistM2Feet	Converts a distance in meters to feet. Returns distance in feet.	1/31/2024
DistMiles2Km	Converts a distance in statute miles to kilometers. Returns distance in kilometers.	1/31/2024
GetDefinedSetting	Gets the value of the given defined setting. This is not user-specific. If you want to get a user-specific setting, use the GetUserDefinedSetting() function instead.	2/6/2024
GetDefinedSettingDate	Gets the date value of the given defined setting. This is not user-specific. If you want to get a user-specific date setting, use the GetUserDefinedSettingDate() function instead.	2/6/2024

GetFilesByDate	Returns a list of files in a given directory by the Date Modified date range. Possible soSearchOption are: SearchOption_soAllDirectories SearchOption_soTopDirectoryOnly Returns True if successful or False if it fails.	1/22/2024
GetFilesDetails	Returns a list of files with details in a given directory and populates into the memory dataset (dsMemDataFilesList). The memory dataset (dsMemDataFilesList) contains the fields: Path, Name, DateCreated, DateModified, Size, Type Possible soSearchOption are: SearchOption_soAllDirectories SearchOption_soTopDirectoryOnly Returns True if successful or False if it fails.	1/22/2024
GetFilesDetailsByDate	Returns a list of files with details in a given directory by the Date Modified date range, which is then populated into the memory dataset (dsMemDataFilesList). The memory dataset (dsMemDataFilesList) contains the fields: Path, Name, DateCreated, DateModified, Size, Type Possible soSearchOption are: SearchOption_soAllDirectories SearchOption_soTopDirectoryOnly Returns True if successful or False if it fails.	1/25/2024
IsItemInString	Determines if any one or all items in a delimited list are contained in the given string. If bolCaseSensitive is set to True, then the comparison is case-sensitive. Otherwise, it is not case-sensitive. If bolContainsAll is set to True, this function only returns True if all items are in the string. Otherwise, it returns False.	2/6/2024
LatLongInSaskatchewan	Determines if the given Latitude and Longitude coordinates are within the province of Saskatchewan. Returns True if the coordinates are within the province of Saskatchewan or False if they are not.	1/31/2024
MassKg2Lbs	Converts a mass in kilograms to pounds. Returns mass in pounds.	2/1/2024
MassLbs2Kg	Converts a mass in pounds to kilograms. Returns mass in kilograms.	2/1/2024
MassShortTons2Tonne	Converts a mass in short tons to tonnes. Short tons are also known as US tons, and tonnes are also known as metric tons. Returns mass in tonnes.	2/1/2024

MassTonne2ShortTons	Converts a mass in tonnes to short tons. Short tons are also known as US tons, and tons are also known as metric tons. Returns mass in US tons (short tons).	2/1/2024
SaveDefinedSetting	Saves any given setting for the given setting key. This is not user-specific. If you want to save a user-specific setting, use the SaveUserDefinedSetting() function instead. Use GetDefinedSetting(strSettingKey, strDefaultValue: String); to retrieve the setting.	2/6/2024
SaveDefinedSettingDate	Saves any given date setting for the given setting key. This is not user-specific. If you want to save a user-specific date setting, use the SaveUserDefinedSettingDate() function instead. Use GetDefinedSettingDate(strSettingKey: String; dtDefaultValue: TDateTime); to retrieve the setting as a DateTime value.	2/6/2024
SplitStringToList	Splits a string containing a given delimiter into items in a string list. Returns a StringList object.	2/6/2024
VolL2USGallons	Converts a volume in liters to US gallons. Returns volume in US gallons.	2/1/2024
VolUSGallons2L	Converts a volume in US gallons to liters. Returns volume in liters.	2/1/2024
ZipFile	Zip a file provided by the strSourceFile parameter. Be sure to include the path of the source file, e.g 'D:\Temp\Test.docx' Set the strPassword parameter to '' to skip password protection. Set the bolIncludePath parameter to True to include relative path in the zip file. Returns True if successful or False if it fails.	1/22/2024
ZipFiles	Zip all files from a directory (strSourcePath parameter), including sub-directories. Set the strPassword parameter to '' to skip password protection. Set the bolIncludePath parameter to True to include relative path in the zip file. Returns True if successful or False if it fails.	1/22/2024
ZipFilesByDate	Zip all files from a directory (strSourcePath parameter) by the Date Modified date range, including sub-directories. Set the strPassword parameter to '' to skip password protection. Set the bolIncludePath parameter to True to include the relative path in the zip file. Returns True if successful or False if it fails.	1/25/2024
ZipFilesFromList	Zip the files from a list of source files (strlFiles parameter). Set the strPassword parameter to '' to skip password protection. Set the bolIncludePath parameter to True to include relative path in the zip file. Returns True if successful or False if it fails.	1/22/2024

4.6 build 1

1. **Enhancement:** In the Form Designer, we improved error handling when a control's property value is manually cleared. Most properties of the controls cannot be blank, so this error handling protects against that mistake.

2. **Issue:** In the 64-bit version of Data Manager, if a query or SQL View is linked to a table as the master, an error occurs when appending a record if the table is empty.
   Status: This issue has been resolved.

4.6 build 0

1. **Major Enhancement:** Added support for Data Manager and Data Manager Designer to run as a 64-bit application in Win64 which helps eliminate any "Out of Memory" errors that may occur. In 32-bit applications, the dynamically allocated memory is restricted to 2 Gigabytes and in 64-bit applications, the restriction is 8 Terabytes (only limited to the available RAM). This is particularly important when importing very large files.

2. **Major New Feature:** Added the following new procedure:

OpenFormAsDialog (strNodeGUID: String; bolIsDialogSizeable: Boolean; intDialogWidth: Integer; intDialogHeight: Integer; strExtraParams: String).

This procedure opens the form as a modal dialog for a form which is defined with the "System Node" node style. When the parameter "bolIsDialogSizeable" is set to True and the parameter "intDialogWidth" or "intDialogHeight" is set to zero, it will initially maximize the dialog. The optional parameter "strExtraParams" can be used for passing the parameters to the dialog for special operations such as filtering the dialog data. Set the parameter "strExtraParams" in 'Key=Value' format with each parameter separated by a semicolon.
For example:
strExtraParams := 'ID=123456;Code=ABC'.

In scripting, the extra parameters can be set or accessed using the DialogMemList object.
For example:
intID := DialogMemList.GetInt64('ID', 0);
strCode := DialogMemList.GetString('Code', '');
DialogMemList.SetString('Status', 'OK');

Usage Example:

```
OpenFormAsDialog('{3A6EC33D-4A05-4ECB-AFB1-5C41BC20E65B}', True, 900, 550);
```

Or

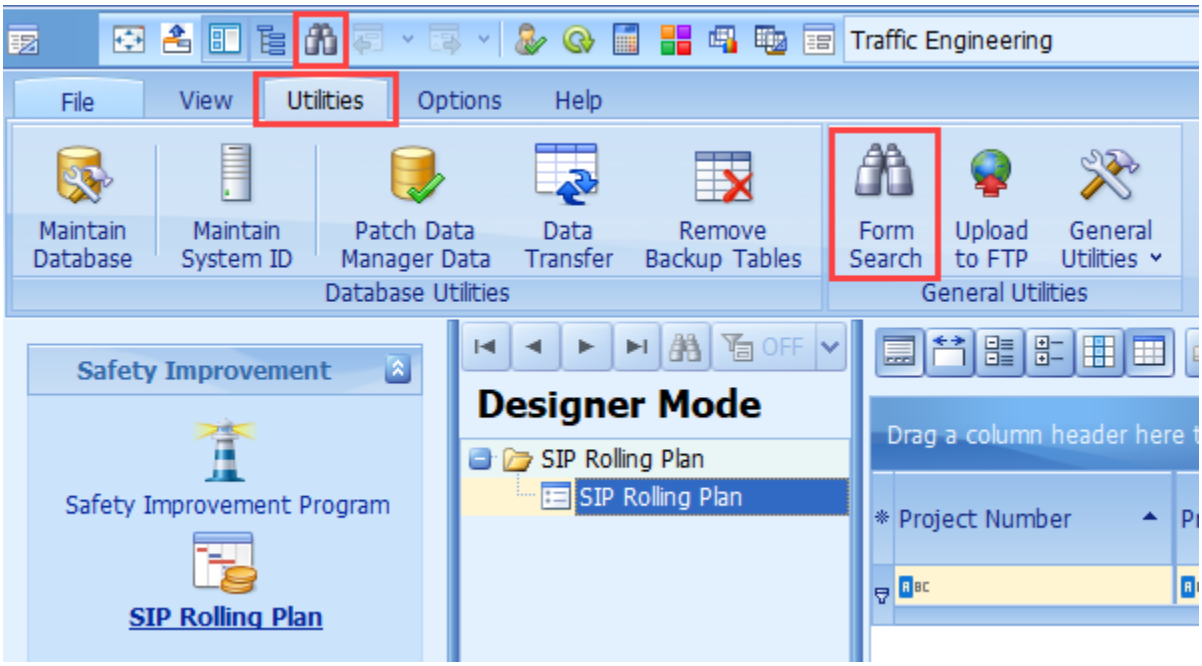
```
strExtraParams := 'ID=' + tblTypes.FieldByName(ID').AsString +  ';Code=' +
tblTypes.FieldByName('Code').AsString;

OpenFormAsDialog('{3A6EC33D-4A05-4ECB-AFB1-5C41BC20E65B}', True, 900, 550,
strExtraParams);
```

Filtering Data Example in OnBeforeOpenTables event for the Form being shown as Dialog:

```
procedure OnBeforeOpenTables;
begin
  if (AppInfo.IsFormAsDialog) then
  begin
    tblData.FilterSQL := 'ID = ' + DialogMemList.GetString('ID', '0');
    tblData.Filtered := True;
  end;
end;
```

3. **Major New Feature:** In Data Manager Designer, we added the ability to search for any text in any form in the entire Data Manager database. This is very handy in many situations. This is very handy in many situations such as looking for a form name, or the text of a message in scripting, or if you want to know what forms are using a particular function and how that function is implemented in the scripting.

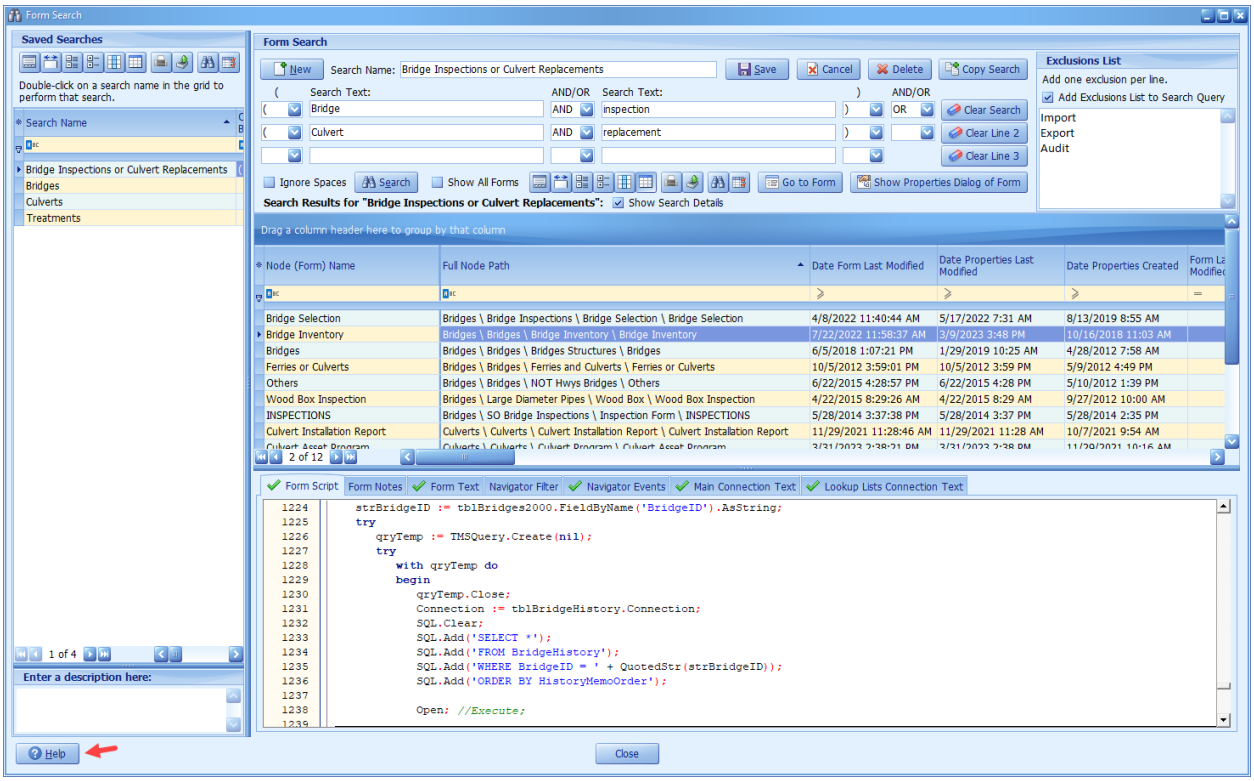
   Here are the various ways to open the Form Search dialog. You can also open this dialog with the **Ctrl-Shift-F** hotkey combination.



   The Form Search dialog can also be opened from inside the Form Designer using this button in the lower right of the Form Designer, as shown here:

   The Form Search dialog has a lot of helpful features. For more information regarding the Form Search, click the "Help" button in the lower left corner of this dialog, as shown below with the red arrow.

Form Search dialog:



4. **Enhancement:** Upgraded support for the latest Geotab © API. Note: using this feature now requires .NET 7 or higher to be installed.

5. **Enhancement:** Improved the GeotabImport function.
   Here is the revised declaration of this function:

```
function GeotabImport(cnData: TMSConnection; intGeotabConInfo_ID:
Int64; strOperation: String; strGeotabTableName: String;
strExtraParams: String): Int64;
```

6. **New Feature:** Add a new function called GeotabGetLocation to get the location of a single device (e.g., vehicle or equipment unit). The purpose of this function is to quickly and efficiently get the location of one unit rather than querying and loading the entire Geotab database. This is helpful for displaying a selected unit a map.

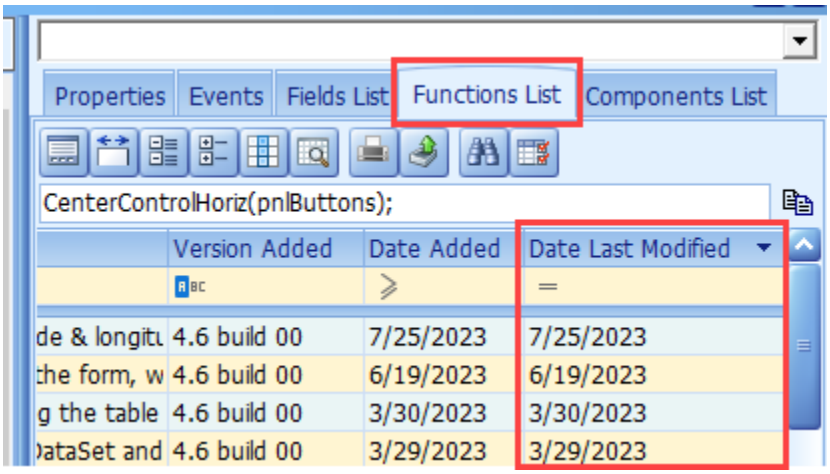
   Here is the declaration of this new function:

```
function GeotabGetLocation(cnData: TMSConnection; intGeotabConInfo_ID:
Int64; strGeotabTableName: String; strDeviceID: String; strExtraParams:
String; var extLatitude: Extended; var extLongitude: Extended): Int64;
```

7. **Enhancement:** Added support for the mouse wheel to scroll vertically in scroll boxes provided that the mouse pointer is not hovering over certain edit controls like combo boxes and memo edit boxes. Hover the mouse pointer over an open area in the scroll box to scroll vertically with the mouse wheel.

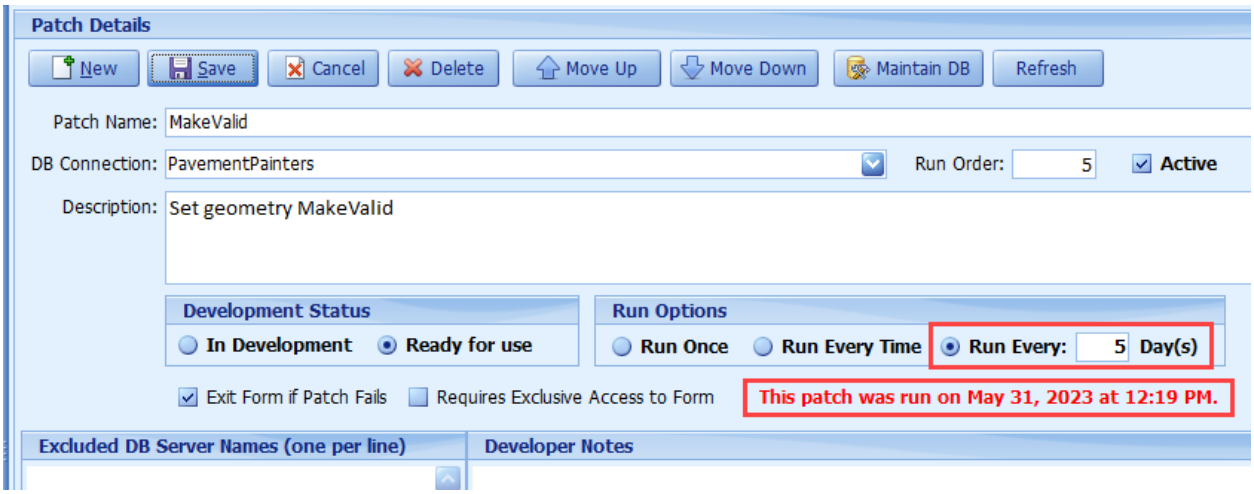
8. **Enhancement:** Added the "Date Last Modified" column to the Functions grid in the Form Designer. Quite often we improve the description or the default syntax of functions in this list

so this date will make it easier to see what function records were updated most recently.



9. **Enhancement:** In the Auto Patch feature in the Form Designer, we added a new Run Option called "Run Every X Day(s)" so the developer can specify the number of days elapse before a patch is run when a user loads a form regardless of how many users access this form. As in the example below, this patch will run every 5 days when the form is loaded. The range of days can be any number from 1 to 999. This option is helpful when update queries need to run more than once but not every time every user goes to the form.

   We also added the date and time to the label that shows when the patch was last run.



10. **Issue:** The Copy and Paste feature for the data source structure in the Main Connection and Lookup Lists Connection dialogs is not working for the Queries data sources.
    Status: This issue has been resolved.

11. **New Feature:** Added the following to the scripting engine:
    a. **AppBits** to the AppInfo object. i.e. AppInfo.AppBits to determine if Data Manager is 32-bit or 64-bit.
    b. **GetAccessDatabaseEngineBits** to determine if the version of the Microsoft Access Database Engine installed is 32-bit or 64-bit. If the function returns -1, the Access Database Engine is not installed.
    c. **RunImport32Bit** function in the ImportData component is to be used in the 64-bit version of Data Manager if the 32-bit version of the Access Database Engine is installed. Conversely, the **RunImport64Bit** function in the ImportData component is to be used in the 32-bit version of Data Manager if the 64-bit version of the Access Database Engine is installed. This enables both the 32-bit version and the 64-bit version of Data Manager to use both the 32-bit version and the 64-bit version of the Microsoft Access Database Engine.

Here is an example of these new functions in use.
Note: This example is for importing Microsoft Access databases only.
The "dbimpMyImport" object is the name of the ImportData component used in this example.

```
var
   intADEBits: Integer;
begin
   intADEBits := GetAccessDatabaseEngineBits;

   if (intADEBits = -1) then
   begin
      MessageDlg('The Microsoft Access Database Engine is not installed.' +
                  #13 +
                 'Please contact your configuration manager.', mtError,
                  ([mbOK]), 0);

      Exit;
   end;

   if (AppInfo.AppBits = 32) then
   begin //Data Manager is 32-bit
      if (intADEBits = 32) then
         //Import using the internal 32-bit version
         if dbimpMyImport.RunImport(False) then
            MessageDlg('Import complete.', mtInformation, SetOf([mbOk]), 0);
      else if (intADEBits = 64) then
         //Import using the external 64-bit exe version
         if dbimpMyImport.RunImport64Bit(False) then
            MessageDlg('Import complete.', mtInformation, SetOf([mbOk]), 0);
   end
   else //Data Manager is 64-bit
   begin
      if (intADEBits = 32) then
         //Import using the external 32-bit exe version
         if dbimpMyImport.RunImport32Bit(False) then
            MessageDlg('Import complete.', mtInformation, SetOf([mbOk]), 0);
      else if (intADEBits = 64) then
         //Import using the internal 64-bit version
         if dbimpMyImport.RunImport(False) then
            MessageDlg('Import complete.', mtInformation, SetOf([mbOk]), 0);
   end;
end;
```

4.5 build 7

1. **Enhancement:** Significantly improved the performance of cutting and pasting many controls at once on a form in the Form Designer.

2. **Enhancement:** Improved support of storing user data in virtual environments (like App-V, VMWare, etc.).

3. **Issue:** In rare situations, cutting and pasting the grid component on Form Designer may cause an error.
   Status: This issue has been resolved.

4.5 build 6

1. **New Feature:** Added the following new procedures for setting the child controls "Enabled" or "ReadOnly" property by tag range. These procedures are like the ones with the same name but without the word "Range". The only difference is these procedures make it possible to use a range of tag numbers so controls can be grouped with different tag numbers. The tag property for all controls can be used to store any integer value either in memory at runtime or stored with the form at design time. So, for example, if you want to use the tag property to contain a calendar year and you want to set the "ReadOnly" or "Enabled" property for all the controls with the tag property from 2018 to 2023, you can call one of these procedures once, instead of calling the previous procedure multiple times. This is more efficient in this case.

   SetChildControlsEnabledByTagRange(ctlParent: TWinControl; intMinTag: Integer; intMaxTag: Integer; bolEnabled: Boolean);

   SetChildControlsReadOnlyByTagRange(ctlParent: TWinControl; intMinTag: Integer; intMaxTag: Integer; bolReadOnly: Boolean);

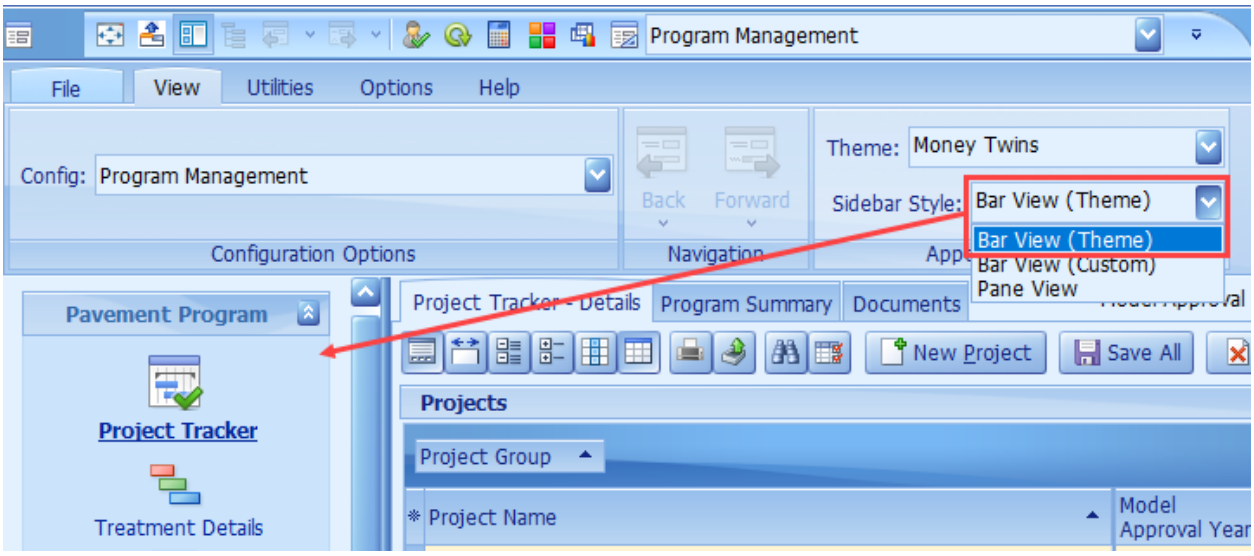
   SetColumnsReadOnlyByTagRange(grdGrid: TArDBGrid; intMinTag: Integer; intMaxTag: Integer; bolReadOnly: Boolean);

4.5 build 5

1. **Major Upgrade:** Upgraded the data connection components to support the latest version of the Microsoft OLE DB Driver for SQL Server and improved support for SQL Server 2019. This upgrade also includes fixing some issues for older versions of SQL Server.

2. **Enhancement:** Significantly improved the performance and usability of the Google Maps component.

3. **Enhancement:** Added a new Sidebar Style called “Bar View (Theme)” that makes the sidebar use the current theme. The name of the existing “Bar View” sidebar style is changed to “Bar View (Custom)” to represent custom colors for the groups in the sidebar instead of using the theme color.



4. **Enhancement:** Many cosmetic improvements throughout the application for various themes.

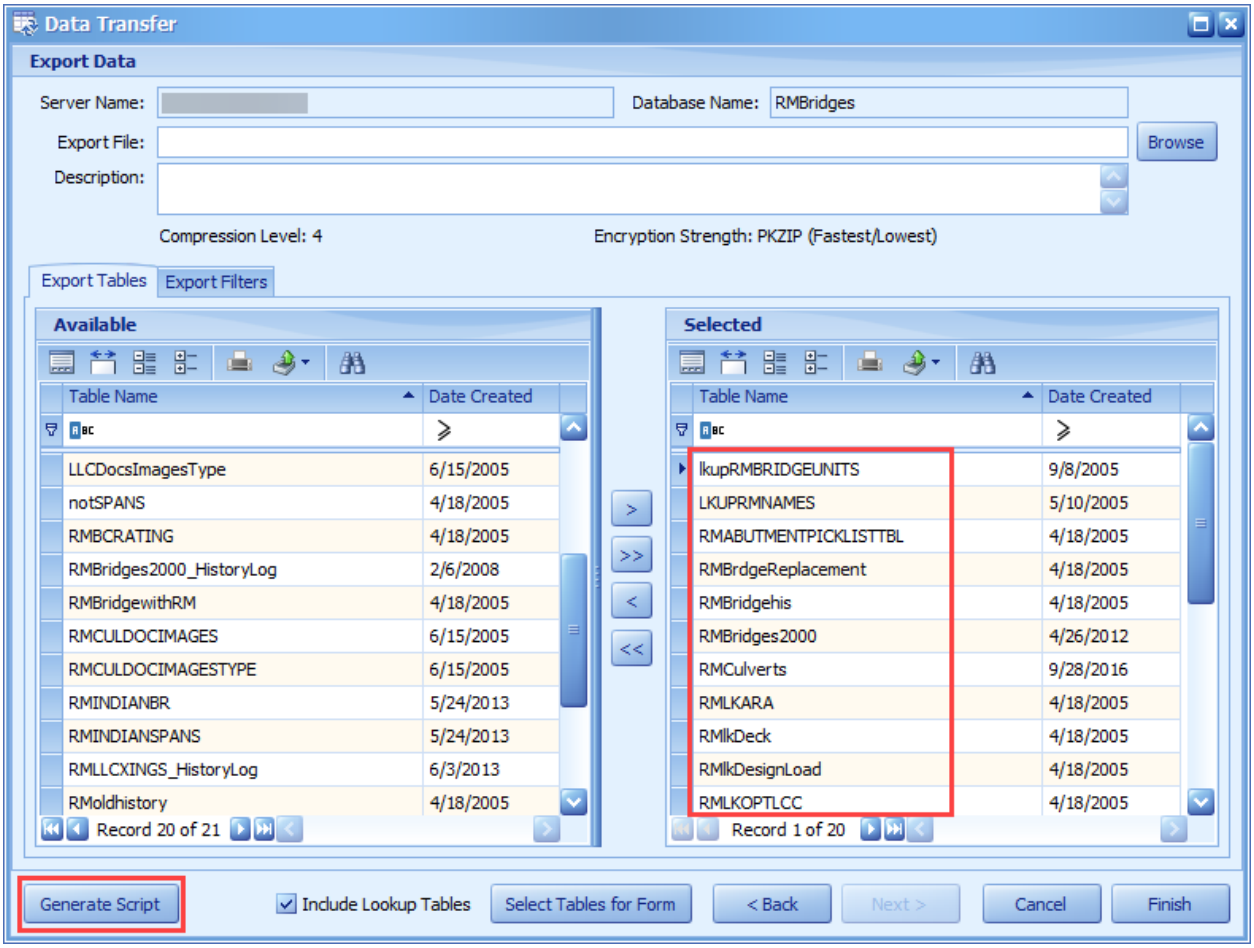
5. **Issue:** The Google Maps component is not drawing properly when multiple instances of Data Manager are running.
   Status: This issue has been resolved.

6. **Issue:** The custom markers on the Google Maps component are drawing slightly off the actual locations depending on the zoom level.
   Status: This issue has been resolved.

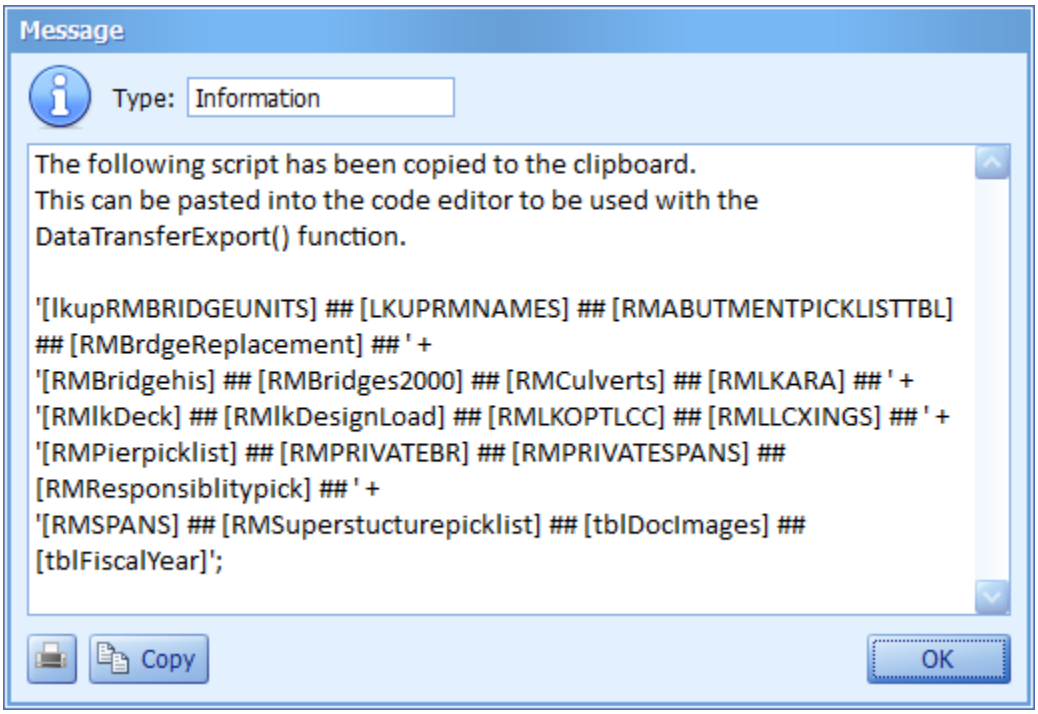
7. **Issue:** In certain situations, the Form Designer intermittently expands the form area width when one of the controls on the form is being selected.
   Status: This issue has been resolved.

4.5 build 4

1. **New Feature:** Added the Generate Script button to the Data Transfer Export dialog. This button will generate the script needed to export the tables that are in the Selected list. Then it copies that script to the clipboard, so it can be pasted into a procedure like the one shown on the following page to be used in the DataTransferExport() function in scripting.



After pressing the Generate Script button, this message will appear:

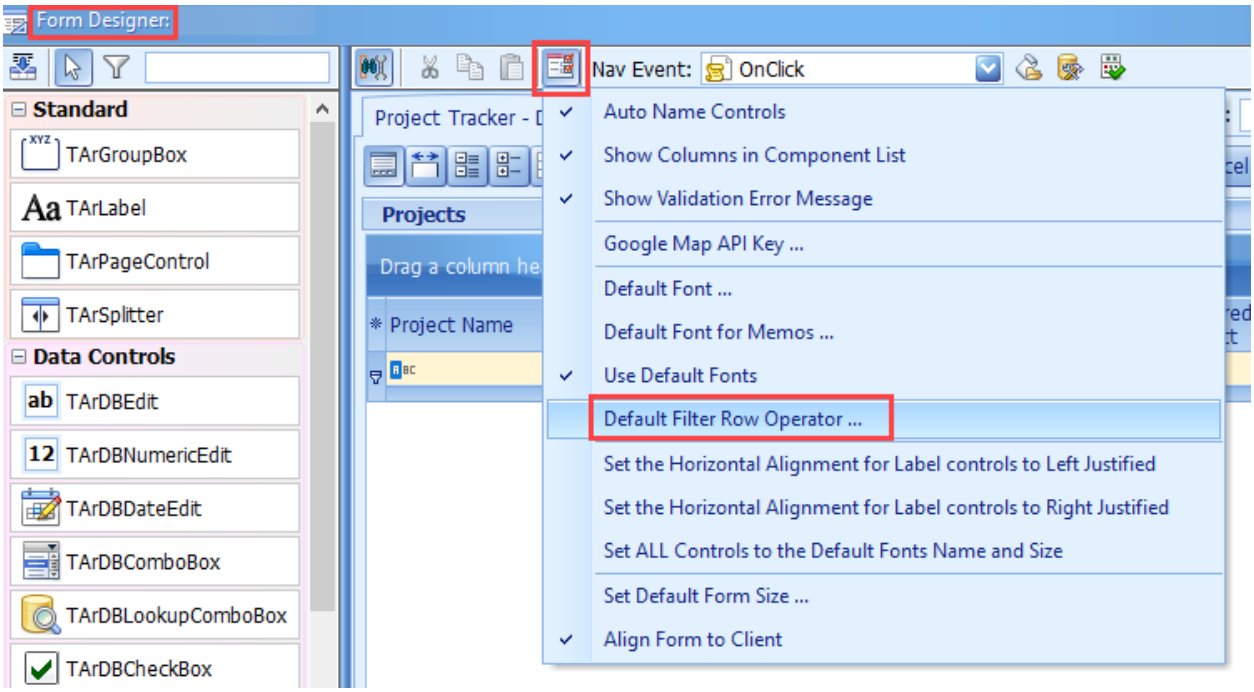


The generated script can be pasted into a procedure like the following to be used in the DataTransferExport() function.

```
procedure btnExportBridgeDataClick(Sender: TObject);
var
   strExportFolder, strExportFileName, strDescription, strExportTables: String;
   wYear, wMonth, wDay, wHour, wMin, wSec, wMSec: Word;
begin
   DecodeDate(Now, wYear, wMonth, wDay);
   DecodeTime(Now, wHour, wMin, wSec, wMSec);

   strExportFolder   := GetMyDocumentsFolder + '\RM Bridge Backup';
   //Date-stamp the export file name
   strExportFileName := strExportFolder + '\RMBridgeTablesDT_' + IntToStr(wYear) + '_' + PadLeft(IntToStr(wMonth), '0', 2) + '_' +
                        PadLeft(IntToStr(wDay), '0', 2) + '_' + PadLeft(IntToStr(wHour), '0', 2) + '_' +
                        PadLeft(IntToStr(wMin), '0', 2) + '_' + PadLeft(IntToStr(wSec), '0', 2) + '.vdt';

   strDescription := 'RM Bridge Tables Export.';

   strExportTables := '[lkupRMBRIDGEUNITS] ## [LKUPRMNAMES] ## [RMABUTMENTPICKLISTTBL] ## [RMBrdgeReplacement] ## ' +
                      '[RMBridgehis] ## [RMBridges2000] ## [RMCulverts] ## [RMLKARA] ## ' +
                      '[RMlkDeck] ## [RMlkDesignLoad] ## [RMLKOPTLCC] ## [RMLLCXINGS] ## ' +
                      '[RMPierpicklist] ## [RMPRIVATEBR] ## [RMPRIVATESPANS] ## [RMResponsiblitypick] ## ' +
                      '[RMSPANS] ## [RMSuperstucturepicklist] ## [tblDocImages] ## [tblFiscalYear]';

   DataTransferExport(tblRMBridges2000.Connection, strExportFileName, strDescription, strExportTables, False, True);
end;
```

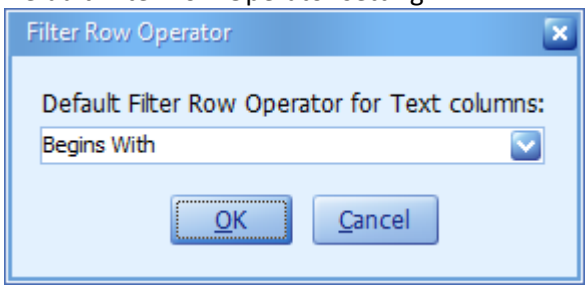
4.5 build 3

1. **New Feature:** In Form Designer, added the ability to set the default Filter Row Operator for text columns in a grid. This setting affects all new text columns that are added to a grid in the Form Designer.

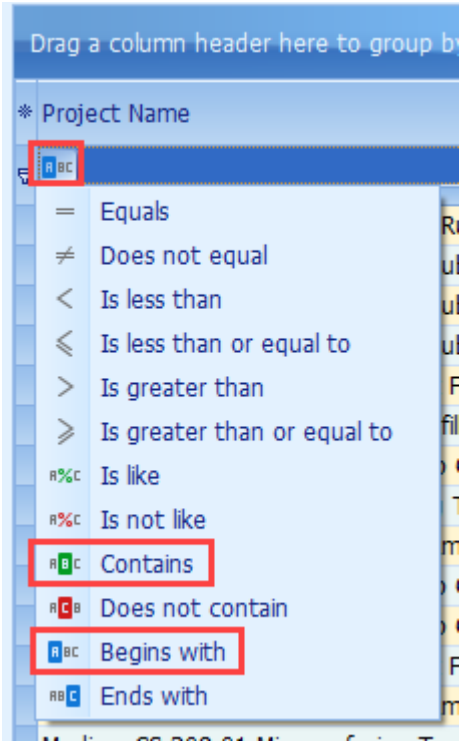


The default Filter Row Operator options for text columns are: “Begins With” and “Contains”. The default in this setting is: “Begins With”.

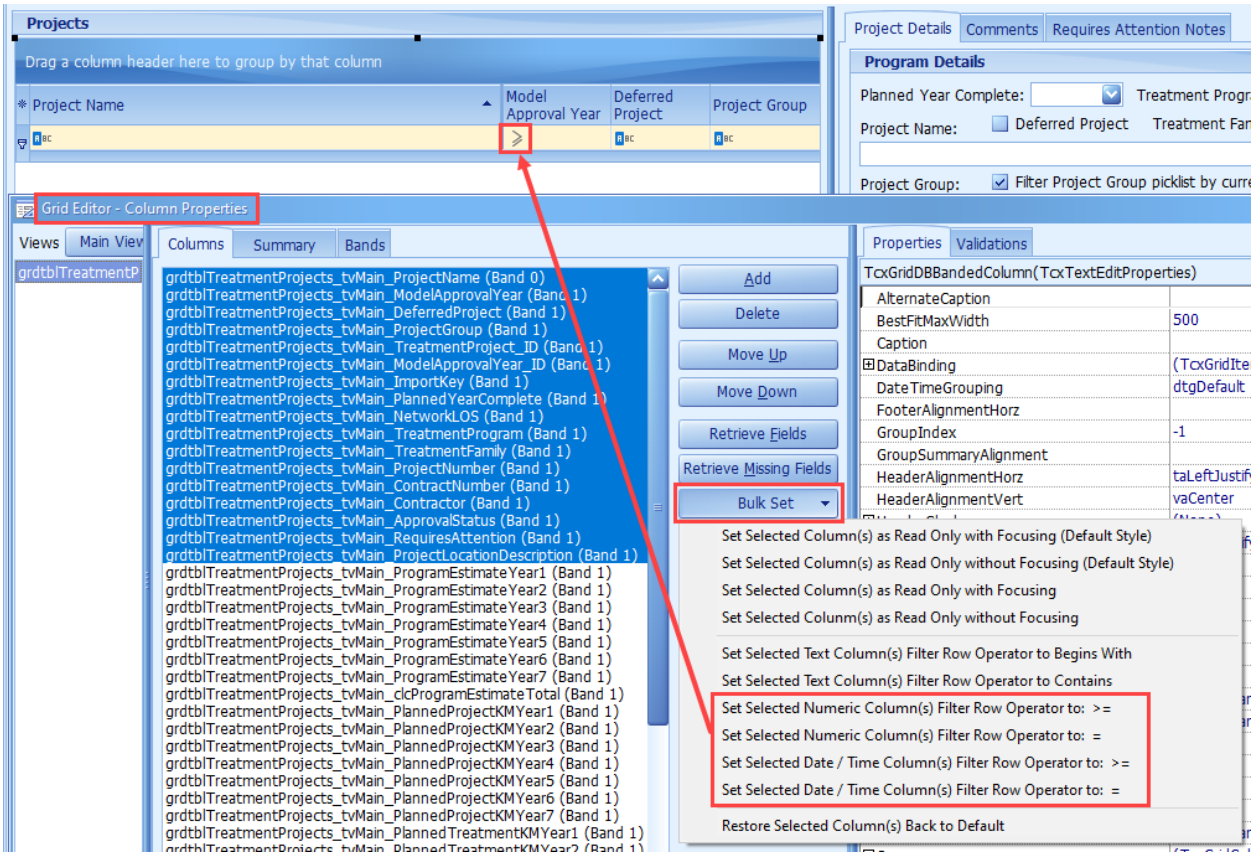
Default Filter Row Operator setting



Filter Row Operator in the grid

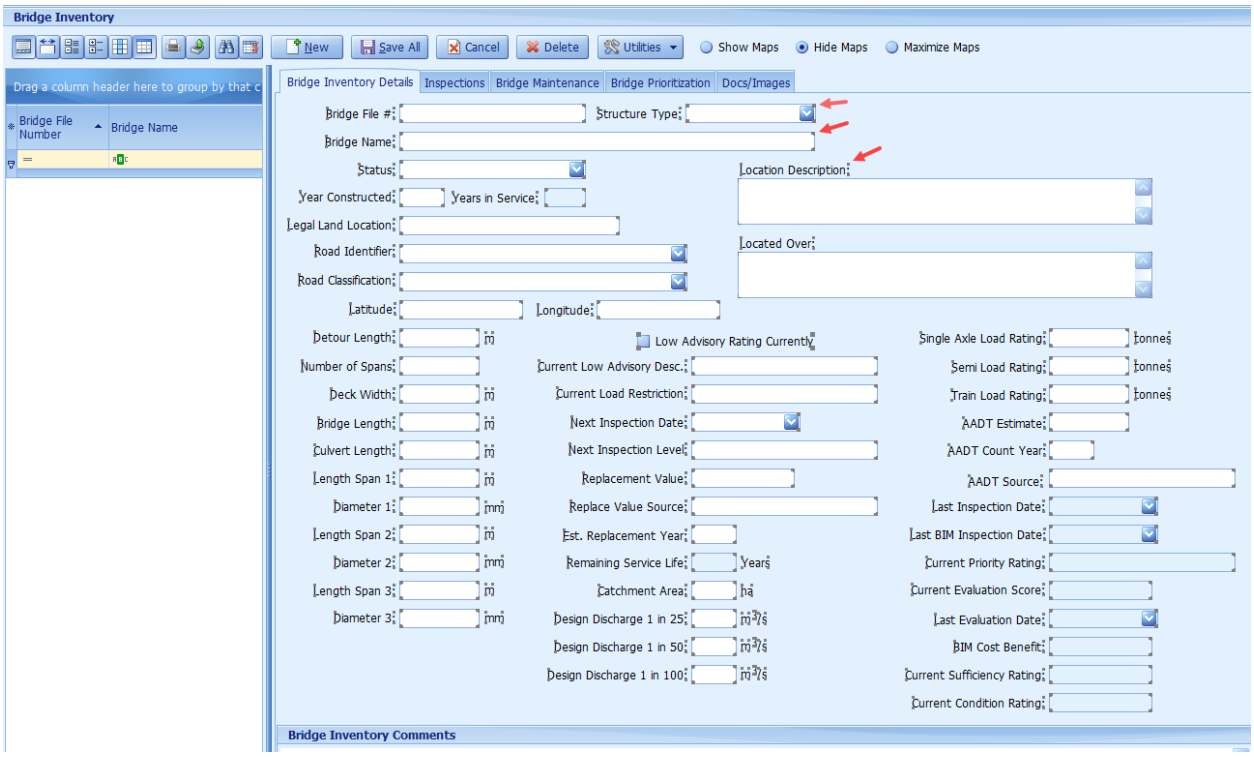


2. **New Feature:** Added additional Bulk Sets for setting the Filter Row Operator for all selected columns that are numeric or date / time. Note: all types of columns can be selected but only the columns that are of the specified data type will be changed so you do not need to individually select only the columns of that data type.



4.5 build 2

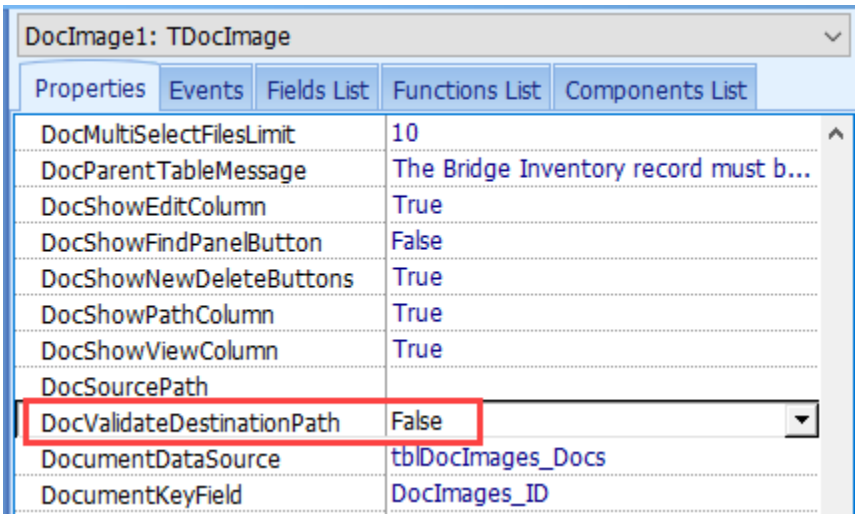
1. **Enhancement:** Significantly improved the performance of selecting many controls at once on a form in the Form Designer.



2. **Enhancement:** Significantly improved the performance of deleting many controls at once on a form in the Form Designer.

3. **Enhancement:** Modified the “Import Data Log” dialog to save and restore the dialog size and grid layout. Added the “Reset Grid Layout” popup menu to the grid for resetting the grid layout.

4. **Enhancement:** Added DocValidateDestinationPath and ImgValidateDestinationPath properties to the Docs and Images component. The reason for these properties is for situations where the permissions of the destination folder do not allow browsing (to protect the folder) or in situations where the destination path is predefined and fixed.
   When this property is set to **True**, the component checks if the destination path exists. If the destination path does not exist, the component will create it automatically.
   When this property is set to **False**, the destination path must exist because the component will not check if it exists and will not try to create the path.



5. **Issue:** Running the import in silent mode may not work in certain situations.
   Status: This issue has been resolved.

6. **Issue:** In the Form Designer, changing the name of the control on the Properties tab to be the same as other existing control's name and then clicking on the form area causes the Form Designer to crash.
   Status: This issue has been resolved.

7. **Issue:** In the Form Designer, changing the property value of the control on the Properties tab and then pressing the Enter key causes the edit box on the Properties tab to lose focus.
   Status: This issue has been resolved.

4.5 build 1

1. **Major New Feature:** Added new functions: RunImport64Bit() and RunImportSilent64Bit() to the TArDBImportData component. This enables users to import Excel and CSV files larger than 2 GB and millions of records. With these 64-bit functions, now the only limitation is available RAM on the computer running Data Manager.

   function RunImport64Bit(bolRollbackIfCancelled: Boolean): Boolean;

   function RunImportSilent64Bit(strImportFileName: String; impImportMode: TImportMode; bolShowCompletedMessage: Boolean; strVerifyFileName: String; strVerifyWorksheetName: String; bolShowProgress: Boolean; bolRollbackIfCancelled: Boolean): Boolean;

2. **Enhancement:** Improved the "Import Data" dialog to handle very large source files much more efficiently.

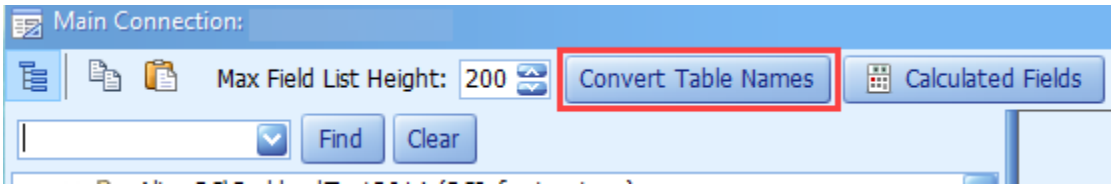
3. **Enhancement:** Added the optional parameter: "bolRollbackIfCancelled" to the RunImport function to give the option to rollback the data or not if the process is cancelled by the user. If the data is rolled back, the destination table will be the same as it was before the import started. If the rollback is turned off, then any data that was already posted to the table at the time of the cancel will remain in the table. This is handy if the user wants to cancel the process then start it again later where it left off.

   function RunImport(bolRollbackIfCancelled: Boolean): Boolean;

4. **Enhancement:** Added two optional parameters: "*bolShowProgress"* and "bolRollbackIfCancelled" to the RunImportSilent function. When the bolShowProgress parameter is True, the progress bar will be displayed during the import process. The "bolRollbackIfCancelled" is the same as in the RunImport function mentioned above except this is only applicable if the bolShowProgress is also set to True.

5. **Issue:** When importing Excel or CSV files, in some cases the index of the columns may be referenced incorrectly.
   Status: This issue has been resolved.

6. **Enhancement:** Improved the "Convert Table Names" feature in the Main Connection and Lookup Connection forms to handle more types of conversions of the tables and views in the form and in the scripts.



7. **Enhancement:** In the Form Designer's Code tab, improved the font color for the selected item within the Auto Completion dropdown list so it is more readable.

8. **Enhancement:** Added support for the SQL Server datetime2 data type in Maintain Database dialog and in the Import component. This data type is needed if date/time data with high precision is needed. The datetime2 data type has a time accuracy of 100 nanoseconds.

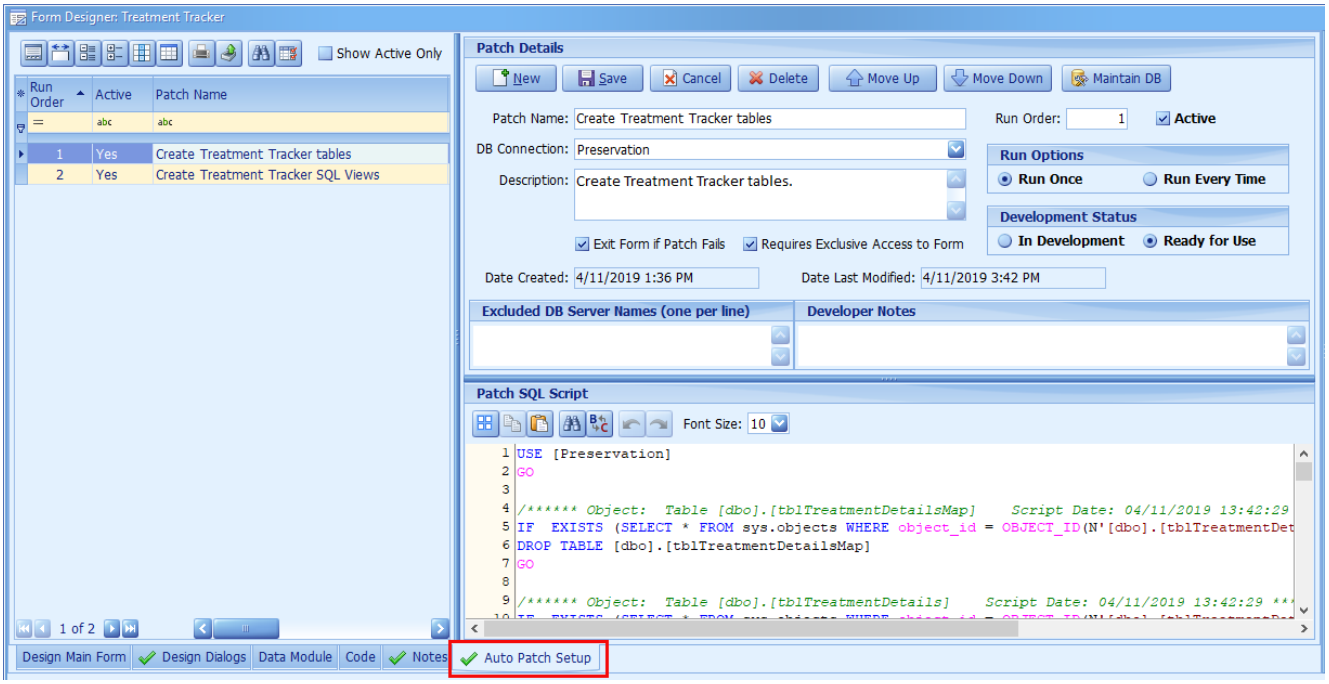
4.5 build 0

1. **Major Upgrade:** Upgraded all components and development tools to provide better support for the latest build of Windows 10. This also resulted in many improvements in all the components including all the editors, grid, GIS Map, Google Map, etc.

2. **Major Upgrade:** The Scripting Engine has also been upgraded to the latest version to significantly improve the performance of forms that contain a lot of scripting. As an example, a form that contains over 9,100 lines of scripting loaded about 30% faster after this upgrade.

3. **Major New Feature:** Added the tab: "Auto Patch Setup" to the Form Designer for creating Patch scripts that will be run automatically when the user(s) click on the form in Viewer mode. The purpose of this feature is to add automatic SQL scripting to be run on remote computers. For a patch to run, the Active check box must be checked, and Development Status must be set to "Ready for Use". Also, the patch will not run on any database servers that are added to the "Excluded DB Server Names" list.
   Run Order is used to control the order that multiple patches will be run. To change the Run Order, use the "Move Up" or "Move Down" buttons.
   Run Options:
   a. Run Once – means the patch will only be run one time. This is used when modifying the database structure, for example.
   b. Run Every Time – means the patch will be run every time by every user. This option will be used less often but can be used to modify or update data, for example.

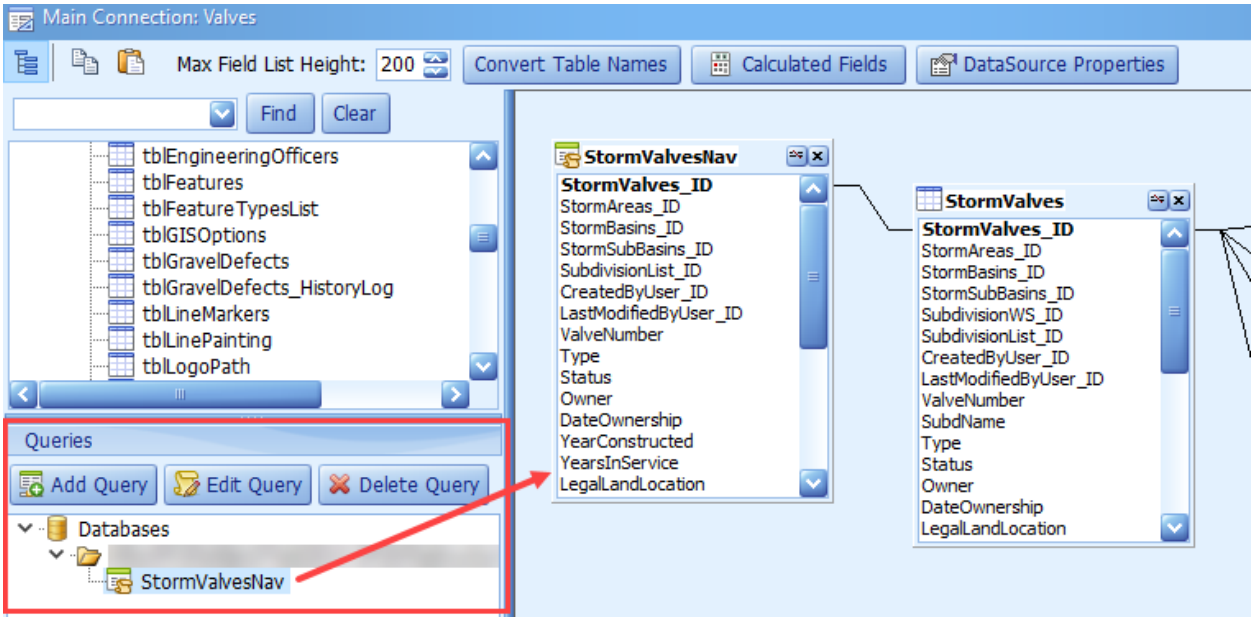
   This tab is only visible when the user security setting "User Can Create Auto Patch" is set to Yes. The default setting for this is No.
   Contact VEMAX if you need any assistance with creating the SQL scripts.



4. **Major Enhancement:** Added support for Microsoft SQL Server 2019. The supported versions of SQL Server are:
    - SQL Server 2019 ← This is the latest official release version, as of January 2022.
    - SQL Server 2017
    - SQL Server 2016
    - SQL Server 2014
    - SQL Server 2012
    - SQL Server 2008 R2 ← not recommended

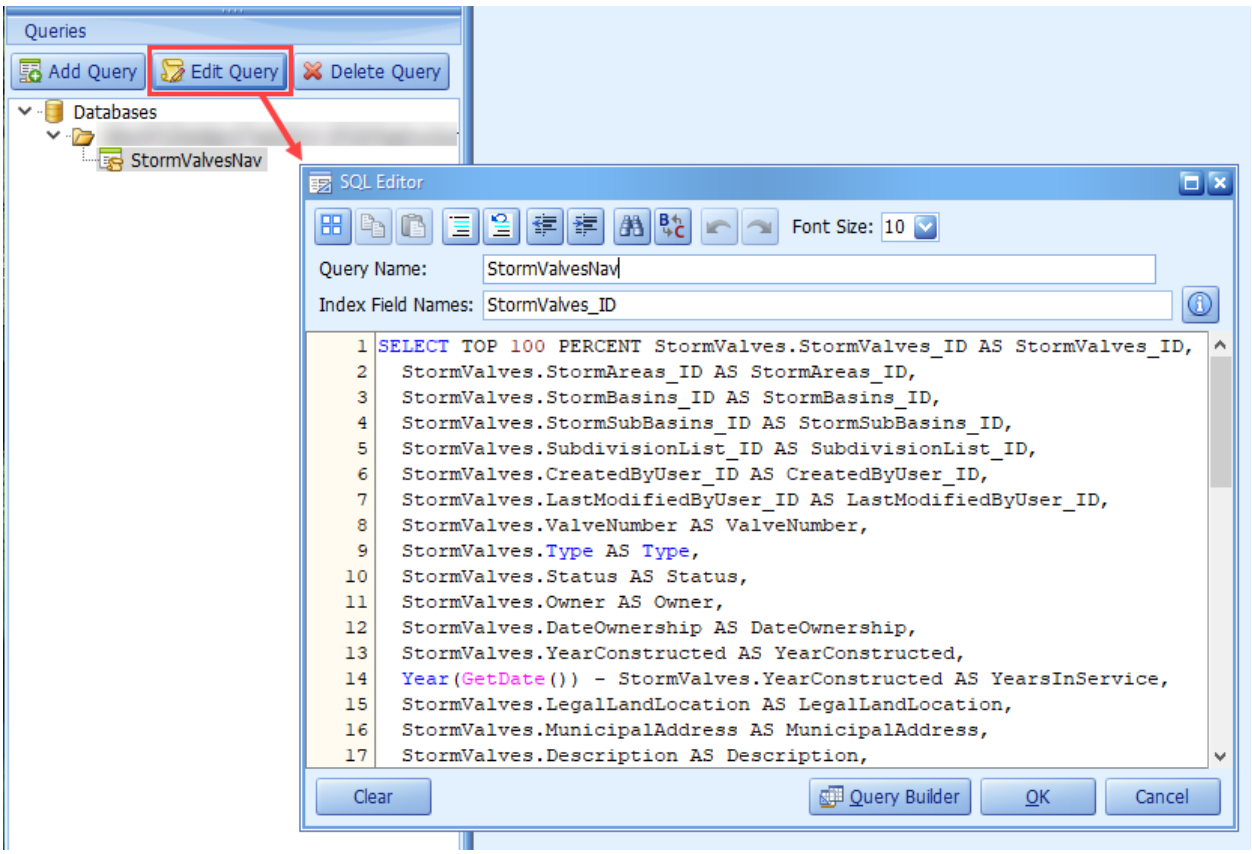
5. **Major New Feature:** Added the Queries feature to the Main Connection and Lookup Lists Connection dialogs. This allows the developers (Data Manager Designer users) to create and maintain queries in Data Manager Designer without the need to create SQL views in SQL Server or in Data Viewer Designer. Single-table and multi-table queries can also be used as a read-only DataSource for the Navigator tree or in the grid. The queries created in Data Manager Designer are stored in the Data Manager database and are part of the configuration of that form just like any part of the Main and Lookup connections.

   To create a query, click on the "Add Query" button. To modify an existing query, select that query in the Queries list and click on the "Edit Query" button.
   Add the query object to the DataSource Panel using drag and drop the same way as for a table or a SQL view.

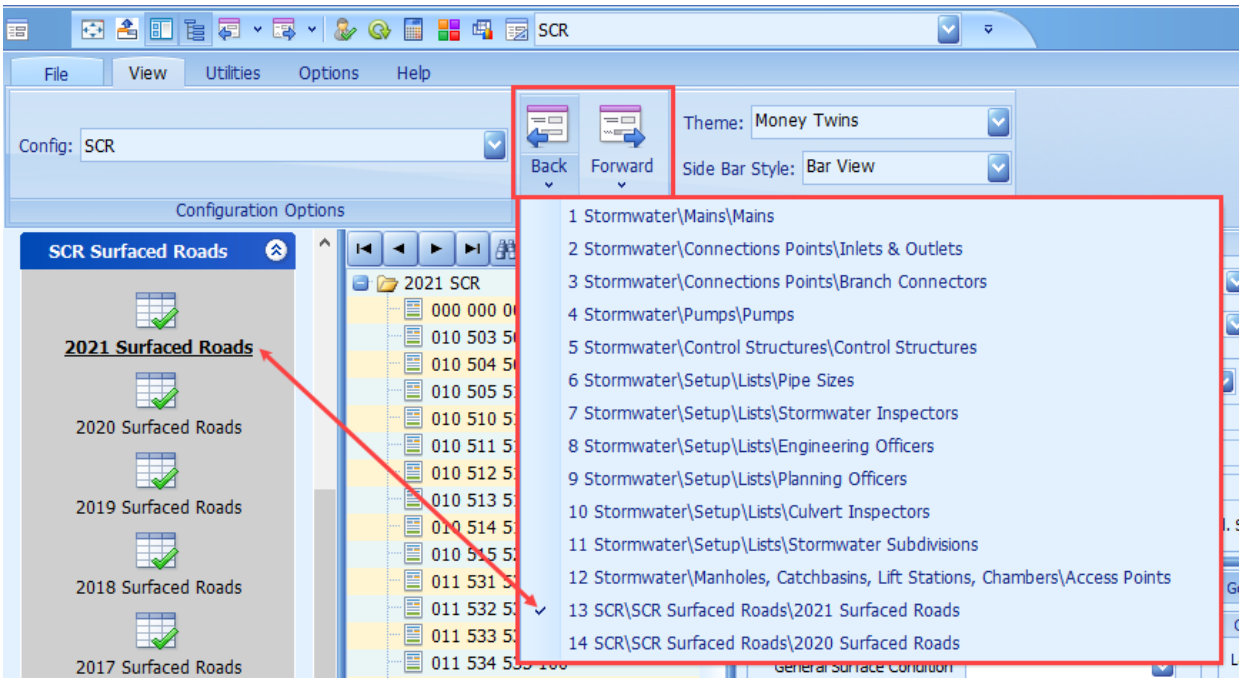


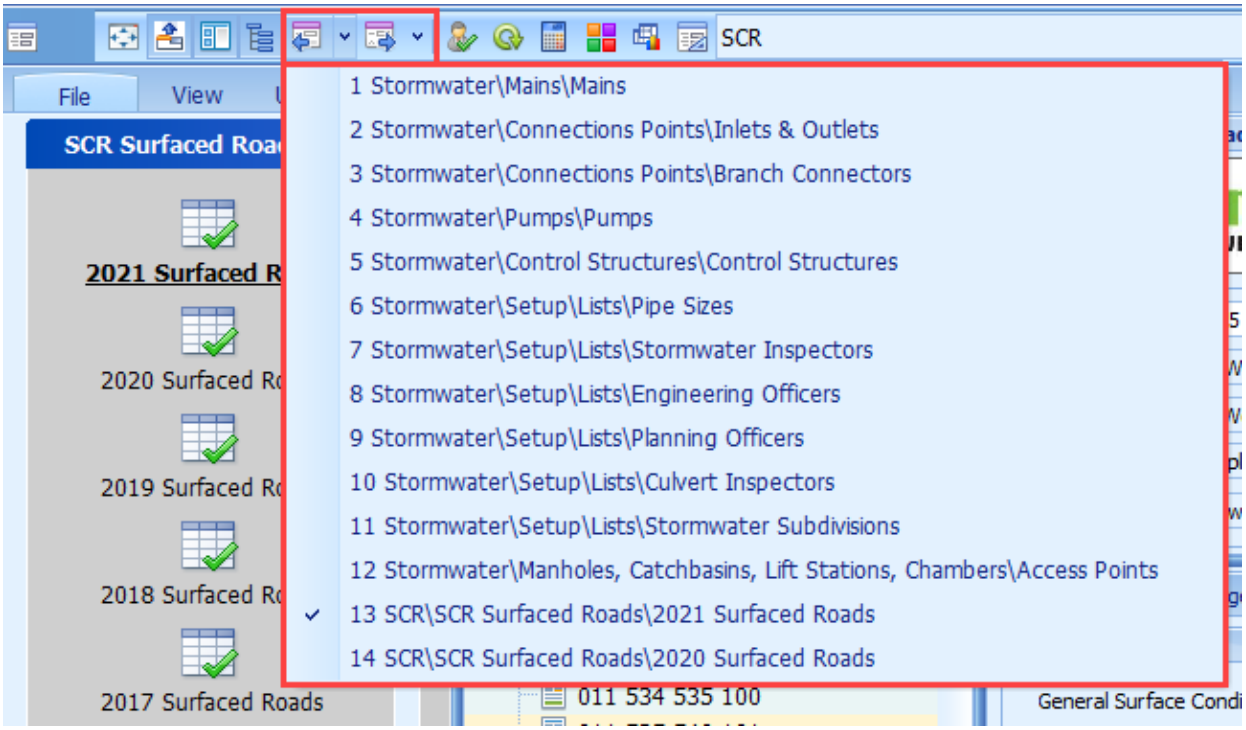
The SQL Editor for the queries is shown below.
The Query Name can be anything that is meaningful to you.
The Index Field Names is used to sort the data in the query.

Click on the  button for more information regarding index field names.


Queries
Add Query
Edit Query
Delete Query
Databases
StormValvesNav
SQL Editor
Font Size: 10
Query Name:
StormValvesNav
Index Field Names:
StormValves_ID
SELECT TOP 100 PERCENT StormValves.StormValves_ID AS StormValves_ID,
StormValves.StormAreas_ID AS StormAreas_ID,
StormValves.StormBasins_ID AS StormBasins_ID,
StormValves.StormSubBasins_ID AS StormSubBasins_ID,
StormValves.SubdivisionList_ID AS SubdivisionList_ID,
StormValves.CreatedByUser_ID AS CreatedByUser_ID,
StormValves.LastModifiedByUser_ID AS LastModifiedByUser_ID,
StormValves.ValveNumber AS ValveNumber,
StormValves.Type AS Type,
StormValves.Status AS Status,
StormValves.Owner AS Owner,
StormValves.DateOwnership AS DateOwnership,
StormValves.YearConstructed AS YearConstructed,
Year(GetDate()) - StormValves.YearConstructed AS YearsInService,
StormValves.LegalLandLocation AS LegalLandLocation,
StormValves.MunicipalAddress AS MunicipalAddress,
StormValves.Description AS Description,
Clear
Query Builder
OK
Cancel

6. **Major New Feature:** Added the Back and Forward buttons to the Navigation group on the View tab in the Ribbon as well as in the quick access bar (at the top of the application) for going back and forth between several recently accessed forms during the current session. To see a list of those forms in the order they were accessed and to navigate to any one of them, click on the down arrow button. The checkmark beside the form in the list, indicates the current form.

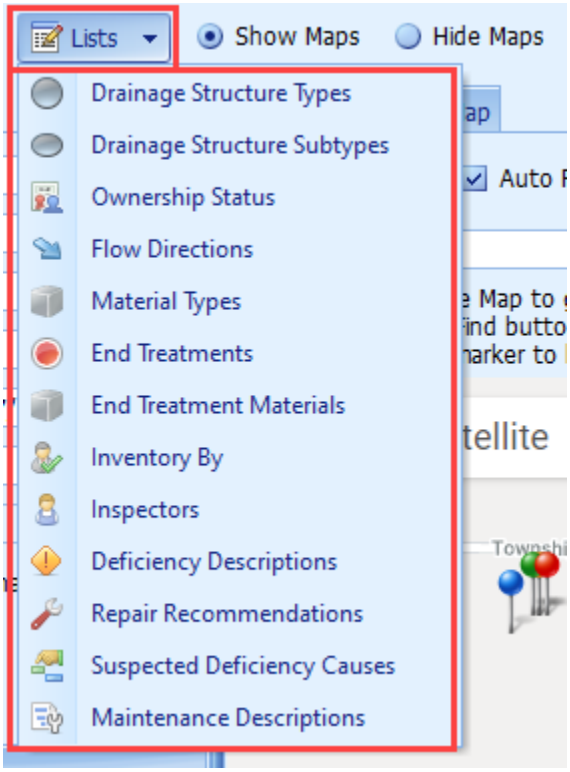




7. **Major New Feature:** Added the GotoForm() procedure to be used in scripting to go to any specific form in Data Manager that the user has access to. If the "strFormName" parameter is blank, it goes to the specified Group Item. This can be used to make it very handy for users to go to an applicable list form, for example.

   GotoForm(strConfigName, strGroupName, strGroupItemName, strFormName);

   Here is an example of how the GotoForm procedure can be used to compliment a dropdown menu button to enable the user to quickly and easily go to various list forms.



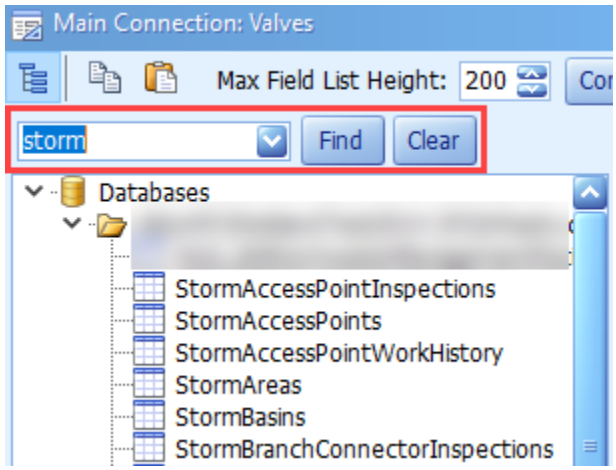
Here is the script for the above-mentioned menu button's MenuItemClick event:

```
procedure btnMenuListsMenuItemClick(Sender: TObject; intMenuItemID: Integer);
begin
   case intMenuItemID of
      1: GotoForm('Stormwater', 'Setup', 'Lists', 'Drainage Structure Types');
      2: GotoForm('Stormwater', 'Setup', 'Lists', 'Drainage Structure Subtypes');
      3: GotoForm('Stormwater', 'Setup', 'Lists', 'Ownership Status');
      4: GotoForm('Stormwater', 'Setup', 'Lists', 'Flow Directions');
      5: GotoForm('Stormwater', 'Setup', 'Lists', 'Material Types');
      6: GotoForm('Stormwater', 'Setup', 'Lists', 'End Treatments');
      7: GotoForm('Stormwater', 'Setup', 'Lists', 'End Treatment Materials');
      8: GotoForm('Stormwater', 'Setup', 'Lists', 'Inventory By');
      9: GotoForm('Stormwater', 'Setup', 'Lists', 'Inspectors');
      10: GotoForm('Stormwater', 'Setup', 'Lists', 'Deficiency Descriptions');
      11: GotoForm('Stormwater', 'Setup', 'Lists', 'Repair Recommendations');
      12: GotoForm('Stormwater', 'Setup', 'Lists', 'Suspected Deficiency Causes');
      13: GotoForm('Stormwater', 'Setup', 'Lists', 'Maintenance Descriptions');
   end; //case
end;
```

8. **Major New Feature:** Added the GotoDataView() function to be used in scripting to go to a specific Data View in Data Viewer that the user has access to.

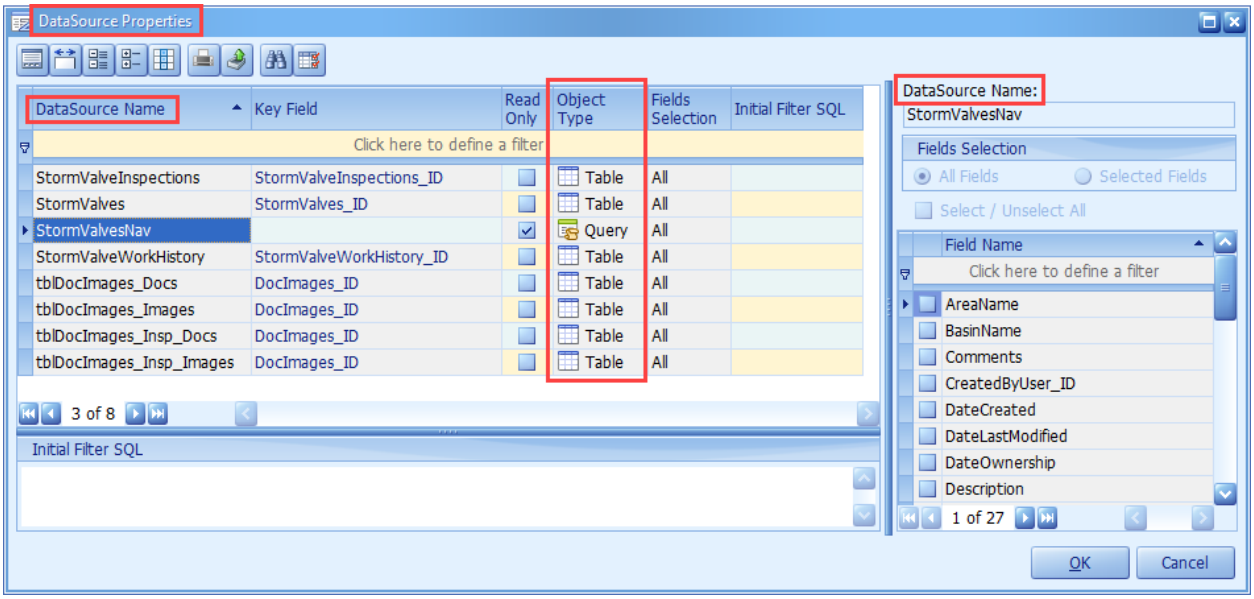
   GotoDataView(strConfig, strGroup, strDataView);

9. **Enhancement:** Added the Find panel to the Databases tree in the Main Connection and Lookup Lists Connection dialogs for finding the table(s) or view(s) in the Databases tree. This is very handy if the list of tables and/or views is long.

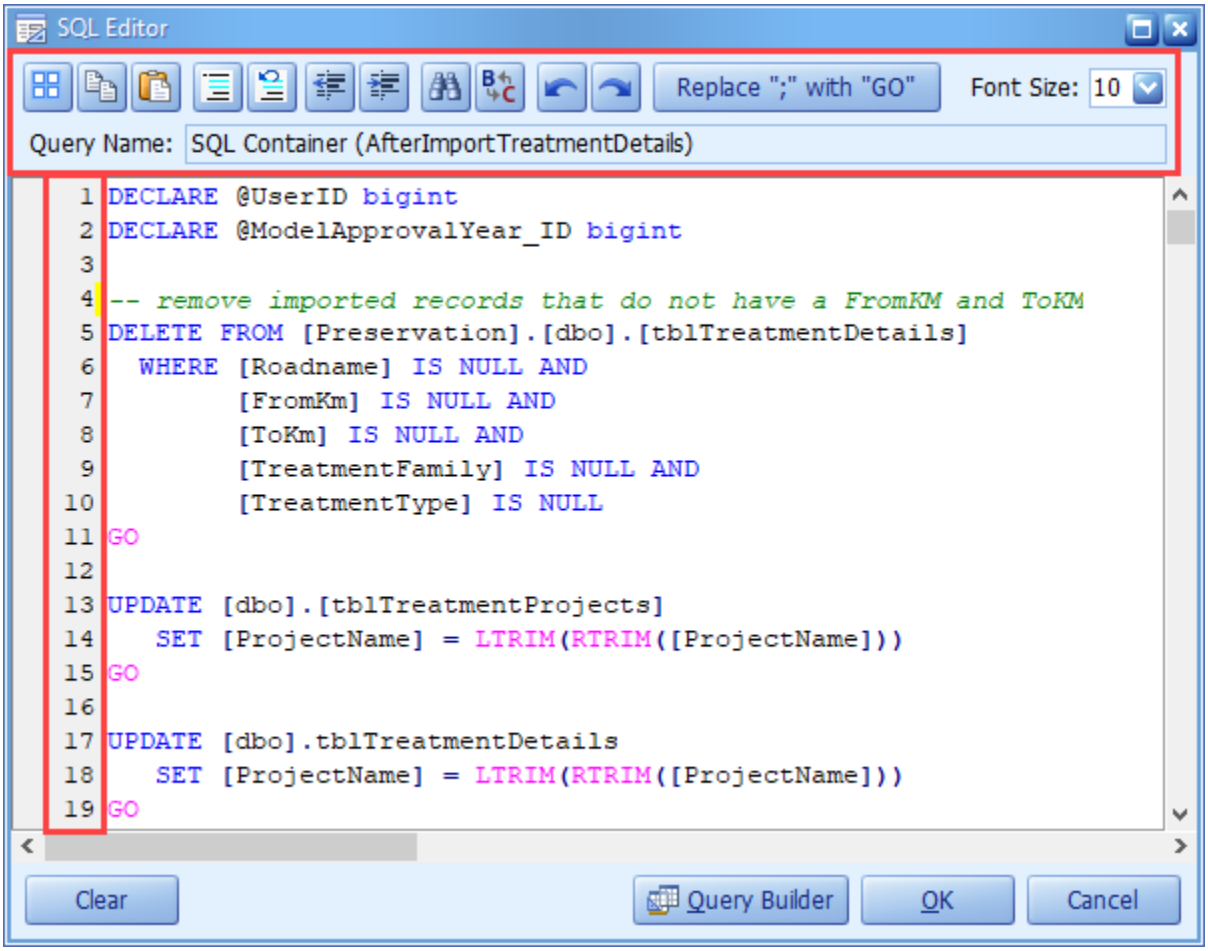


10. **Enhancement:** Added "Remove Database" popup menu to the database tree in the Main Connection and Lookup Lists Connection dialogs for removing the database from the database tree. Also enhanced the process to load the Main Connection and Lookup Lists Connection dialogs with proper controls when an error occurred with the database connection.

11. **Enhancement:** Changed the caption for the "Table Properties" dialog to "DataSource Properties" to accommodate more object types: Table, View, Query. The column caption "Table Name" has been changed to "DataSource Name". Also added the "Object Type" column that displays the object type for the DataSource with corresponding icon.



12. **Major Enhancement:** Replaced the scripting memo component in the SQL Container Editor with a much more advanced one. Also added line numbers and a toolbar to the SQL Container Editor. The buttons on this toolbar are as follows:
Select All, Copy, Paste, Find, Find and Replace, Undo, Redo, Replace ";" with "GO". The Font Size is used to increase or decrease the size of the font in the memo box.
The Replace ";" with "GO" button is used to replace all semicolons with a "GO" on a new line.

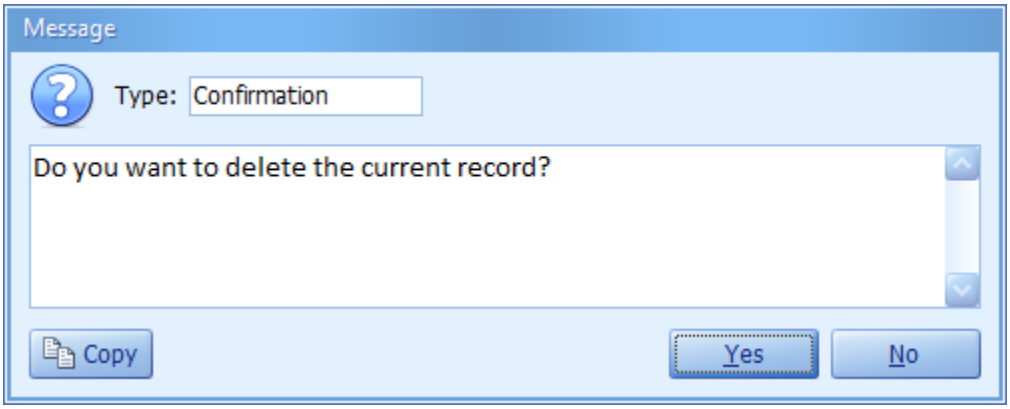


13. **Major Enhancement:** Redesigned the function: MessageDlg() to display a more user-friendly and a much nicer looking message dialog. Also added overloaded parameters to support passing "More Information" and/or a "System Error" message and/or a "Message Code" to be displayed on the message dialog.
The message code will only be displayed if a non-zero value is used. If the message code is used, it should be unique because it is useful for the users to identify the exact message that is displayed when communicating to the support staff. This will enable the developer to search the script for that message code to find where the message is called.
All message dialogs will automatically use this new design without the need to make any scripting changes.

Syntax of the MessageDlg with the additional parameters:

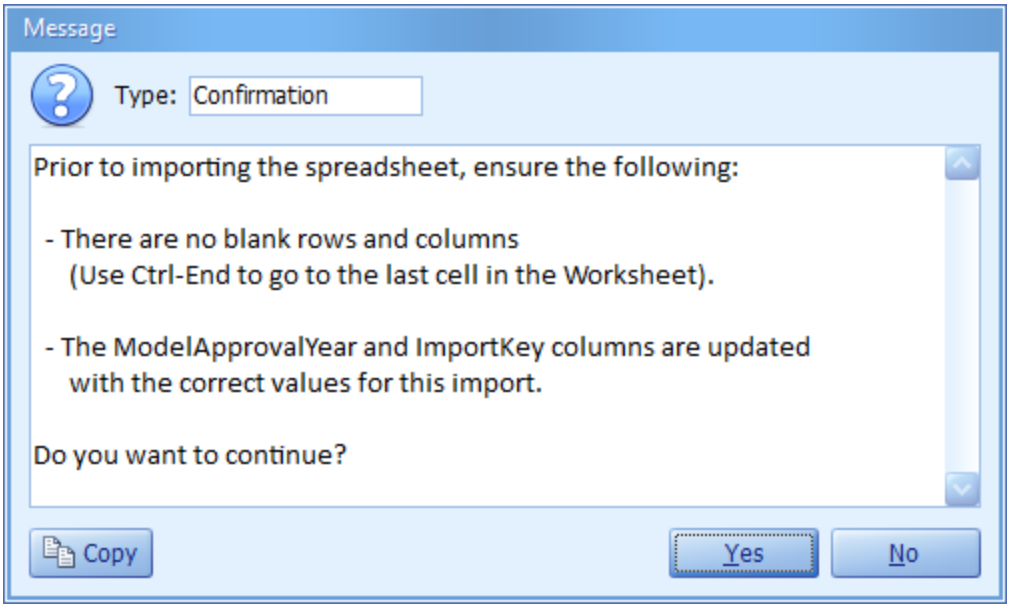
MessageDlg('', mtInformation, SetOf([mbOk]), **0**, strMoreInfo, strSystemMessage, **0**);

Example confirmation message dialog:



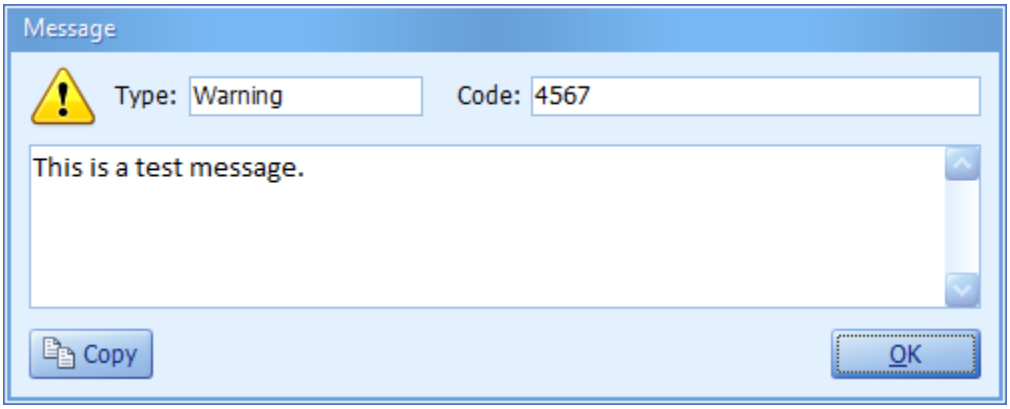
The Copy button enables a user to easily copy the text of the message to send in an email. This copy button will also include system information that may be helpful to support staff.

The height of the message box will grow automatically to fit the text of the message.



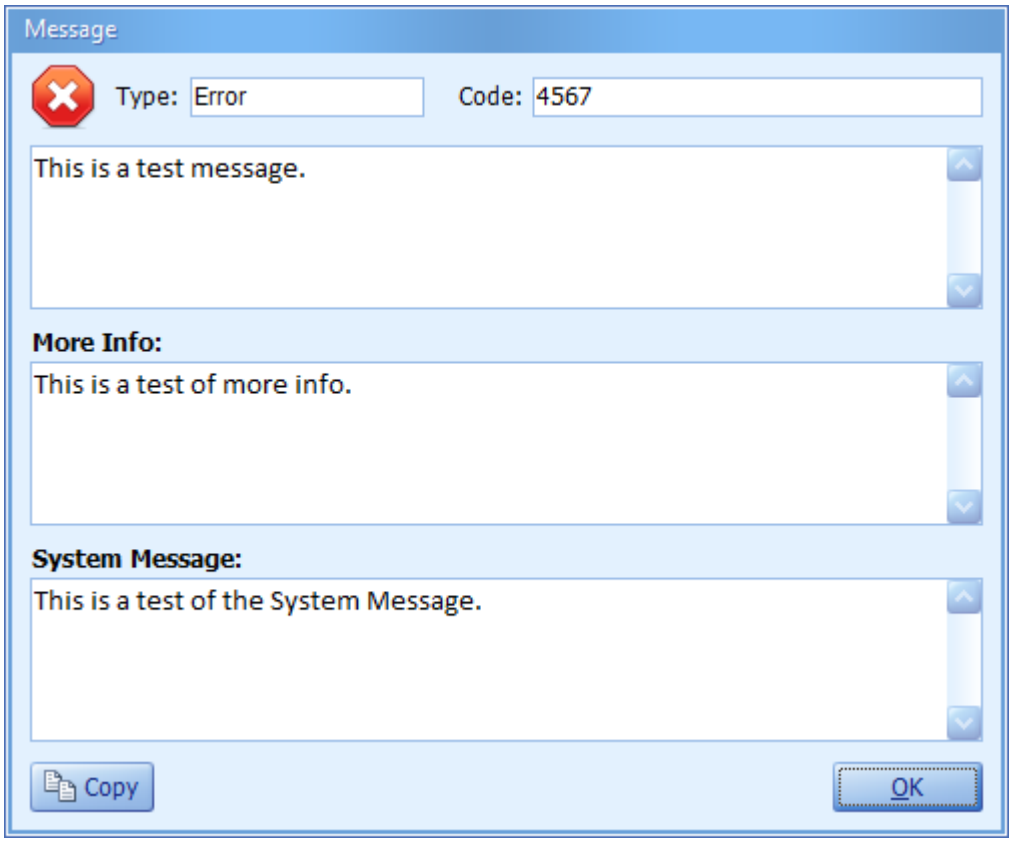
Example of a Warning message with no more info, no system info but with a message code:

MessageDlg('This is a test message.', mtWarning, SetOf([mbOk]), **0**, '', '', **4567**);

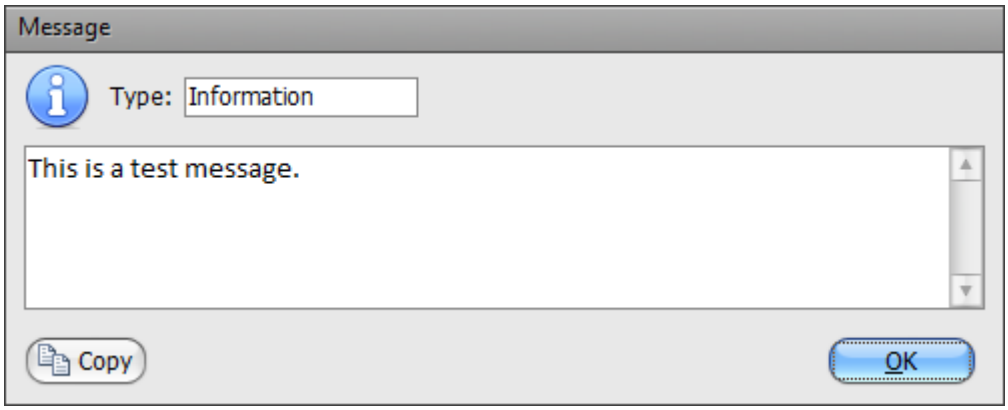


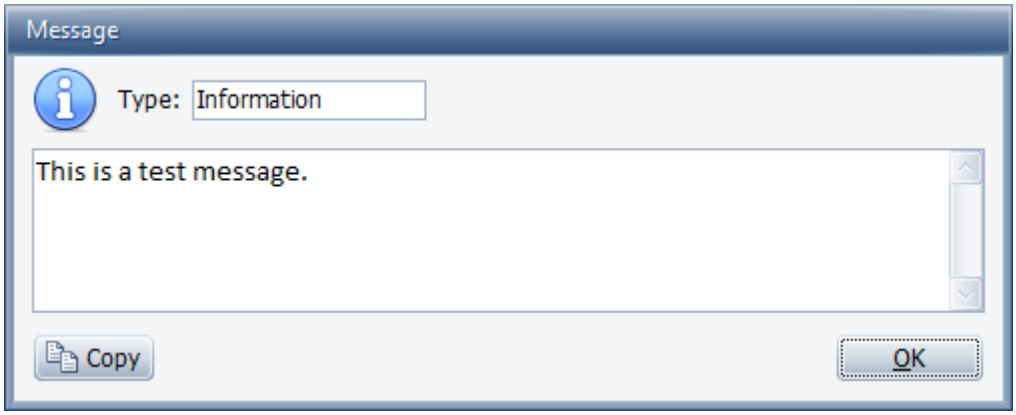
Example of an Error message with more info, system info, and a message code:

```
MessageDlg('This is a test message.', mtError, SetOf([mbOk]), 0,
           'This is a test of more info.',
           'This is a test of the System Message.', 4567);
```

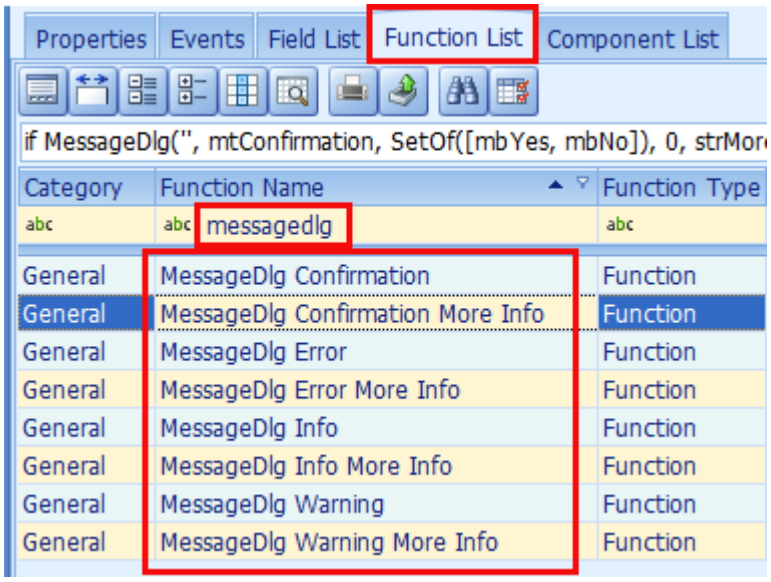


The new MessageDlg automatically supports all the themes:

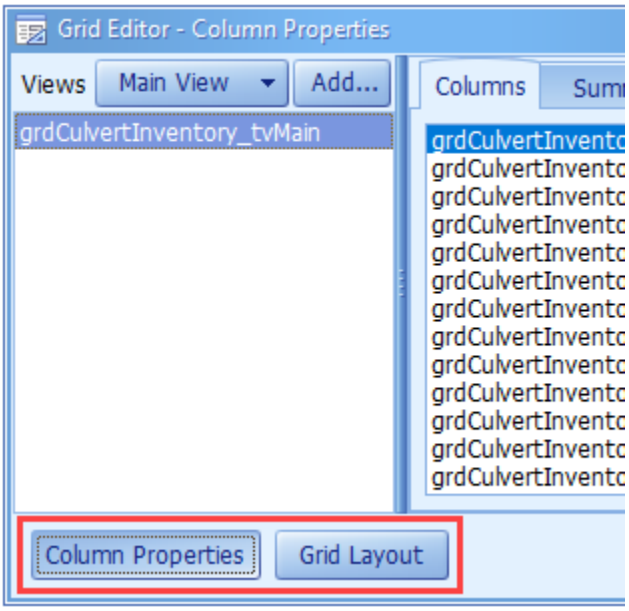




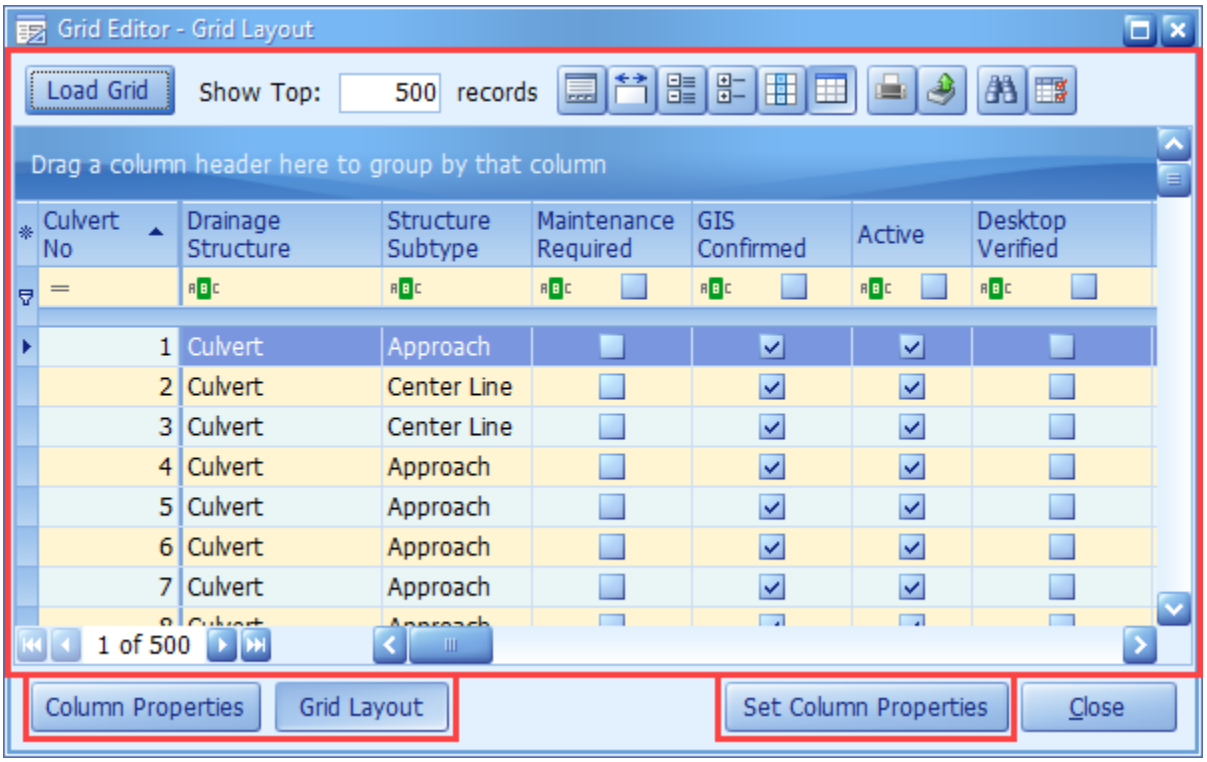
Search for “messagedlg” in the Function Name column in the Functions List to get a list of the various ways to use the MessageDlg function.



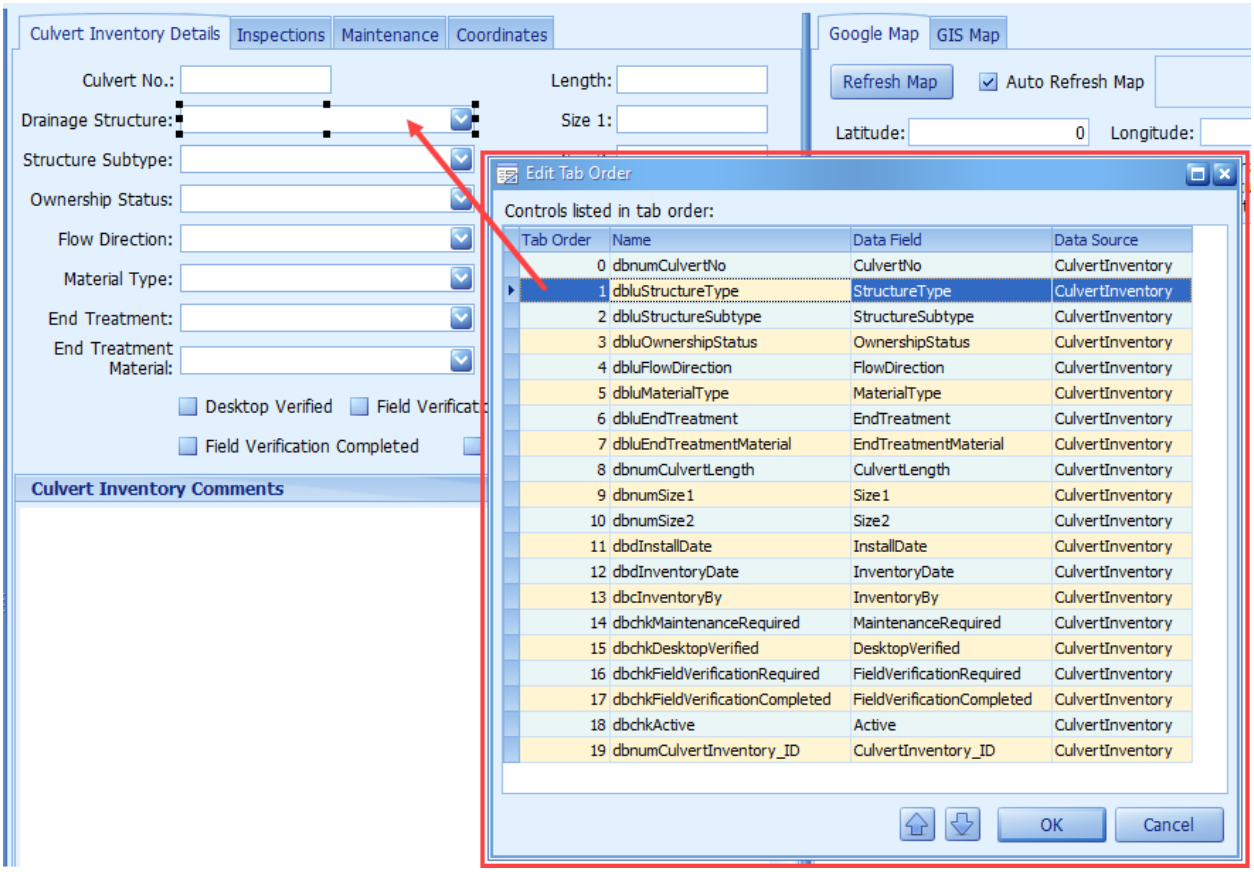
14. **Major Enhancement:** In the Grid Editor, we have added a Grid Layout page to visually set several properties of the columns in the grid such as, column order, width, sort order, footers, etc. This is a huge time saver over setting the properties of each column individually.



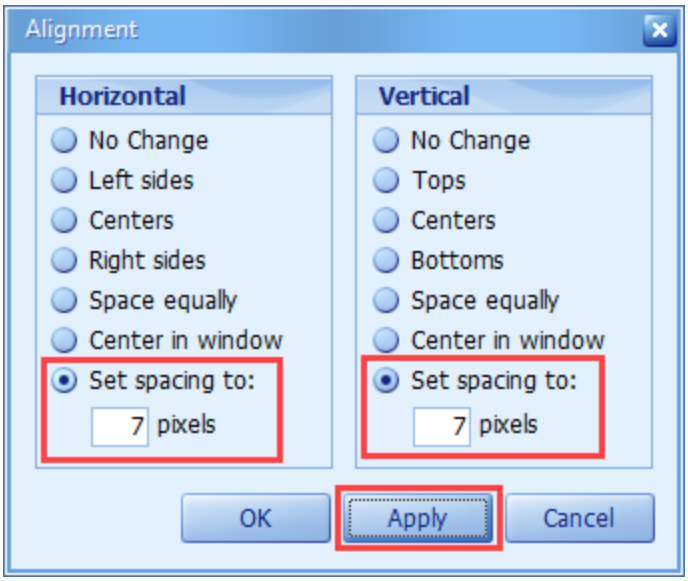
In the Grid Layout page, click on the Load Grid button to load the grid with a set number of records. The default is 500 records. Set the layout of the grid just like in Data Manager. When you are done, click on the "Set Column Properties" button to set all the properties of the changes that you made.



15. **Major Enhancement:** Redesigned the Tab Order dialog to include the following features:
    a. Grid includes the columns: “Tab Order”, “Name” of the control, “Data Field”, and “Data Source” for data-aware controls.
    b. The tab order of the items in the grid can be changed using drag and drop or by the  buttons.
    c. When an item is selected in the grid, the corresponding control on the form will show the “grab handles” so you know what control that is.
    d. Verifying that the tab order is correct can be done simply by traversing the grid with the up and down arrow keys on the keyboard. This is the handiest feature of this new design because it makes it much easier to know if the tab order is correct visually.
    e. The Tab Order dialog is automatically resized depending on the number of items in the grid. The fewer the items, the smaller the dialog will be so that it does not cover up too much of the form unnecessarily.



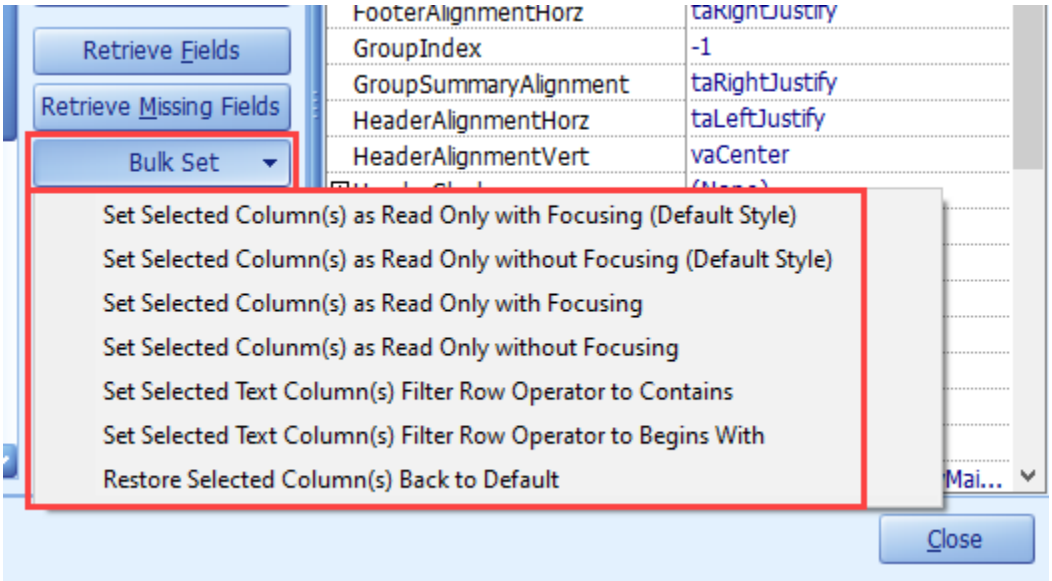
16. **Enhancement:** In the Alignment dialog, added "Set spacing to" options and edit boxes for setting custom spacing for the horizontal and vertical alignments. This is very handy when you want selected controls to have a specific amount of space between them regardless of the size of each control. We also added the Apply button so you can instantly see the results of the change and adjust the spacing a second time if necessary.



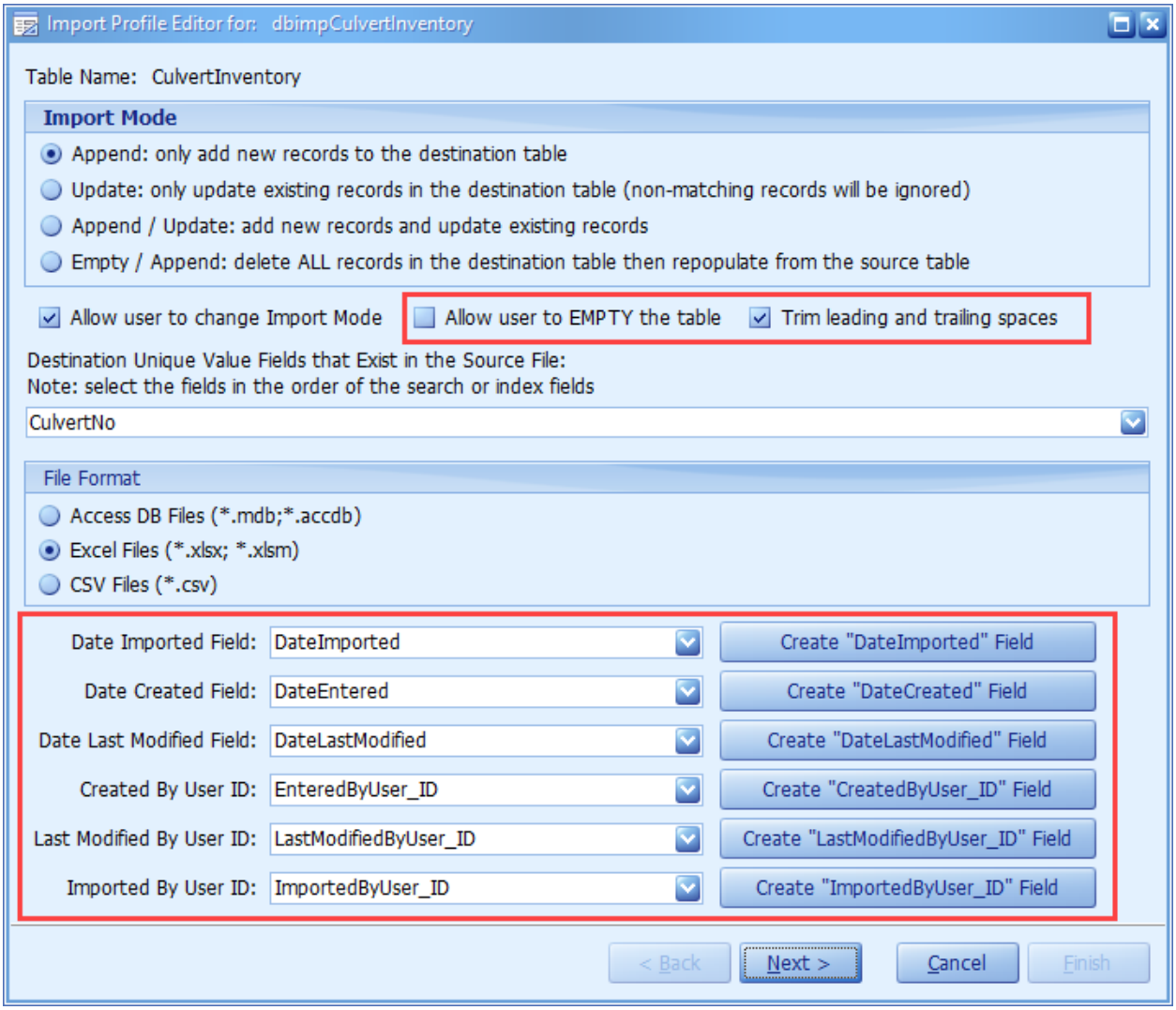
17. **Enhancement:** The Apply button was added to the Size dialog for the same reason as for the Alignment dialog.

18. **Enhancement:** Modified the "Retrieve Fields" and "Retrieve Missing Fields" buttons on the "Grid Editor" dialog to skip setting the "Date Entered", "Date Last Modified", "Entered By User", and "Last Modified By User" to read only style when the DataSource is already read only.

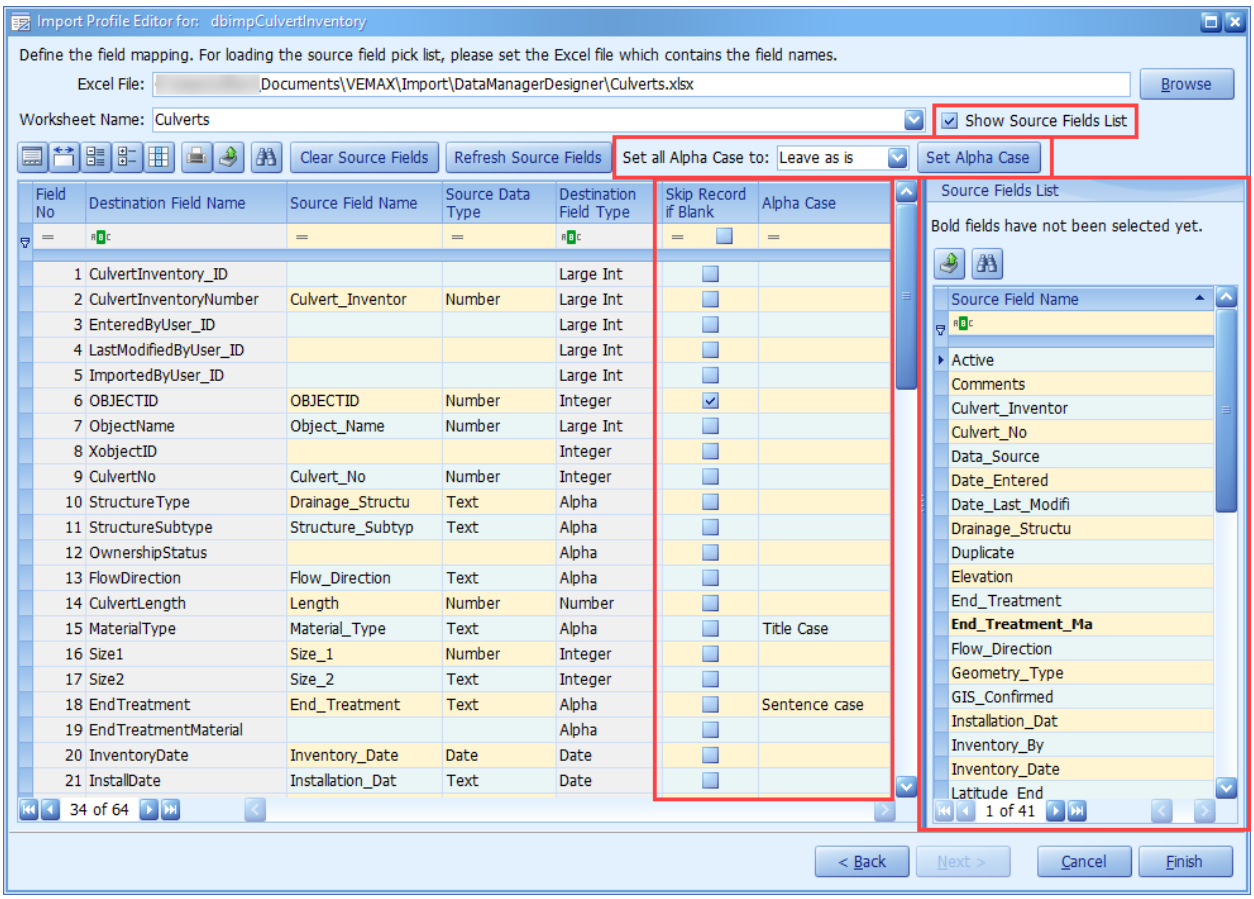
19. **Enhancement:** Made several improvements to the logic of "Bulk Set" feature for various field types like numeric, date, time, etc..



20. **Major Enhancement:** Redesigned the Import from Excel and csv. The import process is significantly faster and no longer requires the Access Database Engine for Excel and csv files. The Access Database Engine is still needed to import Access databases.
The following enhancements were made to the Import Profile Editor dialog:
    - Added the option: "Allow user to EMPTY the table". This enables the developer to allow users to select the Import Mode except the Empty option.
    - Added the option: "Trim leading and trailing spaces". Quite often data in Excel spreadsheets have unnecessary spaces before and after text in cells. If this option is checked, those spaces will be removed during the import process.
    - Added the picklists to selected fields for "Date Imported", "Date Created", "Date Last Modified", "Created By User ID", "Last Modified By User ID", "Imported By User ID". These fields will be automatically updated during the import process which means scripting is no longer required to update these fields for imported records. If any of these fields do not exist in the table, click on the "Create…" button beside that field to add that field to the table.



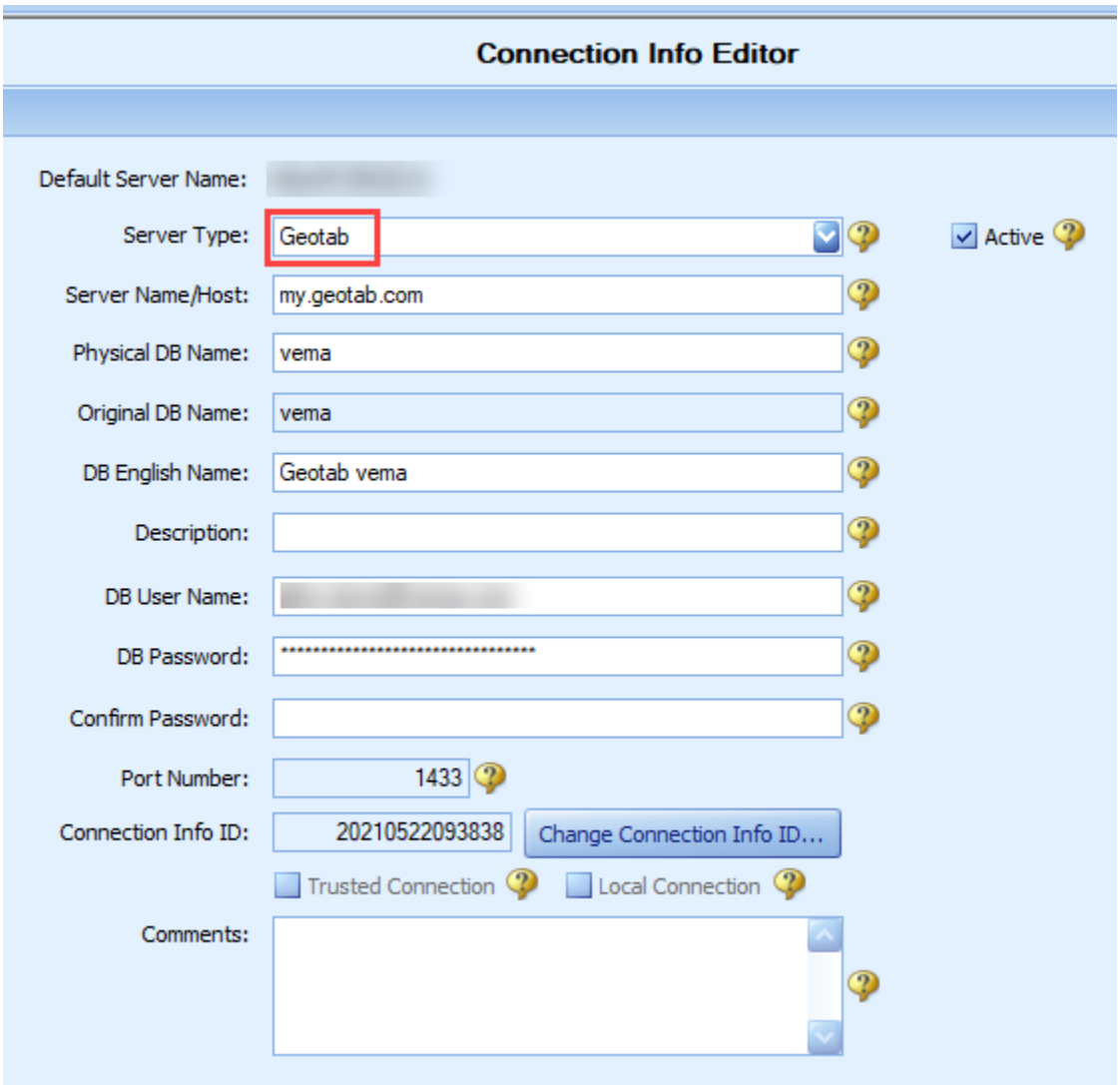
21. **Enhancement:** In the Import Profile Editor dialog, added the following to the Field Mapping page:
    - Added a column called: “Skip Record if Blank”. If checked, the record being imported will be skipped completely if the cell is blank. This can apply to any field but is particularly handy for fields that have unique data and should not have any blank values. For example, if there are blank rows in the spreadsheet, they will be skipped if the unique field is set to skip record if blank.
    - Added a column called: “Alpha Case”. This is used to convert data to Uppercase, Lowercase, Title Case, or Sentence Case. This only applies to Alpha fields so it will be ignored if a non-alpha field is set. Press the “Set Alpha Case” button to set all the Alpha fields to the selected case and to clear the Alpha Case for all non-alpha fields.
    - Added the Source Fields List (grid on the right side) to show all the columns in the selected worksheet.
      The bold fields in this list are the ones that have not been selected in the grid on the left side.



22. **Enhancement:** Modified the “Import Profile Editor” in the Import component to remember the last mapped field names and repopulate them when an import file is selected. This is handy when the column names in the spreadsheet do not match the field names in the table.

23. **New Feature:** Added support for importing Vehicle Location and Trips data directly from the Geotab © website: https://www.geotab.com
This is useful for tracking equipment and vehicles equipped with Geotab © GPS devices.
A Geotab account is required for this feature.

An unlimited number of Geotab database login accounts can be setup in the Connection Info Editor in the Custom Control Panel (CCP). The new Server Type: "Geotab" was added to the Server Type picklist to support this feature.



The script to import the Vehicle Location data is:

```
procedure btnDownloadGeotabDataClick(Sender: TObject);
var
  intRecordsImported: Int64;
begin
  intRecordsImported := GeotabImport(tblGeotabVehicleLocation.Connection,
                                     20210522093838,
                                     AppConstants.GeotabImportVehicleLocation, '');

  if (intRecordsImported = 1) then
  begin
    RefreshTable(tblGeotabVehicleLocation);
    ShowMessage('Successfully imported: 1 record.');
  end
  else if (intRecordsImported > 1) then
  begin
    RefreshTable(tblGeotabVehicleLocation);
    ShowMessage('Successfully imported: ' + IntToStr(intRecordsImported) + ' records.');
  end
  else if (intRecordsImported = 0) then
    ShowMessage('No additional records to import.')
  else if (intRecordsImported < 0) then
    ShowMessage('Error: cannot import the vehicle trips.');
end;
```

The script to import the Vehicle Trips data is:

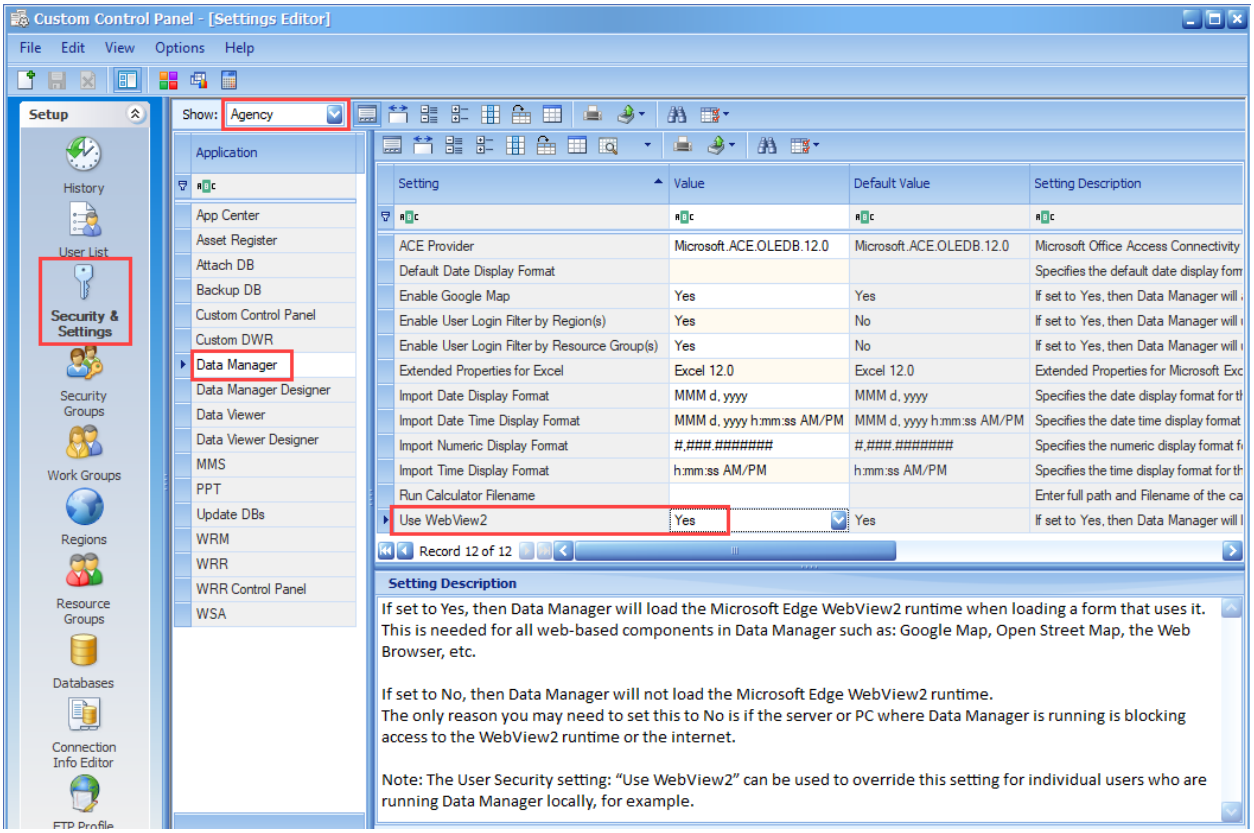
```
procedure btnDownloadGeotabDataClick(Sender: TObject);
var
  intRecordsImported: Int64;
begin
  intRecordsImported := GeotabImport(tblGeotabVehicleTrips.Connection,
                     20210522093838,
                     AppConstants.GeotabImportVehicleTrips, '');

  if (intRecordsImported = 1) then
  begin
    RefreshTable(tblGeotabVehicleTrips);
    ShowMessage('Successfully imported: 1 record.');
  end
  else if (intRecordsImported > 1) then
  begin
    RefreshTable(tblGeotabVehicleTrips);
    ShowMessage('Successfully imported: ' + IntToStr(intRecordsImported) + ' records.');
  end
  else if (intRecordsImported = 0) then
    ShowMessage('No additional records to import.')
  else if (intRecordsImported < 0) then
    ShowMessage('Error: cannot import the vehicle trips.');
end;
```

24. **New Feature:** Added a new Agency setting for Data Manager called: “Use WebView2”.

If set to Yes, then Data Manager will load the Microsoft Edge WebView2 runtime when loading a form that uses it.
This is needed for all web-based components in Data Manager such as: Google Map, Open Street Map, the Web Browser, etc.

If set to No, then Data Manager will not load the Microsoft Edge WebView2 runtime.
The only reason you may need to set this to No is if the server or PC where Data Manager is running is blocking access to the WebView2 runtime or the internet.

Note: The User Security setting: “Use WebView2” can be used to override this setting for individual users who are running Data Manager locally, for example.



25. **New Feature:** Added a new Agency setting for Data Manager called: “Enable Google Map”.

If set to Yes, then Data Manager will allow the Google Map to be loaded.

If set to No, then Data Manager will not allow the Google Map to be loaded.
The only reason for setting this to No is if you want to turn off the Google Map without removing it from the forms.

Note: The setting: “Use WebView2” must also be turned on and the Microsoft Edge WebView2 runtime must be installed for the Google Map to work.

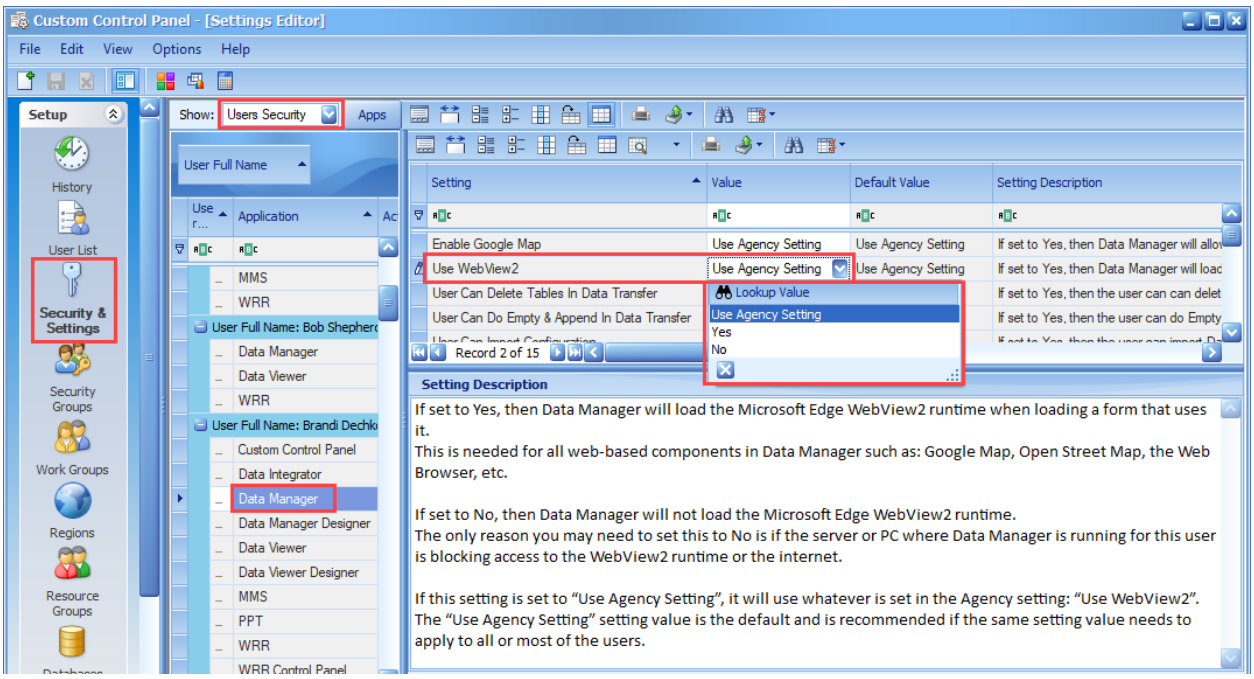
26. **New Feature:** Added a new User setting for Data Manager called: “Use WebView2”.

This setting can be used to override the corresponding Agency setting for individual users who are running Data Manager locally, for example.

If set to Yes, then Data Manager will load the Microsoft Edge WebView2 runtime when loading a form that uses it.
This is needed for all web-based components in Data Manager such as: Google Map, Open Street Map, the Web Browser, etc.

If set to No, then Data Manager will not load the Microsoft Edge WebView2 runtime.
The only reason you may need to set this to No is if the server or PC where Data Manager is running for this user is blocking access to the WebView2 runtime or the internet.

If this setting is set to “Use Agency Setting”, it will use whatever is set in the Agency setting: “Use WebView2”.
The “Use Agency Setting” setting value is the default and is recommended if the same setting value needs to apply to all or most of the users.



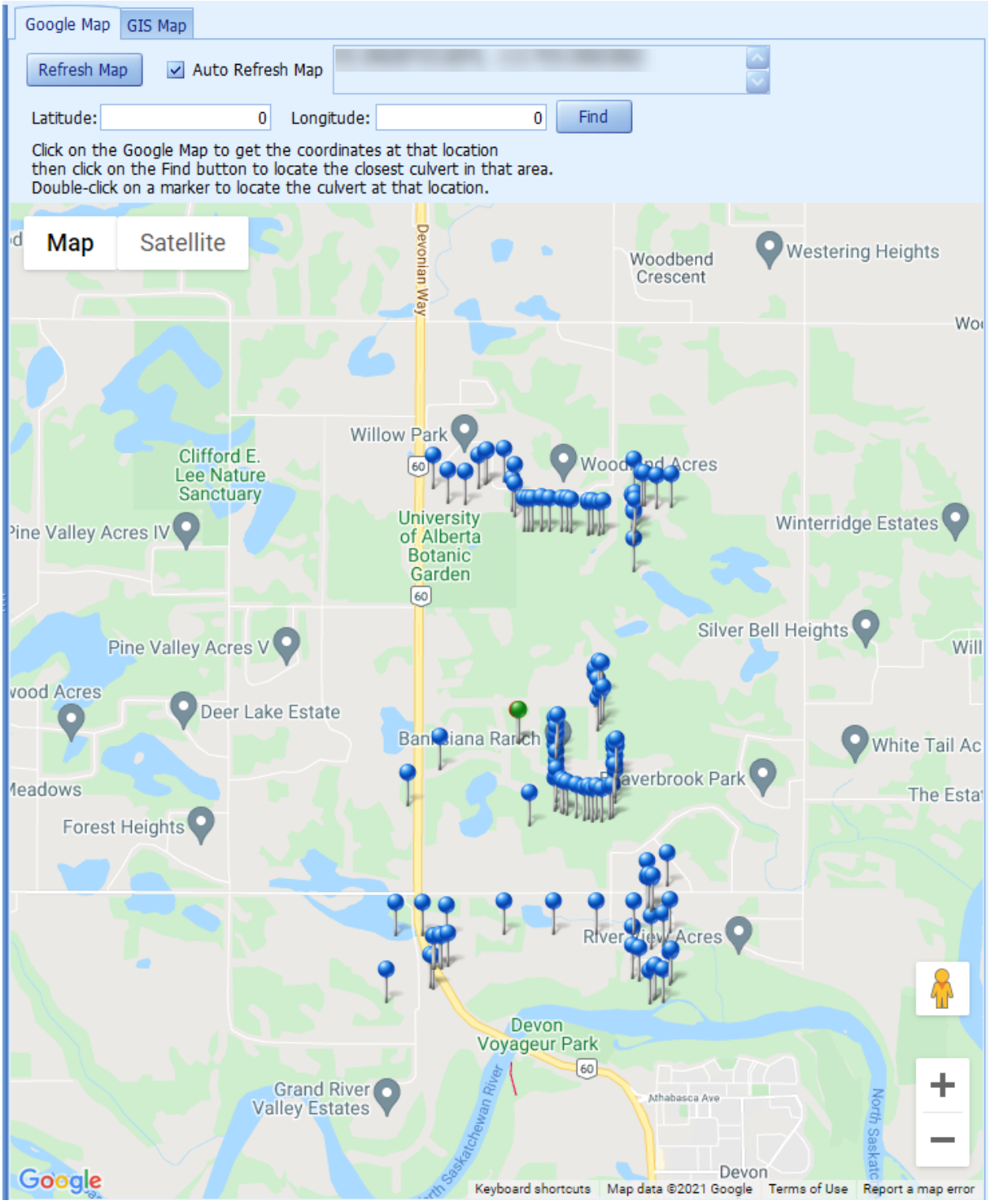
27. **New Feature:** Added a new User setting for Data Manager called: “Enable Google Map”.

If set to Yes, then Data Manager will allow the Google Map to be loaded.

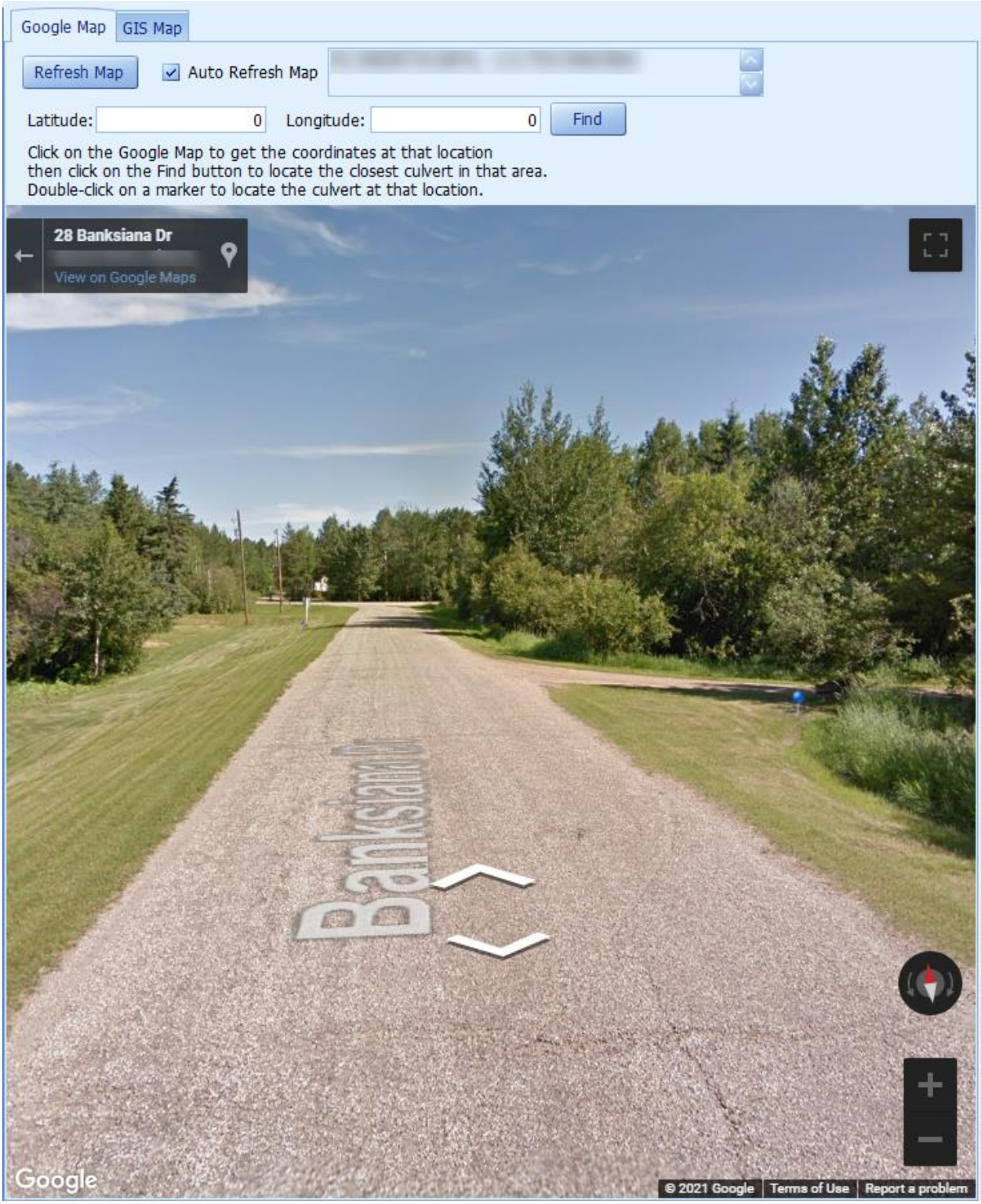
If set to No, then Data Manager will not allow the Google Map to be loaded.
The only reason for setting this to No is if you want to turn off the Google Map without removing it from the forms.

If this setting is set to “Use Agency Setting”, it will use whatever is set in the Agency setting: “Enable Google Map”.
The “Use Agency Setting” setting value is the default and is recommended if the same setting value needs to apply to all or most of the users.

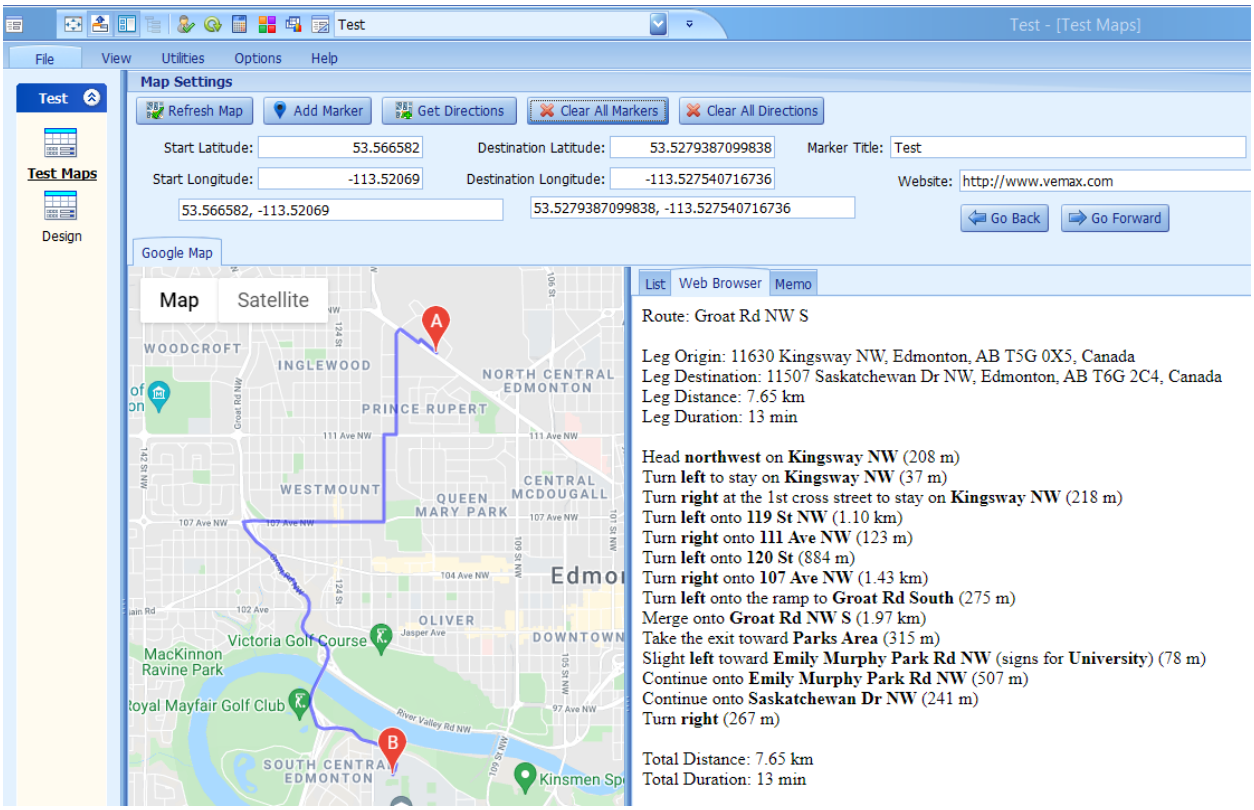
28. **Major Upgrade:** The Google Maps component has been replaced with a much more modern one. Note: A Google Maps API Key is required for this map.



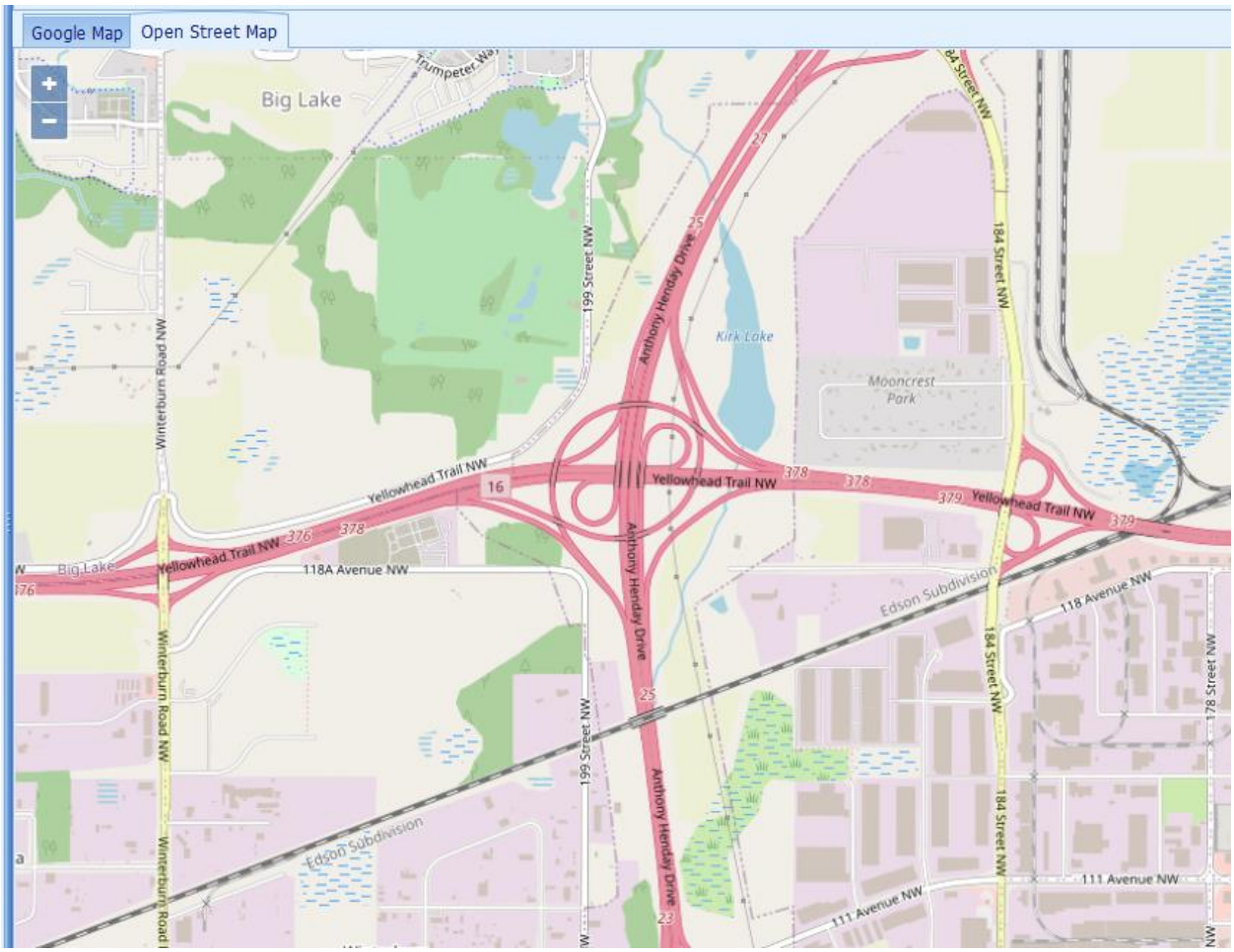
The Google Map supports features such as Street View and Satellite View.



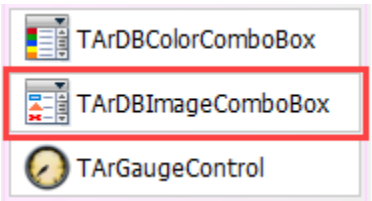
29. **New Feature:** Added the ability to use Google Maps services to get the fastest route from one location to another. We also added the ability to get step-by-step directions, total distance, and total duration of that route.

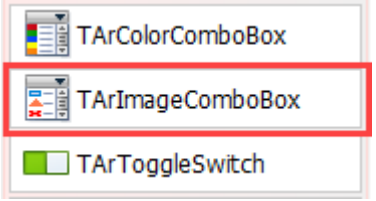


30. **Major New Feature:** Added the Open Street Map component. The Google Map is more advanced and looks nicer however Open Street Map is free to use and does not require an API Key.

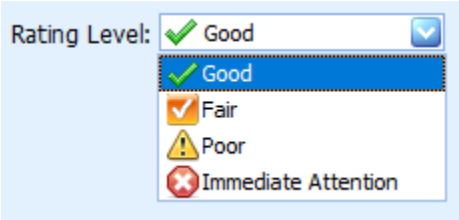


31. **New Feature:** Added a new data-aware control called "TArDBImageComboBox" and a new non-data-aware control called "TArImageComboBox" which are edit controls that display a list of images and text strings within a dropdown window.

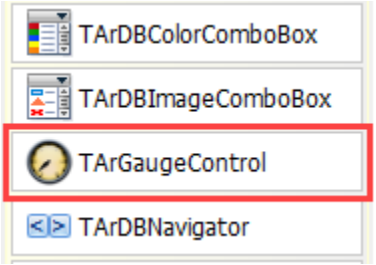




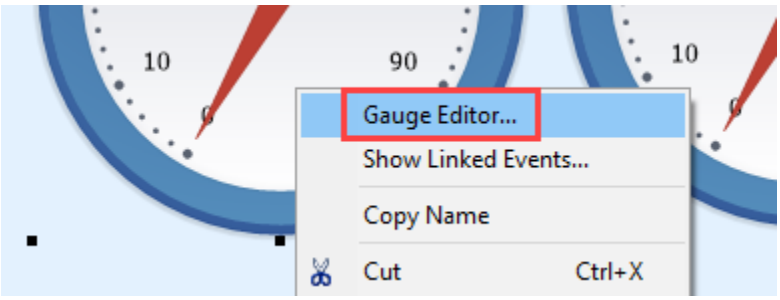
Any 16 x 16 pixel image can be used in the Image List. Here is an example of what the Image Combo boxes look like:



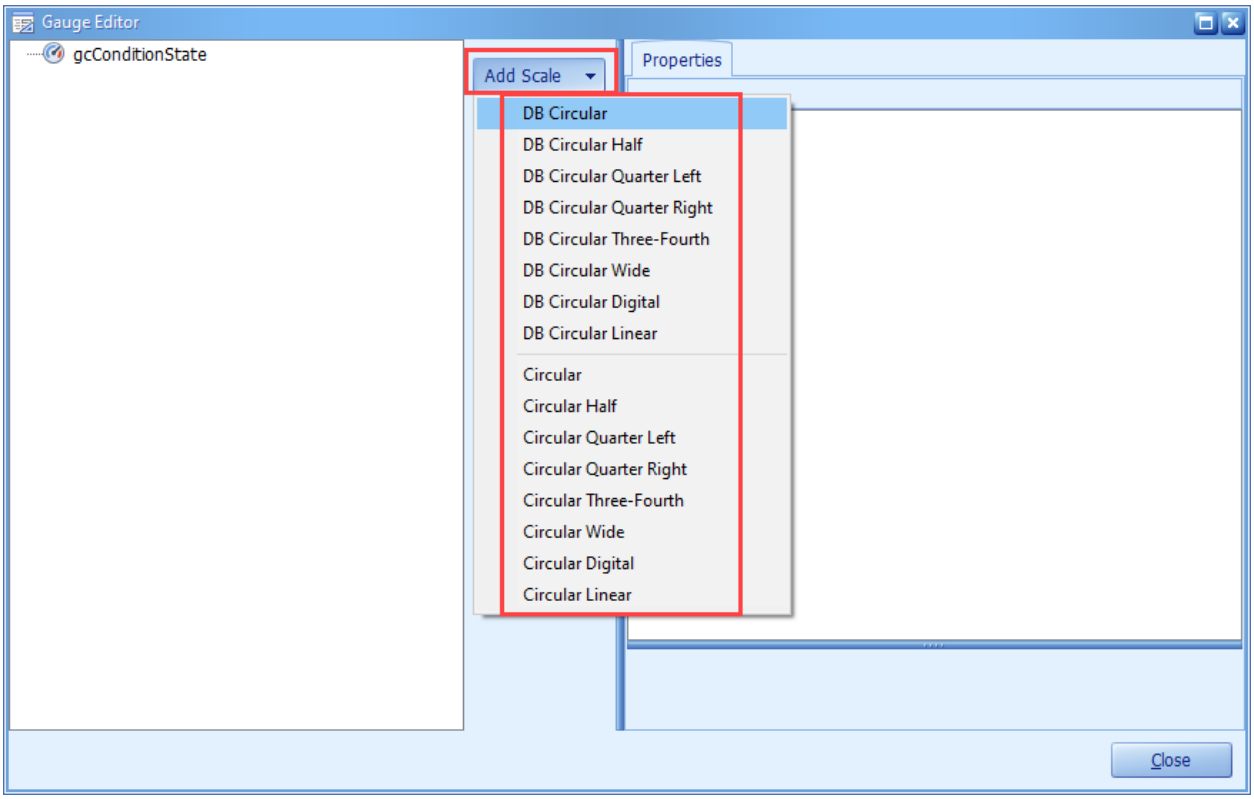
32. **Major New Feature:** Added a gauge control that supports several shapes and styles. The gauge can be data-aware or non-data-aware.



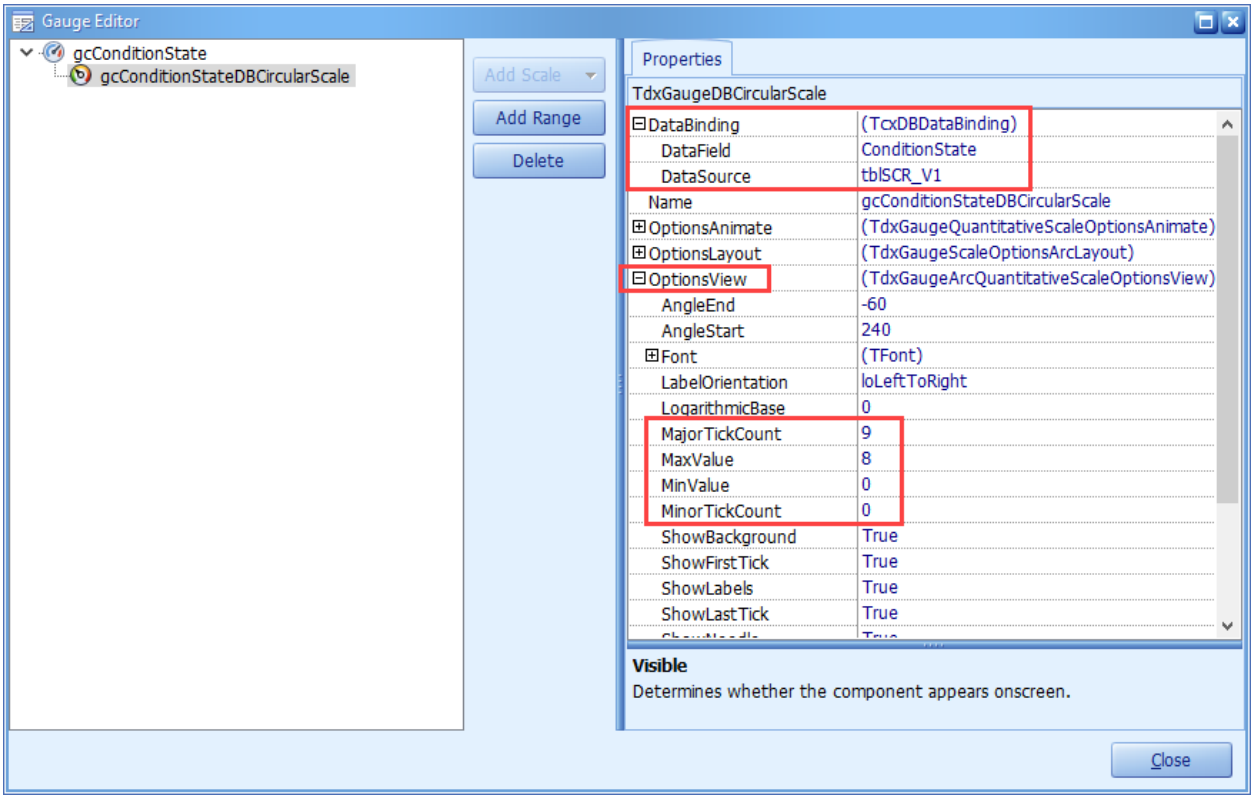
To edit the gauge properties, right-click on the gauge and click the “Gauge Editor” menu item.



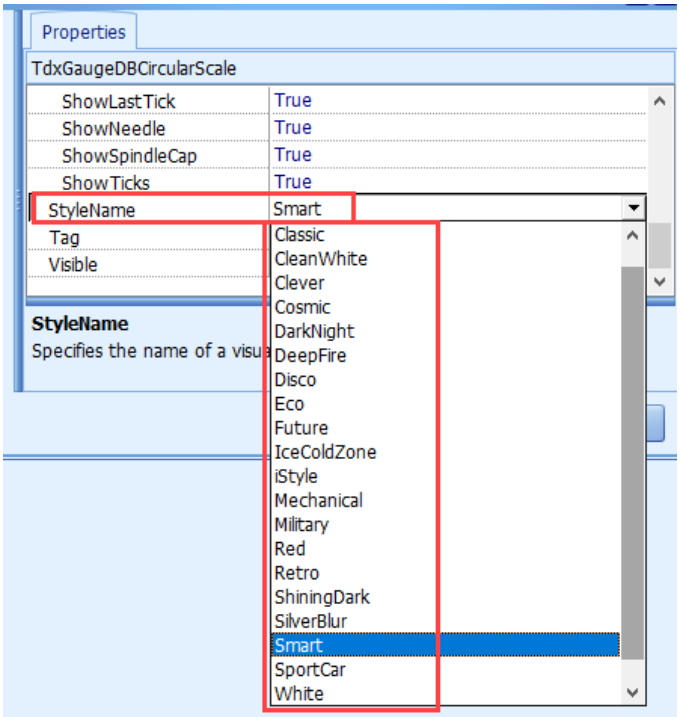
Steps to create a gauge:
In the Gauge Editor dialog, click on the “Add Scale” button.
There are 16 different types of scales. The first 8 scales start with “DB” in the name to indicate that these are data-aware scales. These are the most common scales to use. The remaining 8 scales are non-data-aware, so the value needs to be set in scripting for these scales.



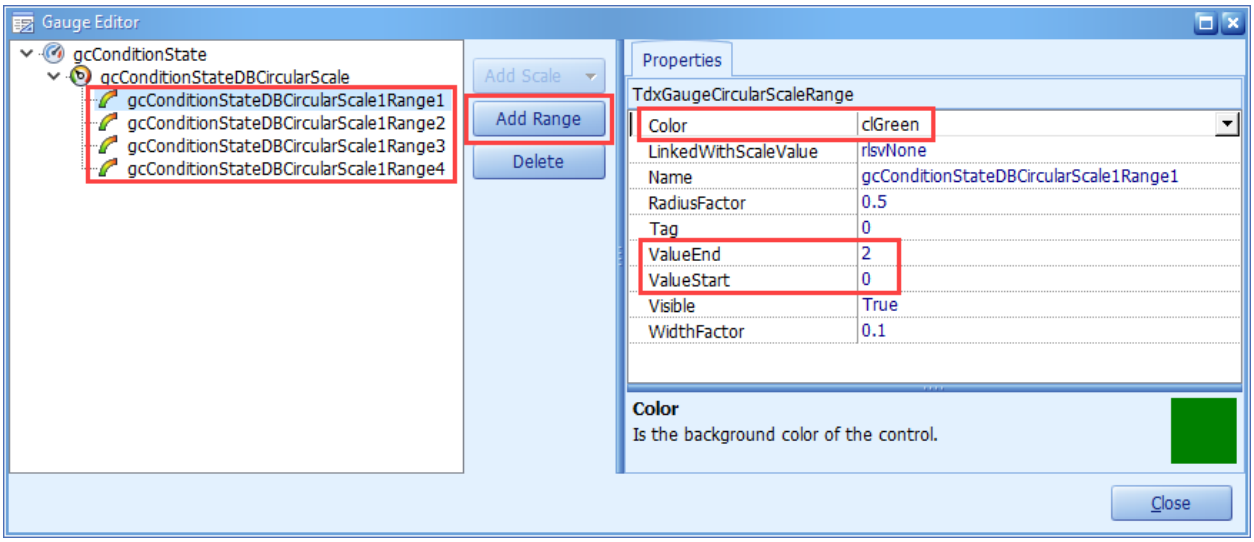
Once the new scale is added, fill in the properties shown below in the red boxes.
For the data-aware scales, select the “DataSource” and “DataField” properties in the “DataBinding” property group.
Expand the “OptionsView” property group then set the “MinValue”, “MaxValue”, “MinorTickCount”, and the “MajorTickCount” properties.
Tip: if your “MinValue” is 0 then your “MajorTickCount” should be 1 more than the “MaxValue”.



To change the look of the gauge, select the “StyleName” property:



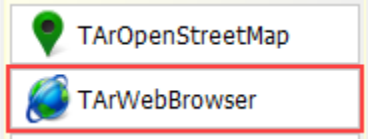
In many cases, you may want to visually identify specific ranges of values along scales in analog gauges.
To add a range, click on the “Add Range” button the set the “Color”, “ValueStart”, and “ValueEnd” properties. You can add as many ranges as needed.

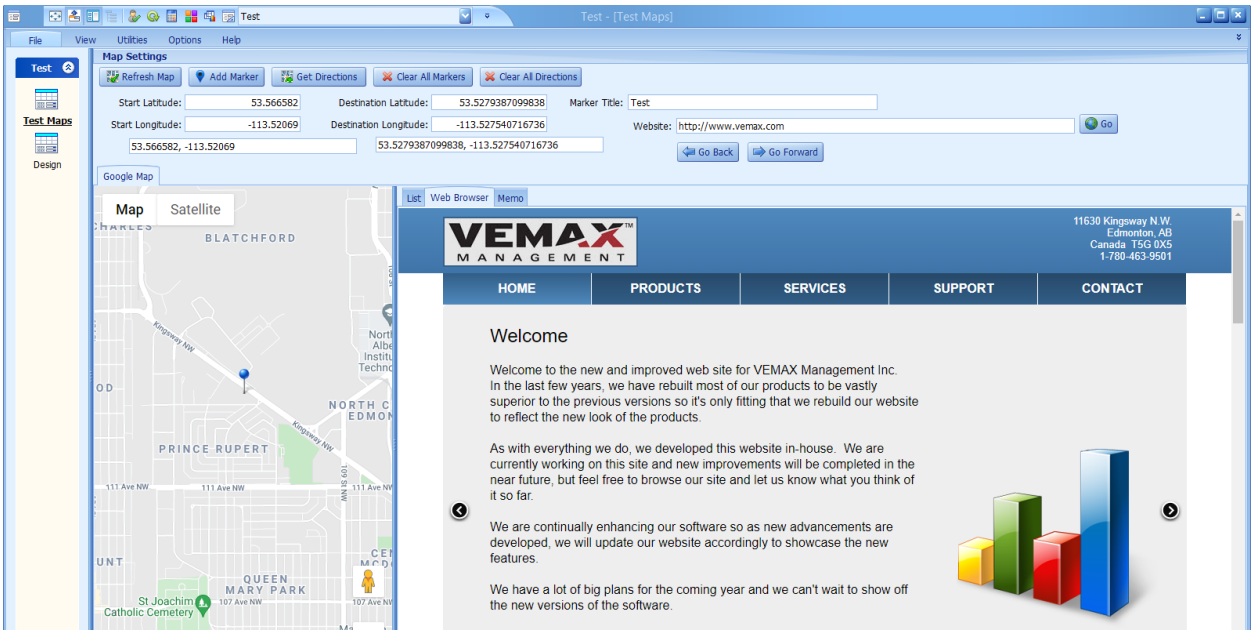


Here is an example of the gauge with the configured properties mentioned above.

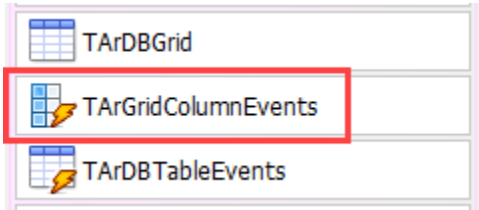
Here is the same gauge but with the “StyleName” property set to “Classic”

33. **Major New Feature:** Added the “TArWebBrowser” component to the “Advanced” tab on the Component Palette. This browser requires the installation of the Edge Chromium (that comes in a Windows 10 Update) or the Microsoft Edge WebView2 runtime.
Download the WebView2 Runtime:
https://developer.microsoft.com/en-us/microsoft-edge/webview2/#download-section

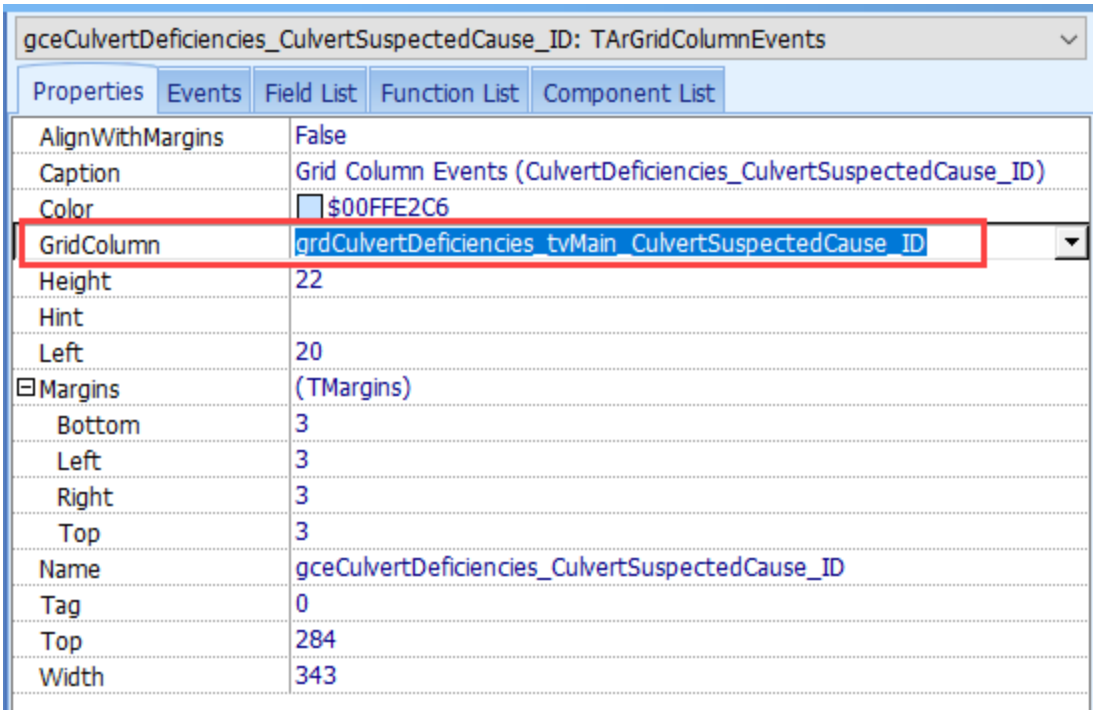




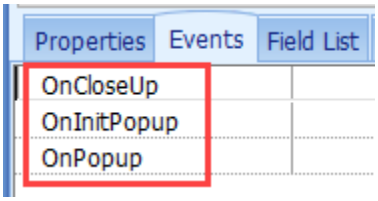
34. **New Feature:** Added a new component "TArGridColumnEvents" to the "Data Controls" tab on the component palette tool bar for setting the grid column events.



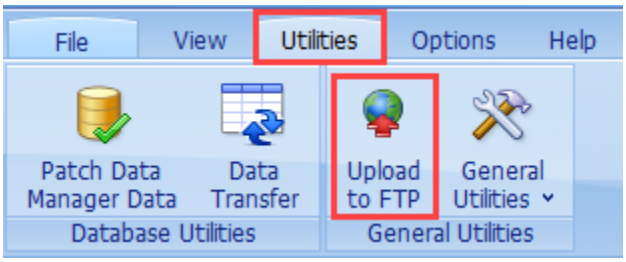
Set the GridColumn property to one of the lookup combo boxes in the grid.

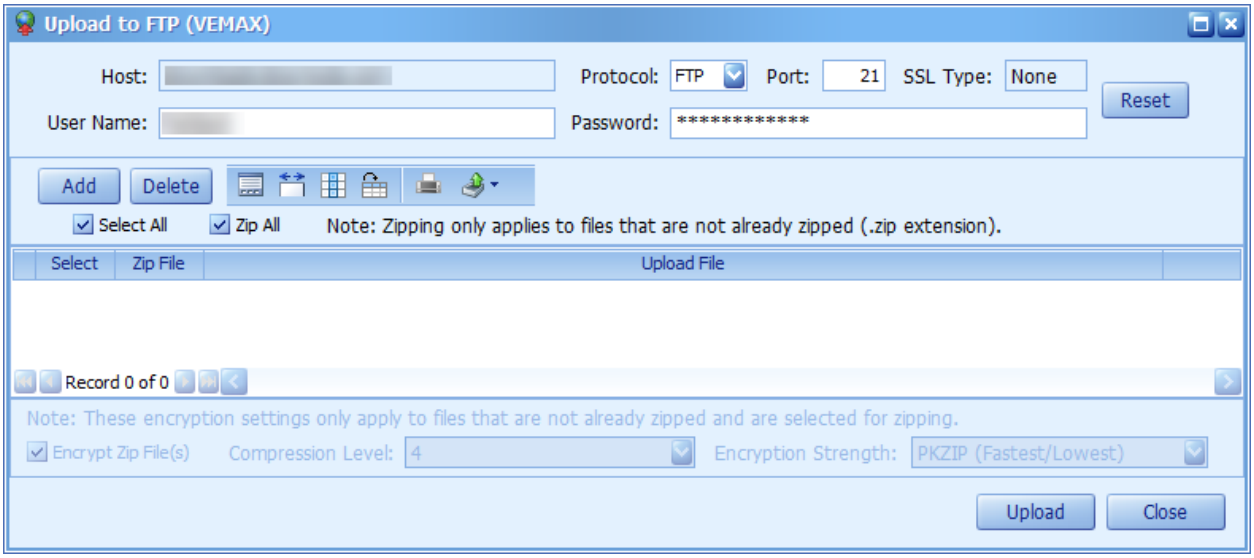


These are the events that are currently supported for lookup combo boxes in the grid.



35. **New Feature:** Added "Upload to FTP" icon to the "Utilities" tab on the Ribbon to open the "Upload to FTP" dialog which allows the user to upload files to a predefined FTP Site such as the VEMAX FTP site. This icon is enabled when the user setting "User Can Upload to FTP" is set to "Yes" in the Custom Control Panel.



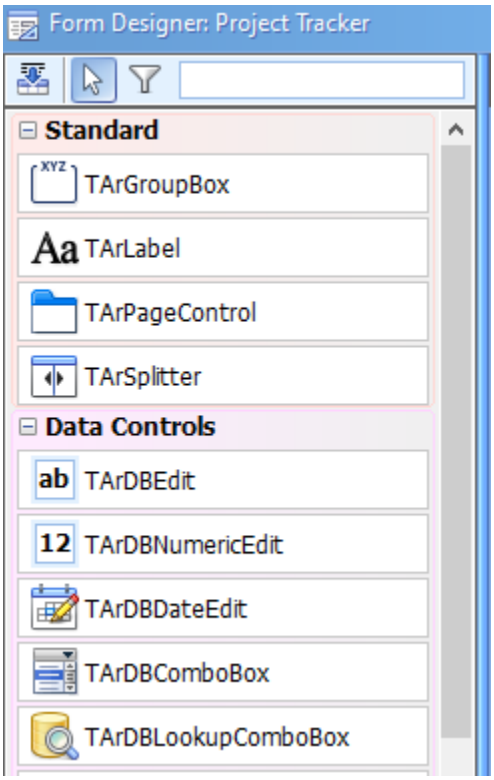


The "Upload to FTP" dialog can also be shown in scripting by calling the follow function:

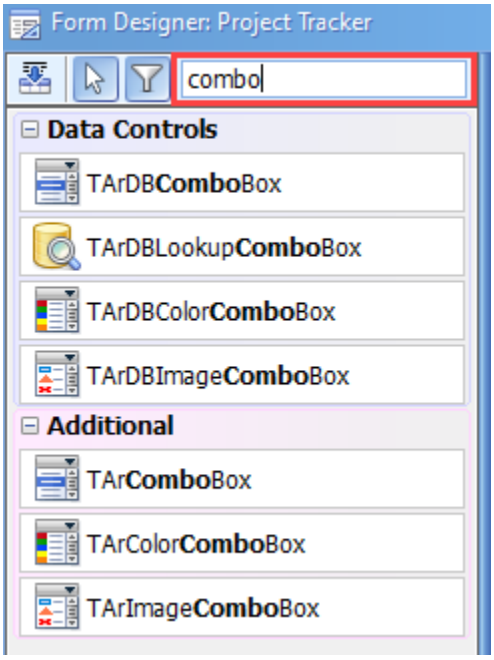
```
strFTPProfile := '';
strUploadFiles := 'D:\Documents\MyFile1.docx, D:\Documents\MyFile2.docx';
bolSilentMode := False;
strErrorMessage := '';

UploadToFTP(strFTPProfile, strUploadFiles, bolSilentMode, strErrorMessage);
```

Description:
strFTPProfile – Specifies the FTP profile name. Set to blank: '' to use the default VEMAX profile.
strUploadFiles – Specifies a list of upload files separated by comma to prefill in the upload files list in the dialog. This is an optional parameter which can be set to blank: ''.
bolSilentMode – Set to True to make the upload process will run in silent mode.
strErrorMessage – Returns the error message if there is one.

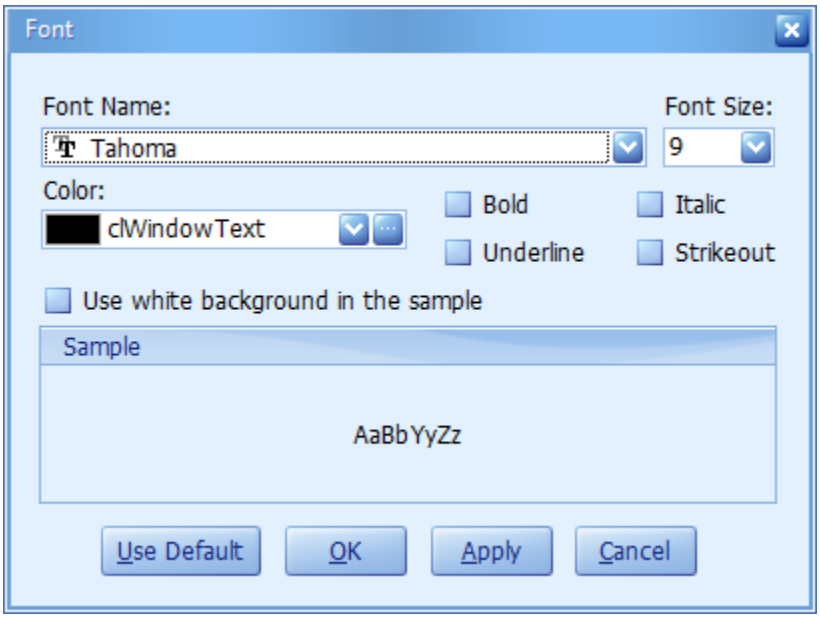
36. **Enhancement:** Replaced all the icons for the components in the Component Palette with newer and nicer looking ones.



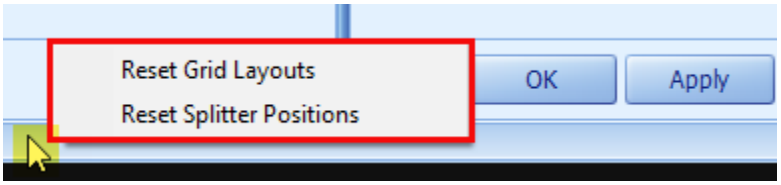
37. **New Feature:** Added a search box to the Component Palette to quickly search for any component name. This feature supports a substring search as shown below and instantly filters the component list as you type.



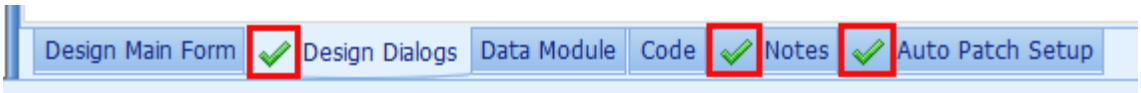
38. **Enhancement:** Redesigned the Font dialog that is used for every component with a Font property. This dialog is more user-friendly than the previous one.



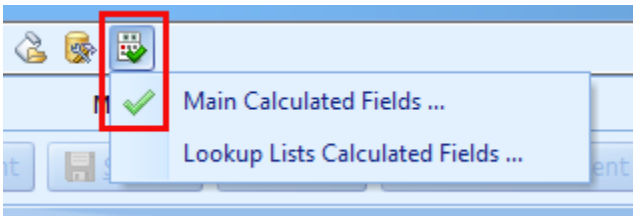
39. **New Feature:** Added a right-click popup menu on the status panel at the bottom of the Form Designer to enable developers to reset the grid layouts and the splitter settings.



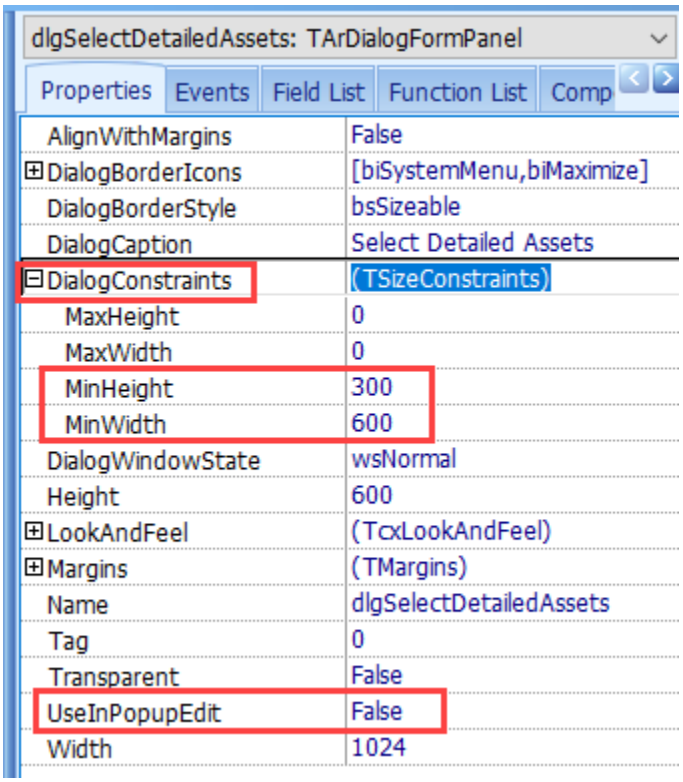
40. **Enhancement:** In Form Designer, a green check mark is shown on the tabs: Custom Dialogs, Notes, and Auto Patch Setup when the corresponding tab has something added.



41. **Enhancement:** In Form Designer and in the Main Connection and Lookup Lists Connection dialogs, a green check mark is shown on the calculated fields button when there are calculated fields added in the Main Connection or Lookup Lists Connection.

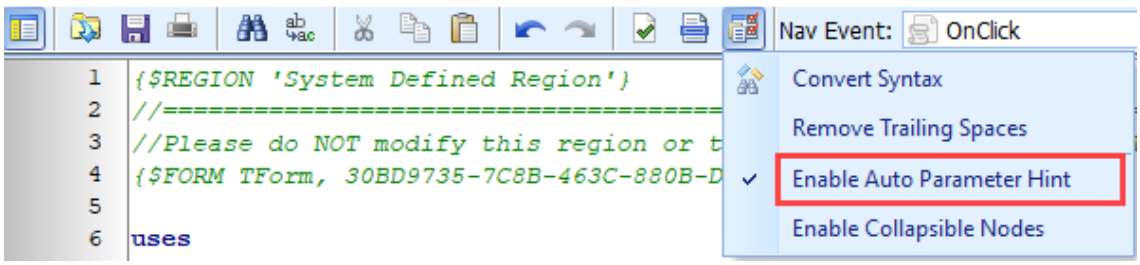


42. **Enhancement:** Added a new property "DialogContraints" to the TArDialogFormPanel component for the dialog boxes. This property is for specifying the dialog's maximum height, minimum height, maximum width, and minimum width. This will enable the developer to prevent the end user from making the dialogs too large or too small.
Also added a new property “UseInPopupEdit” to the TArDialogFormPanel component which indicates to use this control in a popup edit.

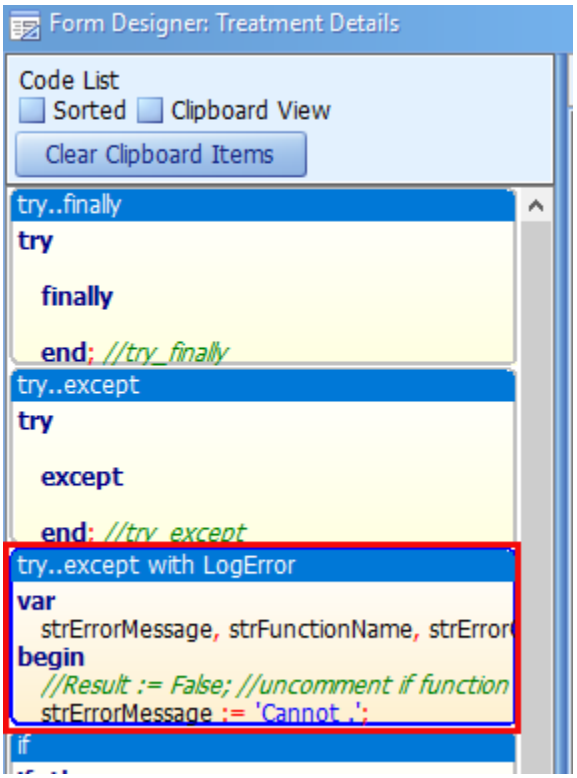


43. **New Feature:** Added a button in the Form Designer to run Data Manager.
This is very handy when you want to test your form. This is more convenient then running Data Manager from the VEMAX App Center.

44. **Enhancement:** Added a new menu item "Enable Auto Parameter Hint" to the “Code Options” popup menu. This option is checked by default. If it is checked, the Parameter Hint will be shown when the developer starts typing the open bracket “(“ after a procedure or function name. This may affect the typing speed in the Code Editor. If this option is not checked, the Parameter Hint will not be shown when typing in the Code Editor. If performance is an issue on your system or if you do not want the hint to display, you can uncheck this menu item.
Also, if the hint is displayed, pressing the Esc. key will hide the hint.



45. **Enhancement:** Added the “try..except with LogError” code segment to the Code List. This is very handy block of code when adding error checking and logging to a function.



46. **New Feature:** Added the following new features to “TDocImage” component:

    New Properties:

    ImageComponent - Specifies the image component on the form for displaying the image on the current record.
    StoreThumbnail - Determines whether to store thumbnail image to the database. If the DefaultOperation property is set to opStoreInDatabase, it will store full size image instead of thumbnail image.
    ThumbnailSize - Specifies the thumbnail size for storing to the database.

    New Procedures:

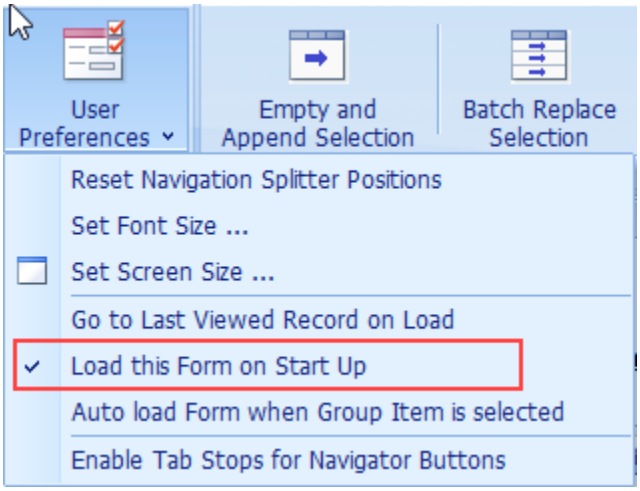
    RefreshThumbnailImage – Refreshes the thumbnail image for the current record in the grid.
    RefreshAllThumbnailImages – Refreshes the thumbnail images for all the records in the grid.

    The user-defined fields added to the DocImage table are now available in both Documents and Images grids.

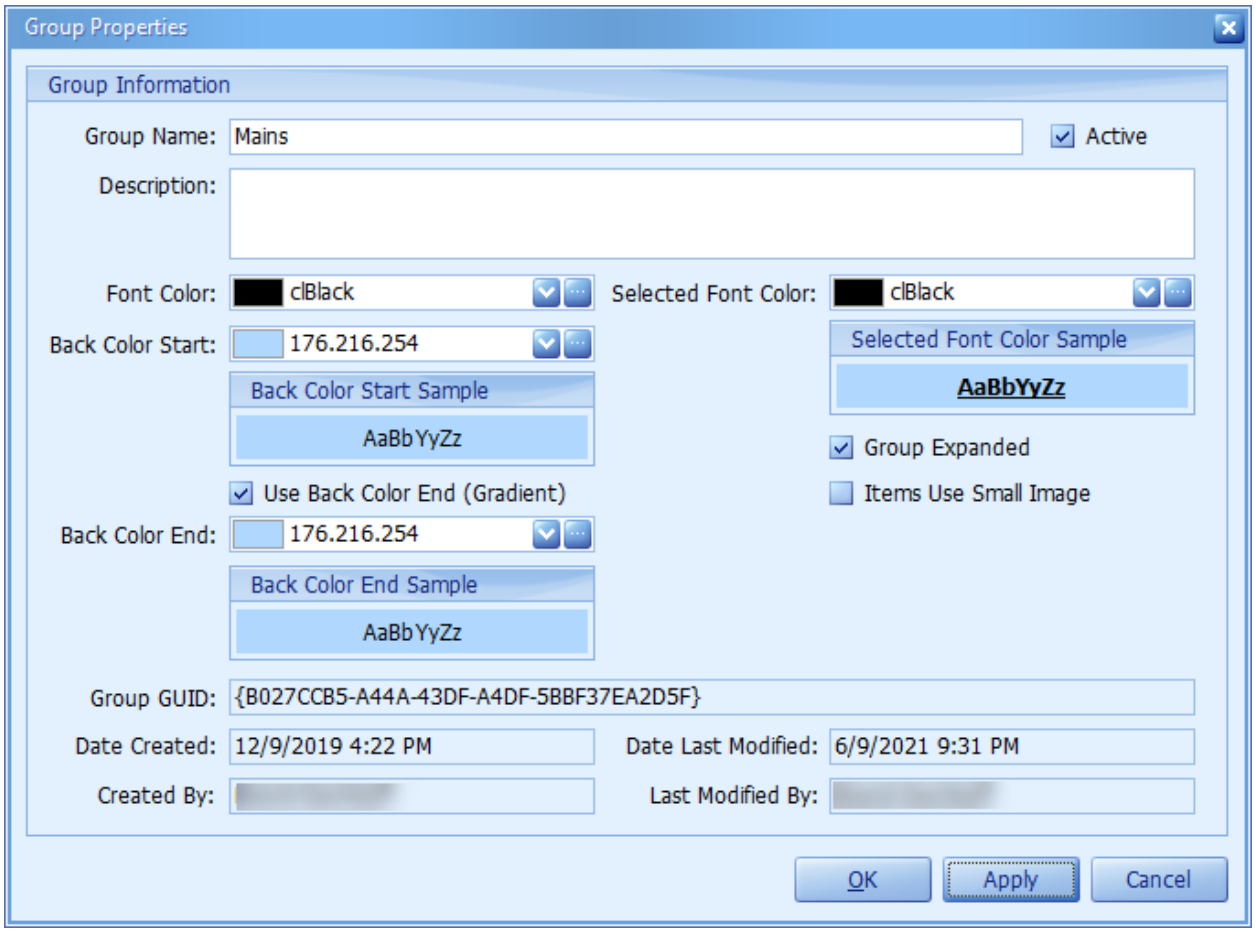
    Added “Documents Grid Columns” and “Images Grid Columns” pop up menus to the “TDocImage” component for customizing the grid columns.

47. **New Feature:** Added a new pop-up menu “Open Form Designer (READ-ONLY)” to the navigator for opening the Form Designer in Read Only mode.

48. **New Feature:** Added a new menu “Load this Form on Start Up” to the “User Preferences” button for setting to load the current form on application start up in Data Manager.



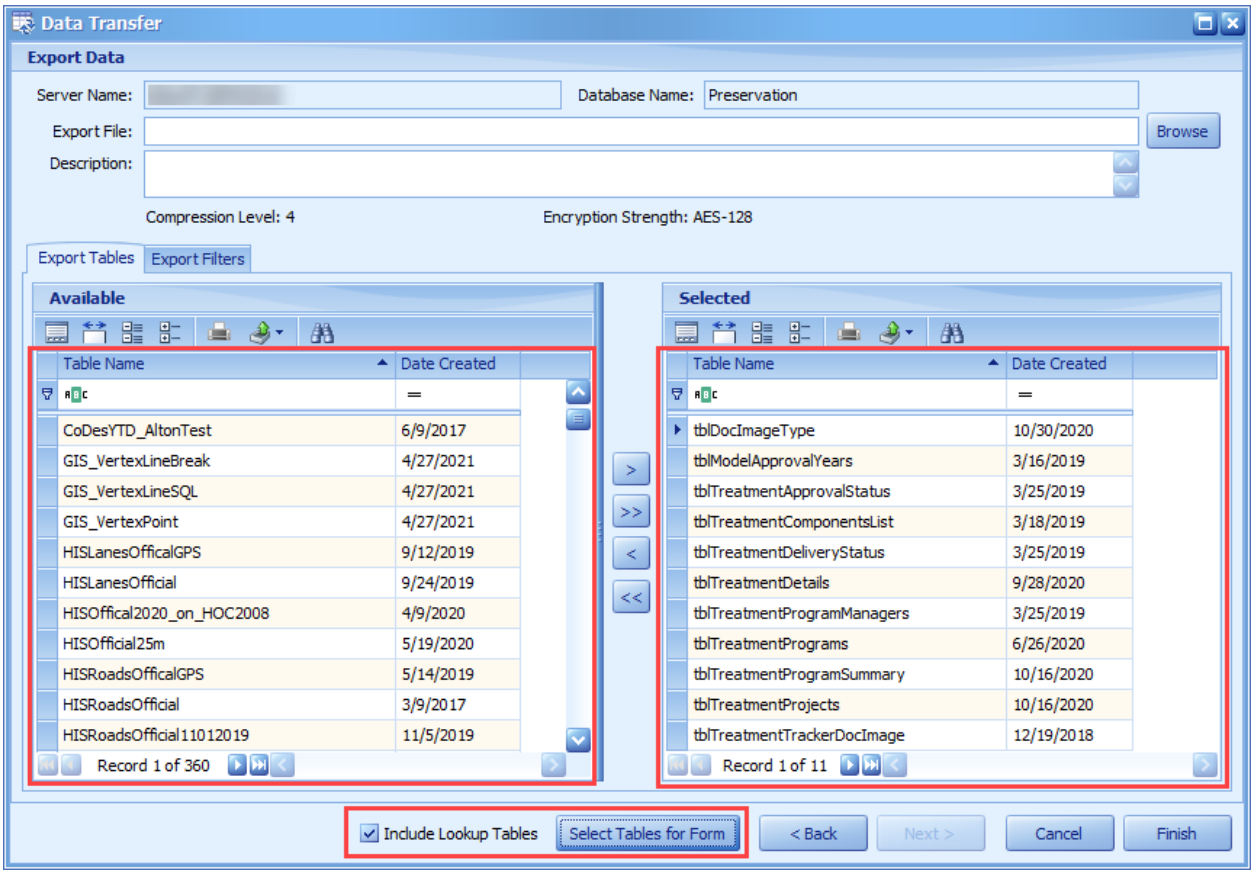
49. **Enhancement:** In the Group Properties dialog, added “Selected Font Color” pick list for specifying the font color for the selected group item and added “Use Back Color End (Gradient)” check box for turning on or off the “Back Color End”.



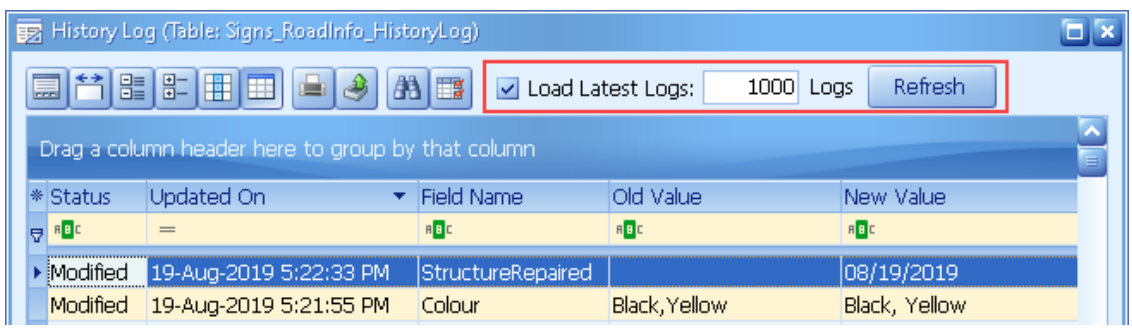
50. **Enhancement:** Modified the “Save Group Item As” dialog to name the node name for the new group item the same as the group item name and select the newly added group item in the sidebar.

51. **Enhancement:** Improved the Data Transfer utility to allow importing the data from the source table which contains a table schema other than “dbo”. Also, improved support for geospatial fields.

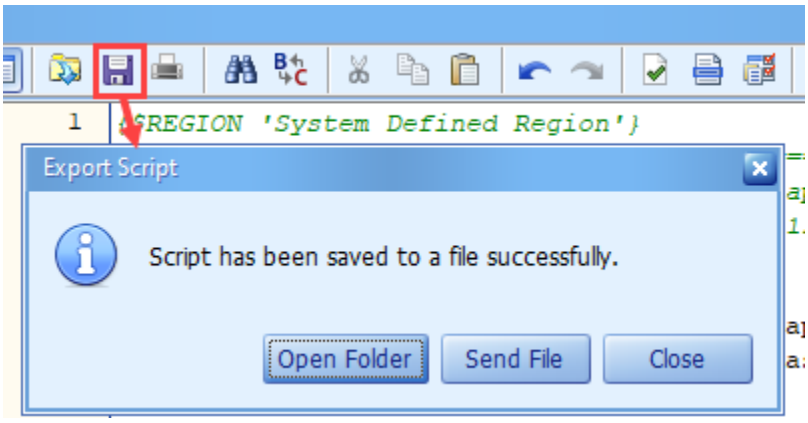
52. **Enhancement:** In Data Transfer, changed the list boxes on the Export Tables tab to grids with searching and sorting capabilities. Also, added the "Select Tables for Form" button for populating the Selected grid with the tables in the main and lookup lists connections that are in the selected database. This button is only active if a form is selected when the Data Transfer button is pressed.



53. **Enhancement:** Added options to the History Log dialog to allow loading the number of latest logs to the grid instead of loading all logs.



54. **Enhancement:** In the Save script button in the Code tab in the Form Designer, added the message asking to Open Folder and Send File after the script has been saved to a file successfully.



55. **New Functions:** Added the function: **GetDateStamp** that returns a Date (or Date/Time) stamp string of the given date and separator. This is useful when adding a date/time stamp to a filename to ensure it is always unique, for example.

    GetDateStamp(dtDateTime: TDateTime; strSeparator: String; bolIncludeTime: Boolean): String;

    The format when bolIncludeTime is True: <Year> strSeparator <Month> strSeparator <Day> strSeparator <Hour> strSeparator <Minute> strSeparator <Second>

    Example:
    strDateStamp := GetDateStamp(Now, '_', True);

    For April 14, 2021, at 15:30:43, the results of the above example would look like this:
    2021_04_14_15_30_43

    Added the function: **CenterControlHoriz** to center the given control horizontally in its parent control. Procedure declaration: CenterControlHoriz(Control: TWinControl);

    Example:
    CenterControlHoriz(pnlButtons);

    Added the function: **CenterControlVert** to center the given control vertically in its parent control. Procedure declaration: CenterControlVert (Control: TWinControl);

    Example:
    CenterControlVert(pnlButtons);

56. **Enhancement:** Added SRID as an optional parameter to the following functions:

    ConvertLatLongToUTM(extLat, extLong, extNorthing, extEasting, strZone, intSRID);
    ConvertUTMToLatLong(extNorthing, extEasting, strZone, extLat, extLong, intSRID);

    These functions convert GIS UTM Northing and Easting to/from Latitude and Longitude. If the optional parameter intSRID is not defined, the SRID value 4326 (Datum WGS 84) will be used. Check this Spatial Reference web site for a list of SRIDs:
    https://spatialreference.org/ref/epsg/

57. **Enhancement:** Added the following functions:

Function Name	Function Declaration	Description
BoolToStrTF	BoolToStrTF(bolValue: Boolean): String;	Returns a string of: "True" if the bo is False.
BoolToStrYN	BoolToStrYN(bolValue: Boolean): String;	Returns a string of: "Yes" if the bol False.
CenterControlHoriz	CenterControlHoriz(Control: TWinControl);	Centers the given control horizonta
CenterControlVert	CenterControlVert(Control: TWinControl);	Centers the given control vertically
DataTransferExportV2	DataTransferExportV2(cnData: TMSConnection; strExportFileName: String; strDescription: String; strExportTables: String; bolShowCompletedMessage: Boolean; bolSilentMode: Boolean): TDMStatus;	Exports data for a list of tables to a ".zip". Set the parameter strExportTables " ## ". Table filtering is also supported in t after the table name, then add the Return TDMStatus values are: dmS Example: strExportTables := '[tblTable1] ## ## [tblTable3] ++ [AnotherField]
DataTransferImportV2	DataTransferImportV2(cnData: TMSConnection; strImportFileName: String; impImportMode: TImportMode; bolDeleteSourceFile: Boolean; bolShowCompletedMessage: Boolean; bolSilentMode: Boolean; strVerifyTableName: String; strVerifyDatabaseName: String; bolKeepNewRecords: Boolean; strDateLastModifiedFieldName: String): TDMStatus;	Imports data from a Data Transfer ".zip". Set the parameter impImportMode impEmptyAppend. The optional parameters strVerifyT are for verifying if the import file co database name. The optional parameters bolKeepN strDateLastModifiedFieldName are destination table by comparing the The parameter bolKeepNewRecord specified. The parameter strDateLastModified "DateLastModified" if it is not speci Return TDMStatus values are: dmS
DayOfTheWeekAbbrevName	DayOfTheWeekAbbrevName(const AValue: TDateTime; bolIncludePeriod: Boolean): String;	Returns the abbreviated name of th specified TDateTime value. The first parameter is the specified indicates whether to include the pe Return value examples are: "Mon"
DayOfTheWeekName	DayOfTheWeekName(const AValue: TDateTime): String;	Returns the full name of the day of TDateTime value.Return value exa
GeotabImport	function GeotabImport(cnData: TMSConnection; intGeotabConInfo_ID: Int64; strOperation: String; strAdditionalParams: String): Int64;	Imports vehicle information such as Geotab. strOperation: AppConstants.Geotab AppConstants.GeotabImportVehicle strAdditionalParams: For passing o default Return Value: < 0: Failed; = 0: No
GetDateStamp	GetDateStamp(dtDateTime: TDateTime; strSeparator: String; bolIncludeTime: Boolean): String;	Returns a Date (or Date/Time) star separator. The format when bolIncludeTime is strSeparator <Day> strSeparator < strSeparator <Second> The format when bolIncludeTime is strSeparator <Day> Example return value: 2019_03_1

GetUserCellPhoneByID	GetUserCellPhoneByID(intUserID: Int64): String;	Returns the user cell phone by Use
GetUserDepartmentByID	GetUserDepartmentByID(intUserID: Int64): String;	Returns the user department by Us
GetUserEmailAddressByID	GetUserEmailAddressByID(intUserID: Int64): String;	Returns the user email address by
GetUserFirstNameByID	GetUserFirstNameByID(intUserID: Int64): String;	Returns the user first name by Use
GetUserFullNameByID	GetUserFullNameByID(intUserID: Int64): String;	Returns the user full name by User
GetUserLastNameByID	GetUserLastNameByID(intUserID: Int64): String;	Returns the user last name by User
GetUserNameByID	GetUserNameByID(intUserID: Int64): String;	Returns the user name by User ID.
GetUserOfficePhoneByID	GetUserOfficePhoneByID(intUserID: Int64): String;	Returns the user office phone by U
GetUserSecurityGroupIDByID	GetUserSecurityGroupIDByID(intUserID: Int64): Int64;	Returns the user security group ID
GetUserSecurityGroupNameByID	GetUserSecurityGroupNameByID(intUserID: Int64): String;	Returns the user security group na
GetUserSupervisorIDByID	GetUserSupervisorIDByID(intUserID: Int64): Int64;	Returns the user supervisor ID by U
GetUserSupervisorNameByID	GetUserSupervisorNameByID(intUserID: Int64): String;	Returns the user supervisor name b
GetUserWorkGroupIDByID	GetUserWorkGroupIDByID(intUserID: Int64): Int64;	Returns the user work group ID by
GetUserWorkGroupNameByID	GetUserWorkGroupNameByID(intUserID: Int64): String;	Returns the user work group name
GotoForm	GotoForm(strConfigName, strGroupName, strGroupItemName, strFormName: String);	Go to a specific Form.
GotoDataView	GotoDataView(strConfig, strGroup, strDataView: String): Boolean;	Go to the specified Config, Group, a it. Returns True if successful and Fa have access to the Data View.
MessageDlg Confirmation More Info	MessageDlg(strMessage: String; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint; strMoreInfo, strSystemMessage: String; intMessageCode: Integer): Integer;	Displays a Confirmation message d more information.
MessageDlg Error More Info	MessageDlg(strMessage: String; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint; strMoreInfo, strSystemMessage: String; intMessageCode: Integer): Integer;	Displays an Error message dialog b information.
MessageDlg Info More Info	MessageDlg(strMessage: String; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint; strMoreInfo, strSystemMessage: String; intMessageCode: Integer): Integer;	Displays an Information message d more information.
MessageDlg Warning More Info	MessageDlg(strMessage: String; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint; strMoreInfo, strSystemMessage: String; intMessageCode: Integer): Integer;	Displays a Warning message dialog more information.
MonthOfTheYearAbbrevName	MonthOfTheYearAbbrevName(const AValue: TDateTime; bolIncludePeriod: Boolean): String;	Returns the abbreviated name of th specified TDateTime value. The first parameter is the specified indicates whether to include the pe Return value examples are: "Jan" o
MonthOfTheYearName	MonthOfTheYearName(const AValue: TDateTime): String;	Returns the full name of the month specified TDateTime value. Return value examples are: "Janua
SentenceCase	SentenceCase(strMyString: String): String;	Returns a string where the first cha and all other characters are lower c
StrToBool	StrToBool(strValue: String): Boolean;	Returns True if the strValue is "Yes returns False. The strValue is not case-sensitive.
TitleCase	TitleCase(strMyString: String): String;	Returns a string where the first cha capitalized and all other characters

UploadToFTP	UploadToFTP(strFTPProfile: string; strUploadFiles: string; bolSilentMode: Boolean; var strErrorMessage: string): TDMStatus;	Shows the "Upload to FTP" dialog f FTP Site or to a user-defined FTP S strFTPProfile – Specifies the FTP pr default VEMAX profile. strUploadFiles – Specifies a list of u include in the upload files list in the which can be set to blank. bolSilentMode – Indicates that the mode. strErrorMessage – Specifies this pa there is one. Return TDMStatus values are: dmS

58. **Enhancement:** Added following table structure columns for all the tables available to the form: Field Order, Data Type, Char Length, Numeric Size and Numeric Decimals to the Fields List in the Form Designer. This is very handy information to have in the Form Designer. Also, since this grid can be printed and/or exported, it makes it easy to document table structure for these tables.

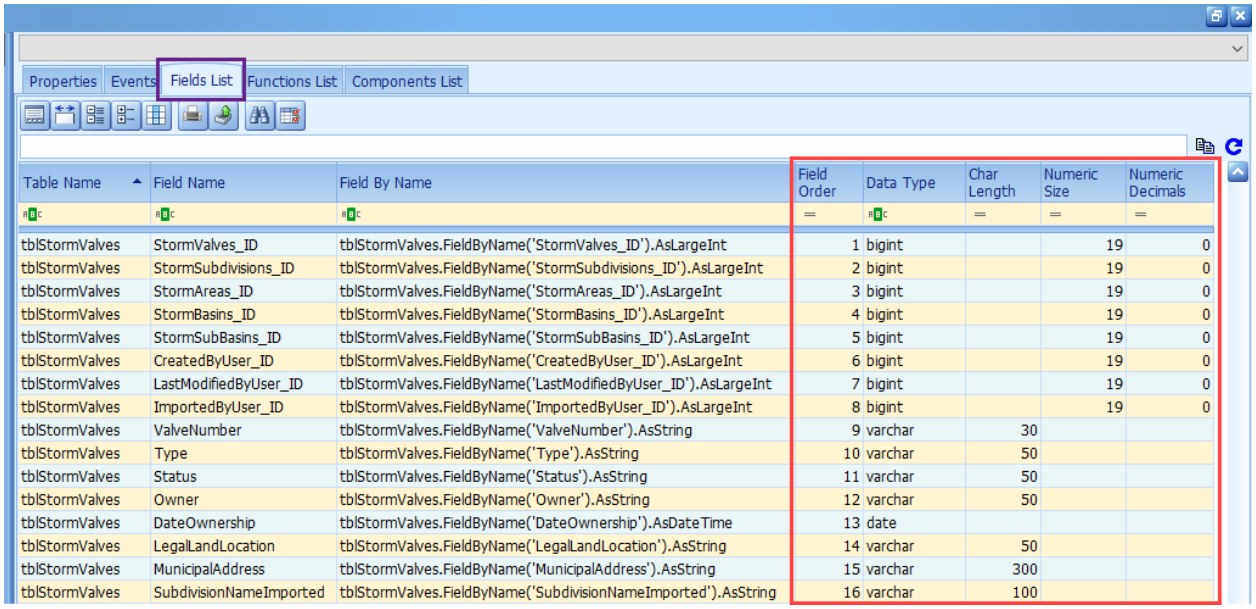


Table Name	Field Name	Field By Name	Field Order	Data Type	Char Length	Numeric Size	Numeric Decimals
tblStormValves	StormValves_ID	tblStormValves.FieldByName('StormValves_ID').AsLargeInt	1	bigint		19	0
tblStormValves	StormSubdivisions_ID	tblStormValves.FieldByName('StormSubdivisions_ID').AsLargeInt	2	bigint		19	0
tblStormValves	StormAreas_ID	tblStormValves.FieldByName('StormAreas_ID').AsLargeInt	3	bigint		19	0
tblStormValves	StormBasins_ID	tblStormValves.FieldByName('StormBasins_ID').AsLargeInt	4	bigint		19	0
tblStormValves	StormSubBasins_ID	tblStormValves.FieldByName('StormSubBasins_ID').AsLargeInt	5	bigint		19	0
tblStormValves	CreatedByUser_ID	tblStormValves.FieldByName('CreatedByUser_ID').AsLargeInt	6	bigint		19	0
tblStormValves	LastModifiedByUser_ID	tblStormValves.FieldByName('LastModifiedByUser_ID').AsLargeInt	7	bigint		19	0
tblStormValves	ImportedByUser_ID	tblStormValves.FieldByName('ImportedByUser_ID').AsLargeInt	8	bigint		19	0
tblStormValves	ValveNumber	tblStormValves.FieldByName('ValveNumber').AsString	9	varchar	30		
tblStormValves	Type	tblStormValves.FieldByName('Type').AsString	10	varchar	50		
tblStormValves	Status	tblStormValves.FieldByName('Status').AsString	11	varchar	50		
tblStormValves	Owner	tblStormValves.FieldByName('Owner').AsString	12	varchar	50		
tblStormValves	DateOwnership	tblStormValves.FieldByName('DateOwnership').AsDateTime	13	date			
tblStormValves	LegalLandLocation	tblStormValves.FieldByName('LegalLandLocation').AsString	14	varchar	50		
tblStormValves	MunicipalAddress	tblStormValves.FieldByName('MunicipalAddress').AsString	15	varchar	300		
tblStormValves	SubdivisionNameImported	tblStormValves.FieldByName('SubdivisionNameImported').AsString	16	varchar	100		

59. **Enhancement:** Added the following properties to the AppInfo object:
    - ConfigName
    - ConfigVersion
    - DBServerVersion
    - DBServerVersionDescription
    - DBServerVersionNumber
    - FormName
    - GroupItemName
    - GroupItemVersion
    - SystemDatabaseName

60. **Enhancement:** Added the "FileName" property to TArCalculatorButton control. Set this property to run user-defined calculator application instead of the default Windows Calculator.

61. **Enhancement:** Changed the Run Calculator button to support running custom Calculator application instead of the standard Windows Calculator. Set the "Run Calculator Filename" agency setting in Custom Control Panel to full path and Filename of the calculator application to be used with the Run Calculator button. Leave this setting blank to use the standard Windows Calculator.

62. **Enhancement:** Added validation checking on Data Manager start up to check if the connected Server Name is the same as in the "Vemax.GlobalInfo.dat" file. If they are not the same, a warning message is shown to let the user know how to fix the problem.

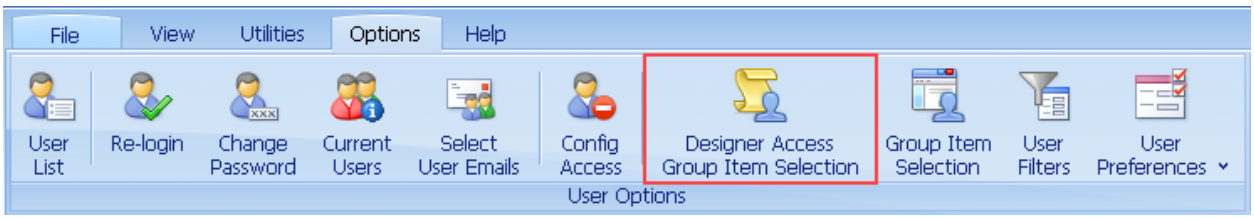
63. **Enhancement:** Added "OnAutoSizeButtonClick" event to TGridToolbar component which occurs when the user clicks the Auto Size button. This event is for customizing the column sizes after the Auto Size button is clicked.

64. **Modification:** For new dialogs in the Form Designer, the biMinimize property is to False by default for the TArDialogFormPanel control.

65. **Enhancement:** Changed the Details button on the About dialog when it is pressed to display a dialog like the one from the Data Viewer's About dialog.

66. **Enhancement:** Added new settings for Maintain Database:
    - User Can Run Maintain Database
    - User Can Delete Tables
    - User Can Delete SQL Views

67. **Enhancement:** Added a new button "Designer Access Group Item Selection" to the ribbon. This button is enabled basing on the setting "User Can Select Designer Access Group Item". Press this button to show the "Designer Access Group Item Selection" dialog for selecting the forms to give Data Manager Designer users designer access. This is a way of giving users access to Data Manager Designer but restricting what forms that can modify.



68. **Enhancement:** Added menu item "Show Empty Groups" to the "User Preferences" popup menu. If it is checked, Data Manager Designer will show the empty groups in the sidebar in Design mode.

69. **Enhancement:** Added a new Navigator Event called: OnLoadFromDataViewer that is triggered every time the form is being loaded (or data is being located) via Data Viewer (i.e., when ever the Edit button in Data Viewer is pressed). This is handy if special processing is needed in this situation.

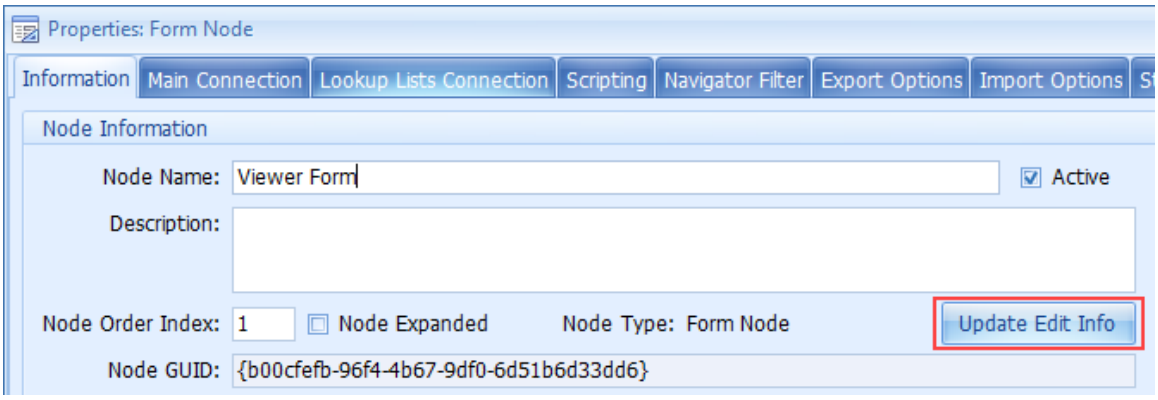
70. **Enhancement:** Added a new property to the AppInfo called: IsLoadingFromDataViewer. It is set to True when the form is loaded via the Edit button in Data Viewer.

71. **Issue:** When the popup menu item "Reset Current Grid Layout" is pressed for the grid on a dialog box, it may cause an access violation error after the dialog box is closed.
    Status: This issue has been resolved.

72. **Enhancement:** Modified the Send button on the message dialog which is shown when successfully export data to use more modern SMTP protocols as defined in the Email profile in the Custom Control Panel. When the Send button is clicked, it will show the following dialog:

73. **Enhancement:** Added support to single row drag & drop in Maintain Database to move the field order. The drag & drop is enabled the same way as the Up & Down buttons.

74. **Enhancement:** Data Manager Designer will automatically update the "Config", "Group", and "Group Item" fields in the Edit Info in Data Viewer Designer for all applicable Data Views when the Group Item is moved to a different Group or Config, or when the Form is moved to a different Group Item. Also added the "Update Edit Info" button to the "Node Properties" dialog to do the same update.



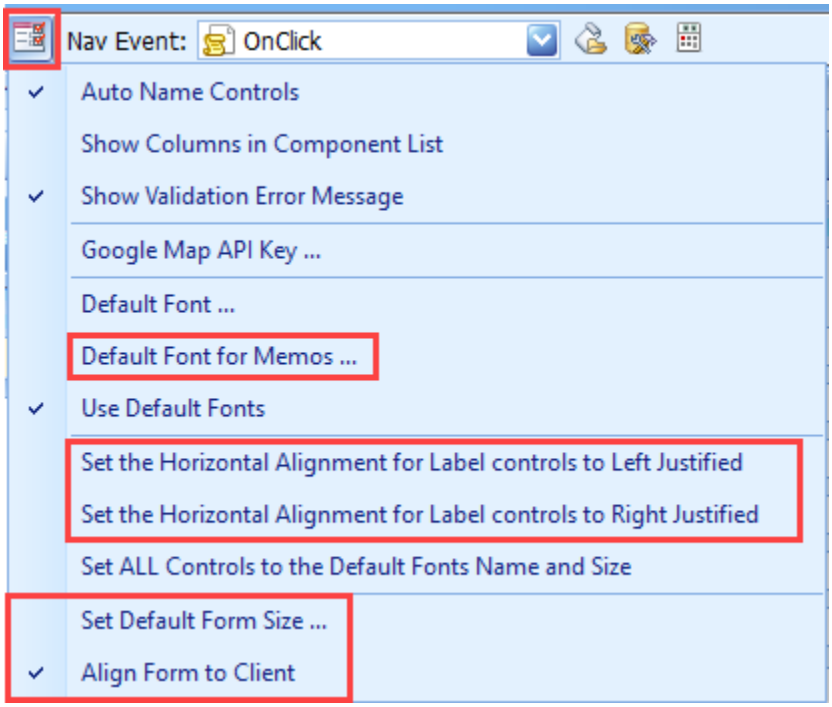
75. **Issue:** The Object Inspector in the Form Designer is drawing a blank property name when the property containing child properties is expanded.
Status: This issue has been resolved.

76. **Enhancement:** Added a property to the SQLContainer component called "AvailableInBeforeOpenTables" so that the developers will know what it is used for and added a function called "GetSQLContainer('<name of SQL Container>')". This will return a TArSQLContainer object. This enables us to set specific parameter values the same way that we do for the SQL Container in the Data Module.
The purpose of this enhancement is to be able to run queries that are defined in SQL Containers in the OnBeforeOpenTables Navigator event.

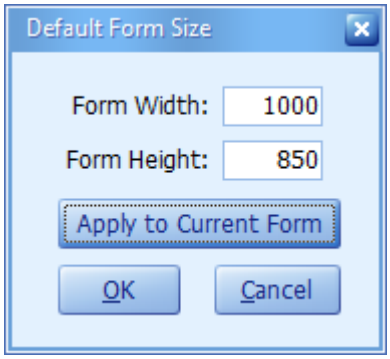
Example:

```
procedure OnBeforeOpenTables;
var
  SQLContainerTemp: TArSQLContainer;
begin
  SQLContainerTemp := GetSQLContainer('sqltblSCR_V1');
  SQLContainerTemp.SetParamValue('@Year', GetYear(Now), True);
  ExecuteSQL(tbltblSCR_V1.Connection, SQLContainerTemp.SQL.Text);

  SQLContainerTemp := GetSQLContainer('sqltblSCR_V11');
  SQLContainerTemp.SetParamValue('@Year', GetYear(Now), True);
  ExecuteSQL(tbltblSCR_V1.Connection, SQLContainerTemp.SQL.Text);
end;
```

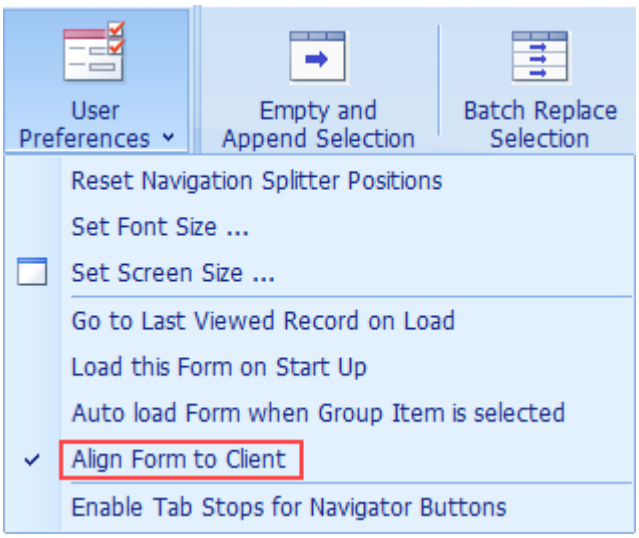
77. **Enhancement:** In Form Designer, added the following to the Form options drop-down menu:
    - Default Font for Memos – used to set the default font for memos because it is preferred to use Calibri for larger blocks of text than for standard edit boxes.
    - Set the Horizontal Alignment for Labels to Left justified. This is useful when all labels are above controls.
    - Set the Horizontal Alignment for Labels to Right justified. This is useful when all labels are on the left side of controls.
    - "Set Default Form Size" and "Align Form to Client" menu items to the Form Options dropdown menu. Click on the "Set Default Form Size" menu item to show the "Default Form Size" dialog to set the default form size for the new form. This is useful for complex forms that may be used by users who have small monitors and scroll bars are needed. Click on the "Apply to Current Form" button on the dialog to apply the form size to the current form. Click on the "Align Form to Client" menu item to turn on or off align the form to the client area.



The Set Default Form Size is shown below:



78. **Enhancement:** In Data Manager, added the “Align Form to Client” menu item to the User Preferences dropdown menu. Click on this menu item to turn on or off align the current form to the client area.



79. **Enhancement:** Added the "TrimSpaces" property to every text edit control. The default setting for this property is True. If set to True, the leading and trailing spaces in this edit control will be trimmed before posting the data to the database.

80. **Enhancement:** Added the "TrimSpaces" property to the TArDBTableEvents component. If set to True, the leading and trailing spaces of ALL text-based fields for the current record in this table will be trimmed before posting the data to the database. If an edit control that has the “TrimSpaces” property set to False, then the spaces for that field will not be trimmed even if this property in the TArDBTableEvents component is set to True.

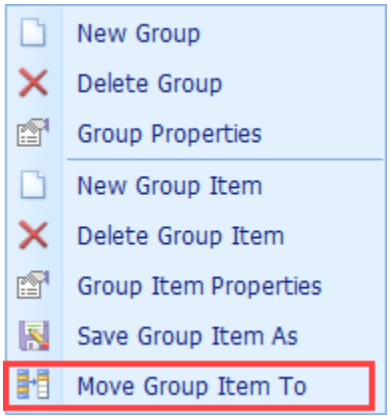
81. **Issue:** The “Remove Field” popup menu item on the “Main Connection” dialog for the Columns grid fails to remove the selected field when the selected field has been renamed or the selected field is deleted from the table or view.
Status: This issue has been resolved.

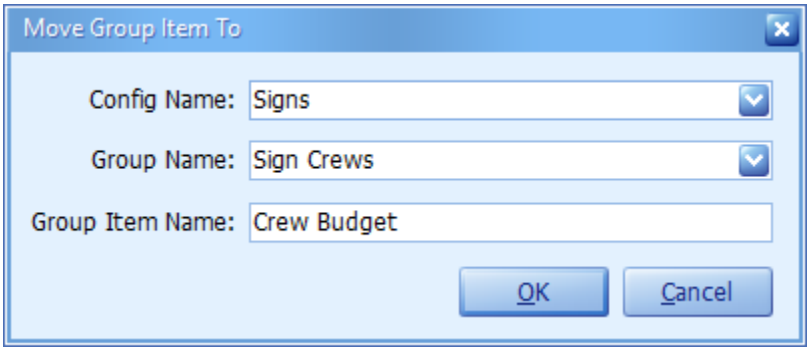
82. **Issue:** An error “Invalid Column Name” occurs on the “Export Form Data” dialog when the OK button is pressed, and the underlining table links information was not saved in correct order.
Status: This issue has been resolved.

4.1 build 9

1. **New Feature:** Added the "OnAfterResetGridLayout" event to the TArDBGrid control. This event occurs after the grid layout has been reset back to default.

2. **New Feature:** Added the "Move Group Item To" menu item to the Sidebar popup for moving the current Group Item to a different Group and/or different Config.





3. **Enhancement:** Enhanced the Export Configuration dialog to include the inactive group items in the export. Added the Active column to the grid for showing the group item's active status.

4. **Enhancement:** Enhanced the Data Transfer utility to support importing data with different time zones.

5. **Enhancement:** Added the fields "Created By" and "Last Modified By" as text fields to the "Config Properties", "Group Properties", "Group Item Properties", and "Node Properties" dialogs.

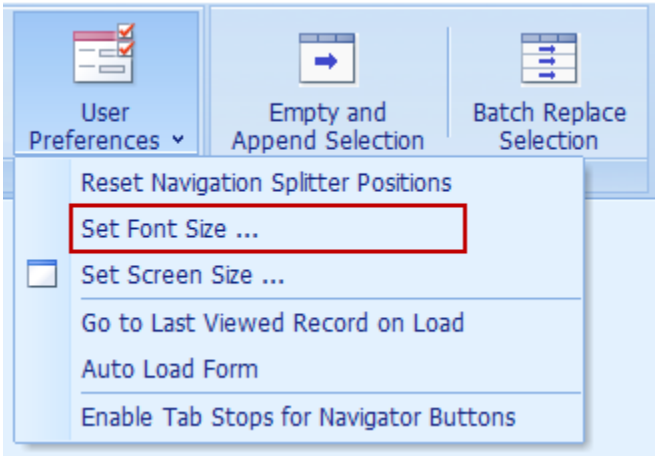
6. **Issue:** The function for the table object: tblMyTable.FieldByName('ID').AsLargeInt is not returning a correct value when the field value is more than 2.1 billion.
   Status: This issue has been resolved.

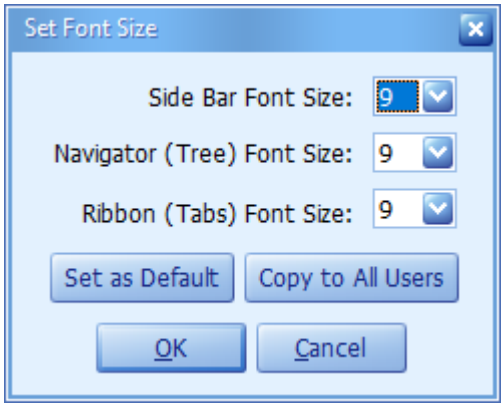
7. **Issue:** The MemList Date functions do not save the seconds portion of date time.
   Status: This issue has been resolved.

4.1 build 8

1. **New Feature:** In the Export and Import Configuration dialogs, the size of the form is saved automatically now.

2. **New Feature:** Added new menu item “Set Font Size” to the “User Preferences” drop down menu for setting the user specific font sizes. The buttons “Set as Default” and “Copy to All Users” are enabled when the user setting “User Can Set Default Font Sizes” is set to Yes. Individual users can also set the font size to their own preference.





3. **New Feature:** Added the Message Bar to the top of the form area for showing messages applicable to the form without using a Message dialog box so they are less intrusive to the users. To show the Message Bar in scripting, use the procedure:

   ShowMessageBar(messageType: TMessageType; strMessage: String; clBackground: TColor; clFont: TColor) and HideMessageBar(messageType: TMessageType).

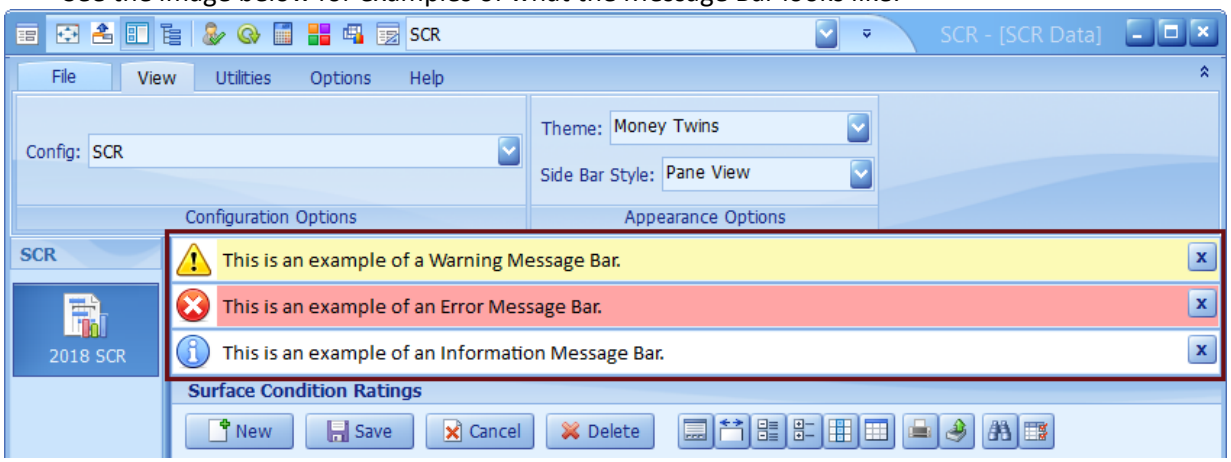
   Pass in one of the parameter values: msgError, msgWarning, msgInformation for the "messageType" parameter.

   e.g.
   ```
   ShowMessageBar(msgError, 'This is an example of an Error
           Message Bar.');

   ShowMessageBar(msgWarning, 'This is an example of a Warning
           Message Bar.');

   ShowMessageBar(msgInformation, 'This is an example of an Information
           Message Bar.');
   ```

   To hide a Message Bar, simply use the procedure:
   ```
   HideMessageBar(msgInformation);
   ```

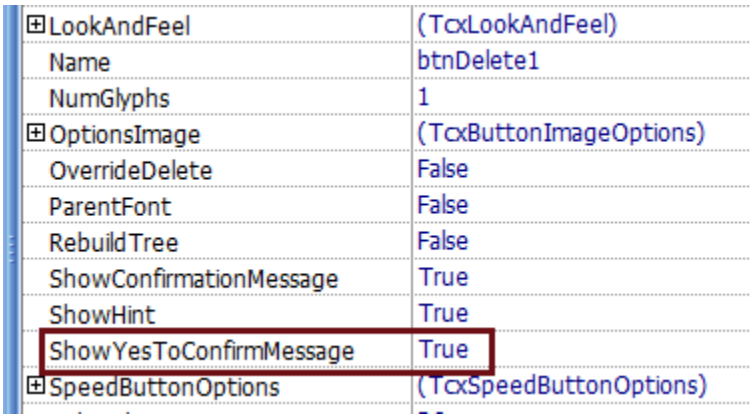
   Or to use this procedure without any parameters to hide all three at the same time:
   ```
   HideMessageBar();
   ```

   See the image below for examples of what the Message Bar looks like:

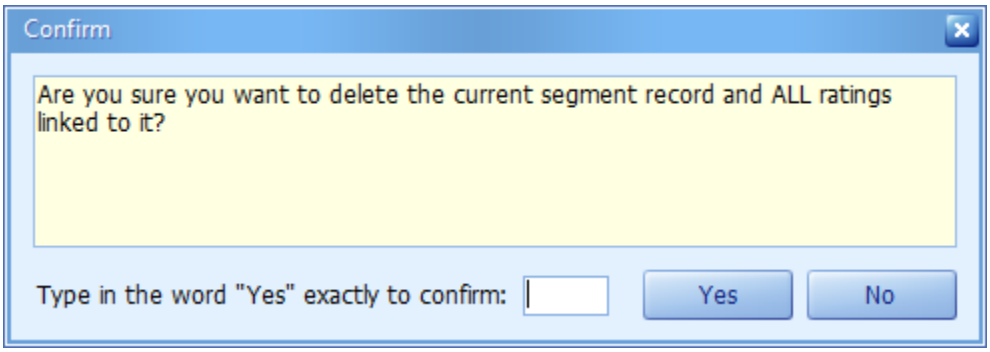


4. **New Feature:** Added a new component "TArDateEdit" to the "Additional" tab on the component palette toolbar.

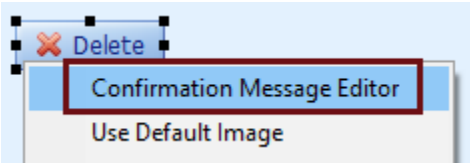
5. **New Feature:** Added the “ShowYesToConfirmMessage” property to the TArDeleteButton control. It indicates to show the edit box for the user to type in the word "Yes" exactly to confirm.



This is what the confirmation dialog will look like if the above property is set to True:



6. **New Feature:** Added the “Confirmation Message Editor” popup menu for the TArDeleteButton and TArCancelButton controls. This is for showing the Confirmation Message Editor dialog which allows the user to enter multiple lines for the message text.



7. **New Features:** Added the following functions to the Functions list:

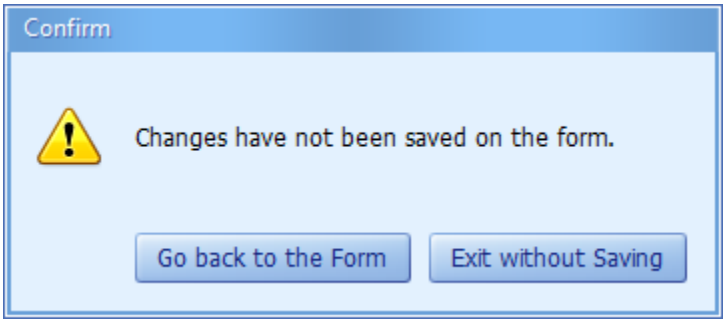
Function Name	Description
AddGridCheckComboBoxItem	Adds an item to the grid check combo box.
ClearGridCheckComboBoxItems	Clears the grid check combo box items.
HideMessageBar All	Hides the ALL Message Bars
HideMessageBar Error	Hides the Message Bar for Error
HideMessageBar Info	Hides the Message Bar for Information
HideMessageBar Warning	Hides the Message Bar for Warning

LockFormUpdate	Disables redrawing of the form area. Call LockFormUpdate before processing data in a loop. Must call the corresponding UnlockFormUpdate after processing data is done to enable redrawing of the form area so it is best to do that in a try/finally block.
	e.g.
	try
	  LockFormUpdate;
	  //do some processing
	finally
	  UnlockFormUpdate;
	end;

ShowMessageBar Error default color	Shows the message bar at the top of the form with an Error icon using the default colors.
ShowMessageBar Error with color	Shows the message bar at the top of the form with an Error icon and a specified background and font color.
ShowMessageBar Info default color	Shows the message bar at the top of the form with an Information icon using the default colors.
ShowMessageBar Info with color	Shows the message bar at the top of the form with an Information icon and a specified background and font color.
ShowMessageBar Warning default color	Shows the message bar at the top of the form with a Warning icon using the default colors.
ShowMessageBar Warning with color	Shows the message bar at the top of the form with a Warning icon and a specified background and font color.
UnlockFormUpdate	Enables redrawing of the form area.

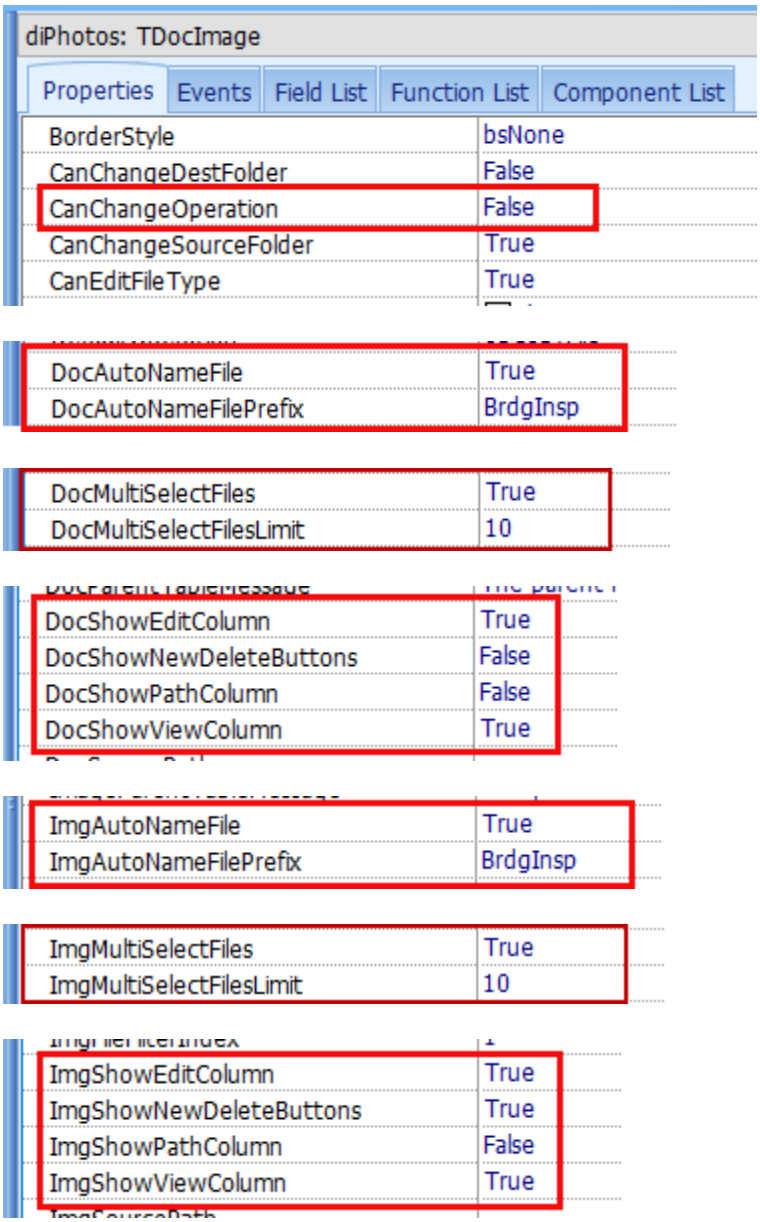
8. **Enhancement:** Enhanced the application to support locating records for up to four tables when the application is being run from Data Viewer's Edit button.

9. **Modification:** Changed the confirmation message dialog that is shown when the user trying to exit the form has modified data that has not been saved.



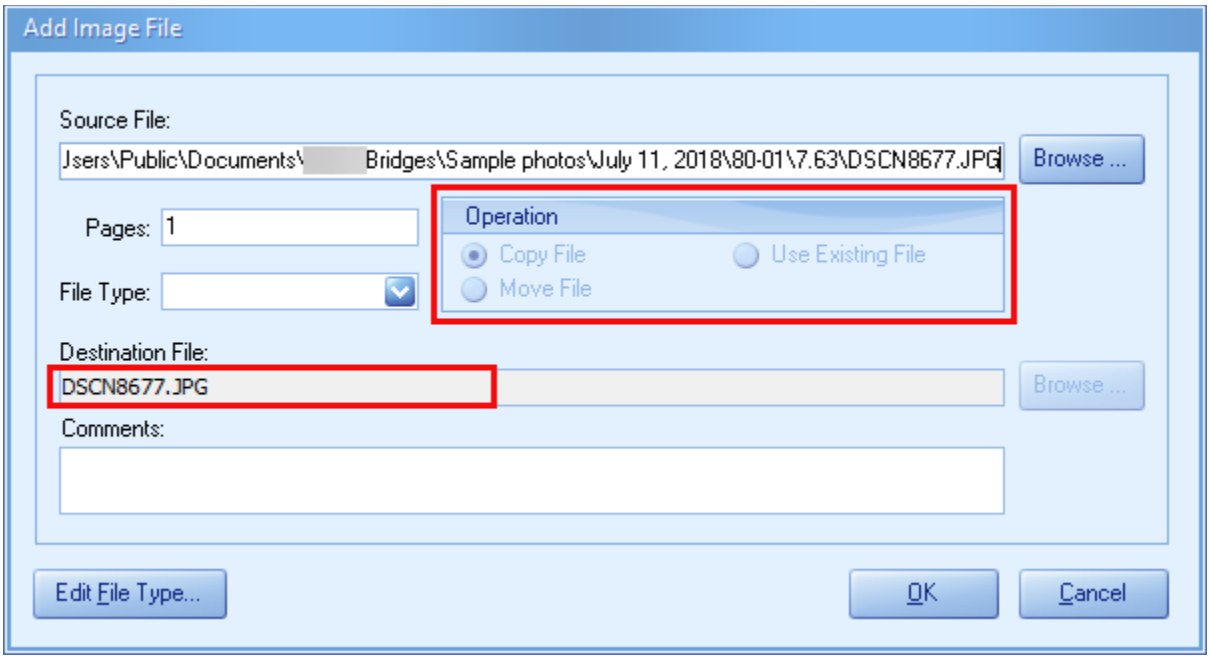
10. **Modification:** Changed the default value for the "RebuildTree" property to False for the standard buttons: TArNewButton, TArSaveButton, TArCancelButton, TArDeleteButton.

11. **Enhancement:** Added the following new properties to the Docs & Images component:

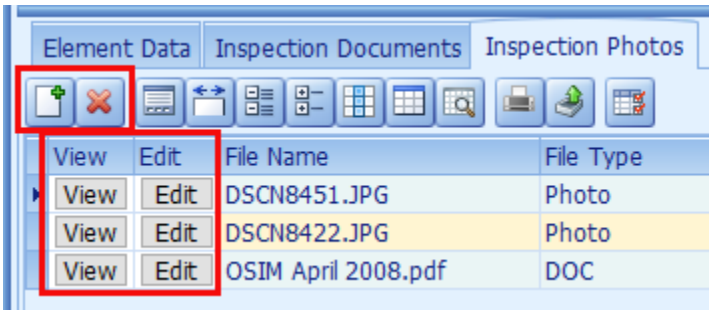


- **CanChangeOperation** property set to False will disable the ability to change the Operation option when adding a file. i.e., if you want to force the operation to always copy the file to a predefined location, set the Default operation to opCopyFile and set the CanChangeOperation property to False.
- **DocAutoNameFile** and **ImgAutoNameFile** properties set to True will rename the file for the Copy File and Move File operations to the **DocAutoNamePrefix** and **ImgAutoNamePrefix** respectively plus the ID number of the record. This is to help ensure that the file names are always unique when copying or moving to a central location on a server.
  For example, if the prefix is culv and the ID is 1234 then a doc file will be called "culv1234.docx". Note: this feature does not apply if the Operation is: Use Existing File.

- **DocShowPathColumn** and **ImgShowPathColumn** properties show and hide the path column accordingly. Set this property to False if you do not want the users to see where the files are located.
- **DocMultiSelectFiles** and **ImgMultiSelectFiles** properties indicate to allow users to multi-select files when adding new files to the grids.
- **DocMultiSelectFilesLimit** and **ImgMultiSelectFilesLimit** properties specify the maximum number of files can be selected at once so that users do not accidently add too many files to a given record. The default is 10 but you can set it to 0 (zero) for unlimited files if needed.



- **DocShowNewDeleteButtons** and **ImgShowNewDeleteButtons** properties will show or hide the New and Delete buttons accordingly.
- Added two new methods to add records to the Documents and Images grid. These can be used in a custom button when the New button is hidden.
    - AddDocumentRecord e.g., MyDocImage. AddDocumentRecord;
    - AddImageRecord e.g., MyDocImage. AddImageRecord;
- **DocShowEditColumn** and **ImgShowEditColumn** properties will show or hide the Edit column accordingly. Hide this column if you do not want users to be able to open the file.
- **DocShowViewColumn** and **ImgShowViewColumn** properties will show or hide the View column accordingly. Hide this column if you do not want users to be able to open the file.

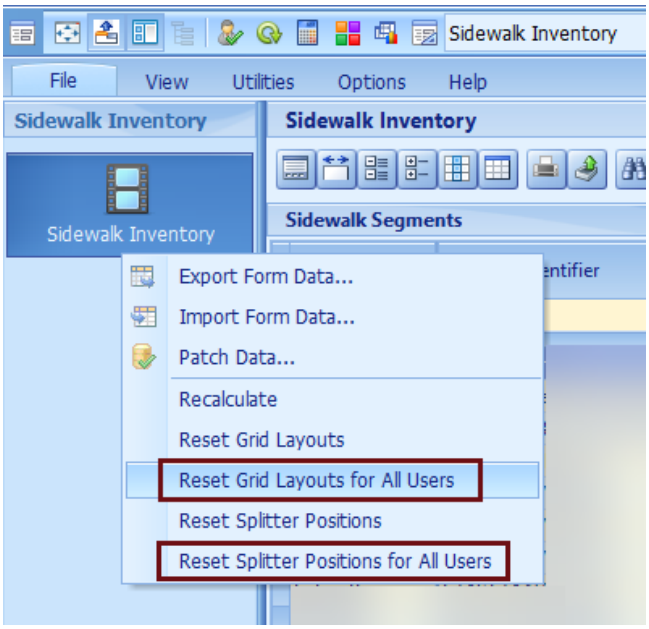


12. **Enhancement:** Added “DocAutoNameFileAddIDSuffix” and “ImgAutoNameFileAddIDSuffix” properties to TDocImage component which indicate to automatically append the ID suffix to the destination file name.

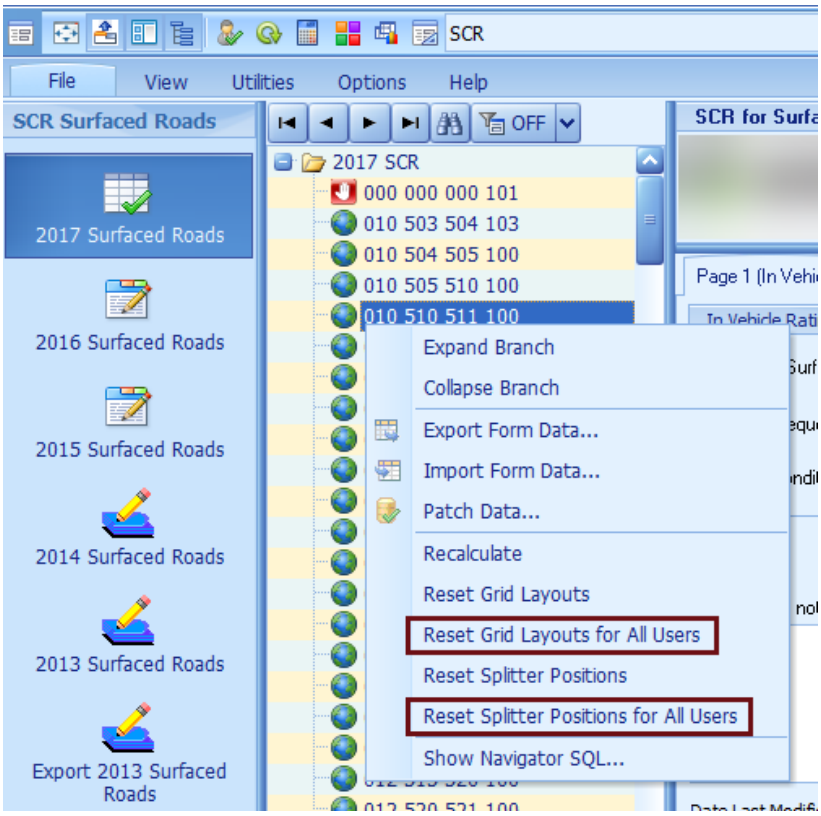
13. **Enhancement:** Enhanced the “Documents Grid Columns” and “Images Grid Columns” dialogs for the TDocImage component to include Up and Down buttons for ordering the documents/images grid columns. User can also drag and drop the grid row to order the documents/images grid columns.

14. **Enhancement:** Added two new menu items to the Sidebar and Navigator that are available only for users with Designer access to reset the grid layouts and splitter positions for all users of that form. This provides the developer of the forms an easy way to put the grids and splitters back to the new designed settings if they are changed. This is particularly important when columns are added to or removed from grids in the Form Designer.

    Menu Items for the Sidebar when the Navigator is not used:

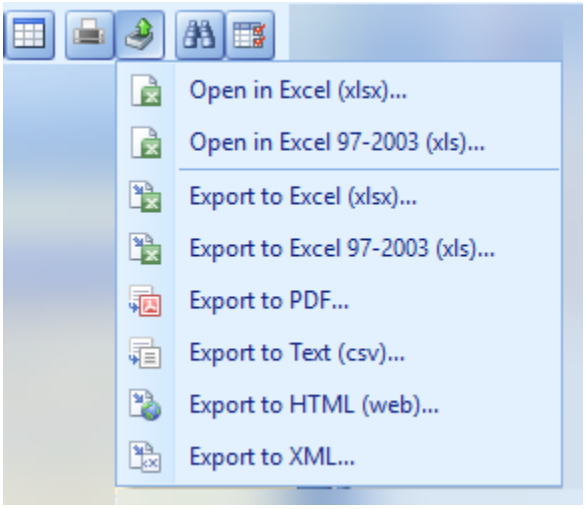


Menu Items for the Navigator when it is used:



15. **Enhancement:** Significantly improved the performance of the importing data from Access, Excel, and CSV.

16. **Cosmetic:** In the Grid Toolbar component, we replaced the export drop-down menu with a newer one that follows the themes. We also changed the caption for some of the menu items to be more consistent with the newer versions of Excel and we added an icon for each menu item.



17. **Issue:** An error can occur in the Export Form Data dialog when the Navigator Filter is used, and the Navigator Filter value contains replaceable text.
Status: This issue has been resolved.

18. **Issue:** In the GIS Layers Editor dialog, some controls are not showing properly when the layer type is “MsSql Spatial”.
    Status: This issue has been resolved.

19. **Issue:** An error occurs when loading the GIS map for the “MsSql Spatial” layer with the spatial field containing null value.
    Status: This issue has been resolved.

20. **Issue:** In the Form Designer, when cutting parent container controls (like Group Boxes, Page Controls, etc.) the assignment of the events for the controls within would be lost.
    Status: This issue has been resolved. Now the Form Designer recursively maintains all the events of all controls when cutting and pasting parent container controls (like Group Boxes, Page Controls, etc.) with unlimited number of levels.

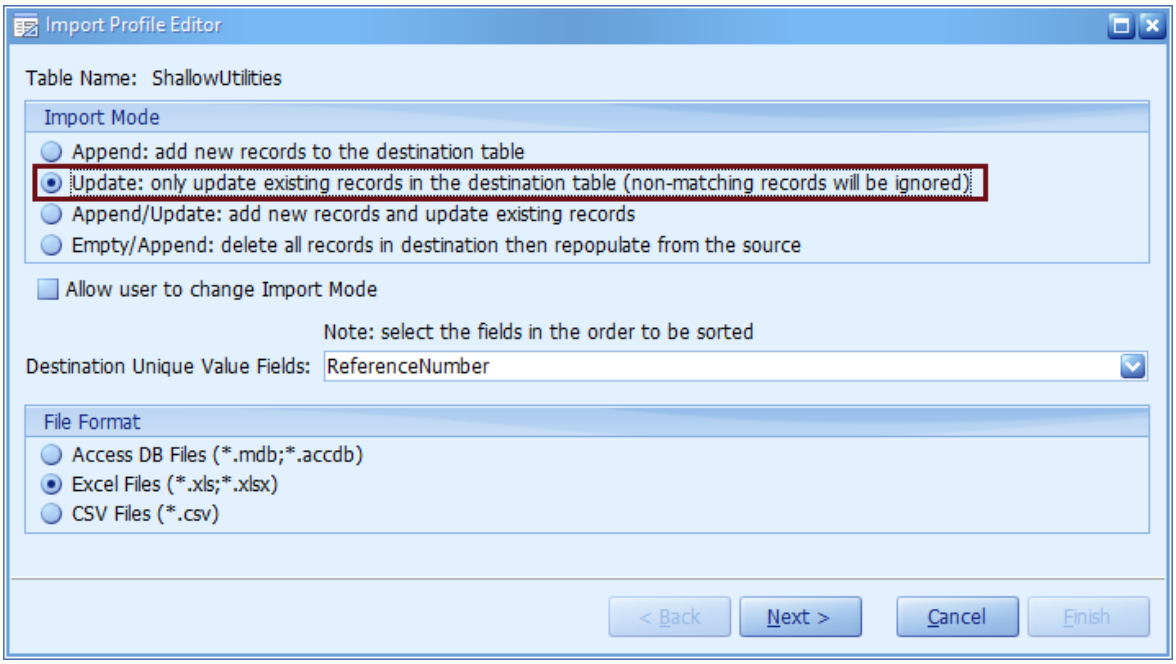
4.1 build 7

1. **Enhancement:** Added new functions to the MemList object for supporting 64-bit Integers in scripting:

   MemList.SetInt64(strKey: String; intValue: Int64): Integer
   MemList.GetInt64(strKey: String; intDefault: Int64): Int64

2. **Enhancement:** Significantly improved the performance for the primary key generation process.

3. **Enhancement:** In the Import Profile Editor dialog, added a new import mode “Update: only update existing records in the destination table (non-matching records will be ignored)”.

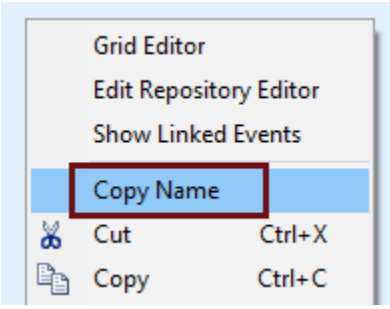


4. **Enhancement:** In the Import Profile Editor dialog, changed the “Destination Unique Value Fields” picklist to support selecting the fields in the order to be sorted. i.e., If the fields: RoadName, FromKM, ToKM are used for searching then select them in that order.

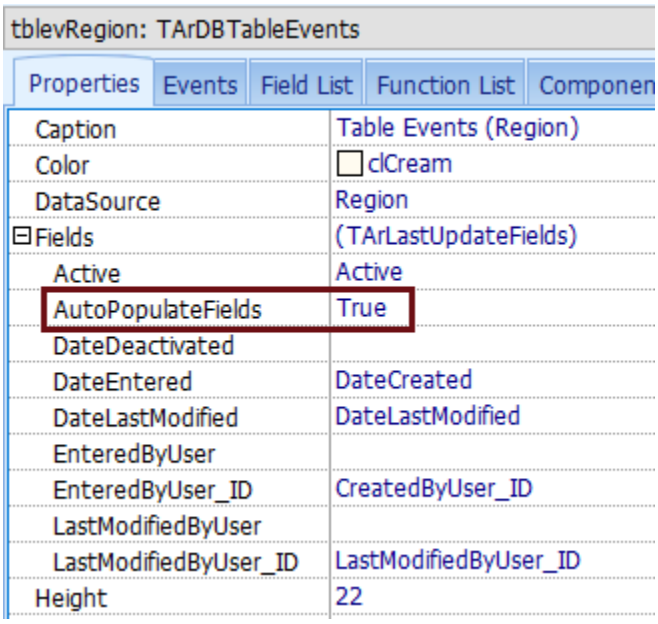
5. **Modification:** The check box “Automatically create secondary index for performance improvement” has been removed. Due to the improved performance of the import data process, the secondary index is not needed anymore.

6. **Cosmetic Enhancement:** Changed the fonts on the main part of the form (including the Ribbon, Sidebar, and Navigator) and dialogs to **Tahoma, Size 9** so the text is easier to read.

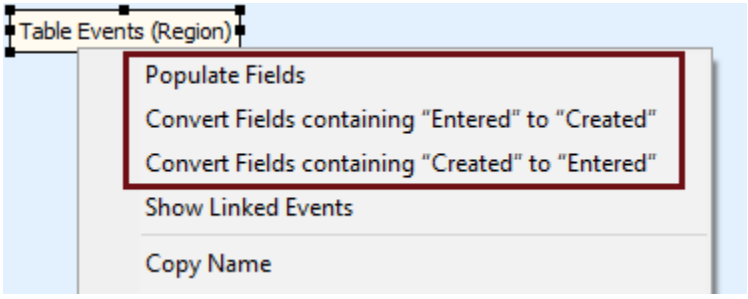
7. **New Feature:** Added the "Copy Name" menu item to the right-click menu of every control to quickly and easily copy the name property of the control to the clipboard. This is handy for scripting.



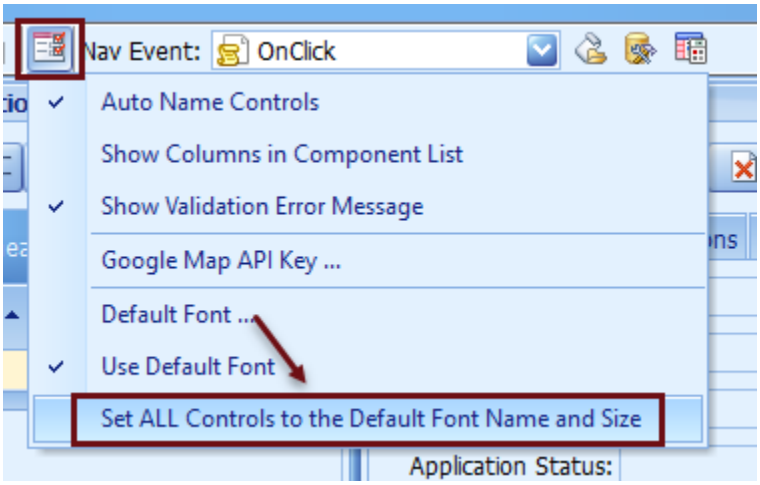
8. **New Property:** Added the AutoPopulateFields property to the TArDBTableEvents control. The default setting of this property is True. Set it to False if you do not what the field names to be populated automatically.



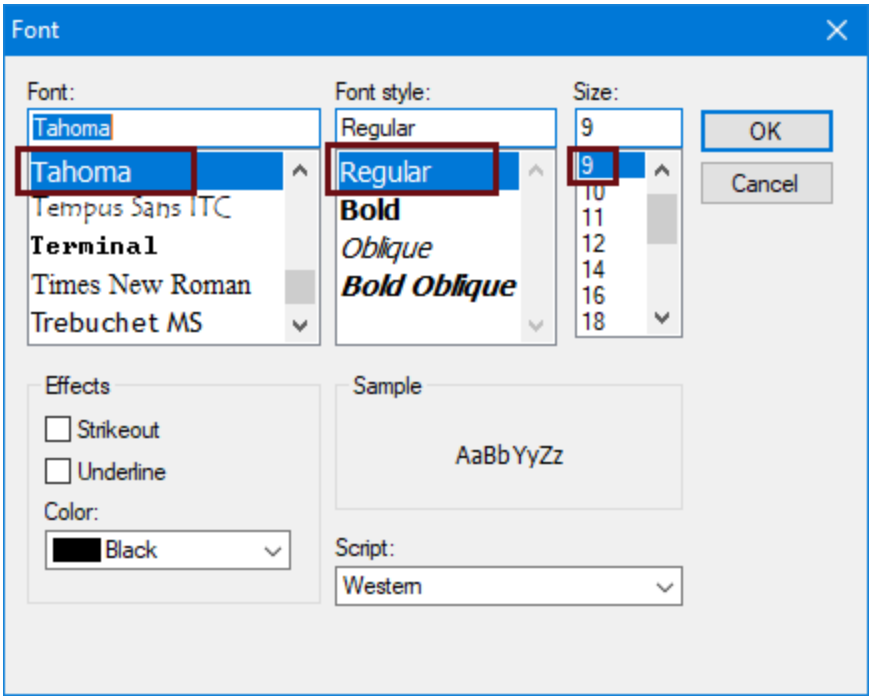
9. **New Feature:** Added the “Populate Fields”, “Convert Fields containing 'Entered' to 'Created'” and “Convert Fields containing 'Created' to 'Entered'“ menu items to the right-click menu of TArDBTableEvents and TArDBLastUpdate controls. Use the “Populate Fields” menu item to populate the Fields for the selected controls with the AutoPopulateFields property set as True. Use the other two menu items to convert the Fields containing “Entered” to “Created” or convert the Fields containing “Created” to “Entered”. This is handy if you have fields like DateEntered and you want the change them to DateCreated and vise versa. Note: the desired fields must already exist in the table in the database. Note: this feature also supports multi-selection of these controls so you can make the same change to all of them at once.



10. **New Feature:** In the Form Designer, added the ability to set font name and size for all the controls on the form to the Default Font. This feature only takes about a second to process complex forms, so it is a significant time-saver when changing the fonts on existing forms. Note: other styles like Bold and Underline are not affected by this process.

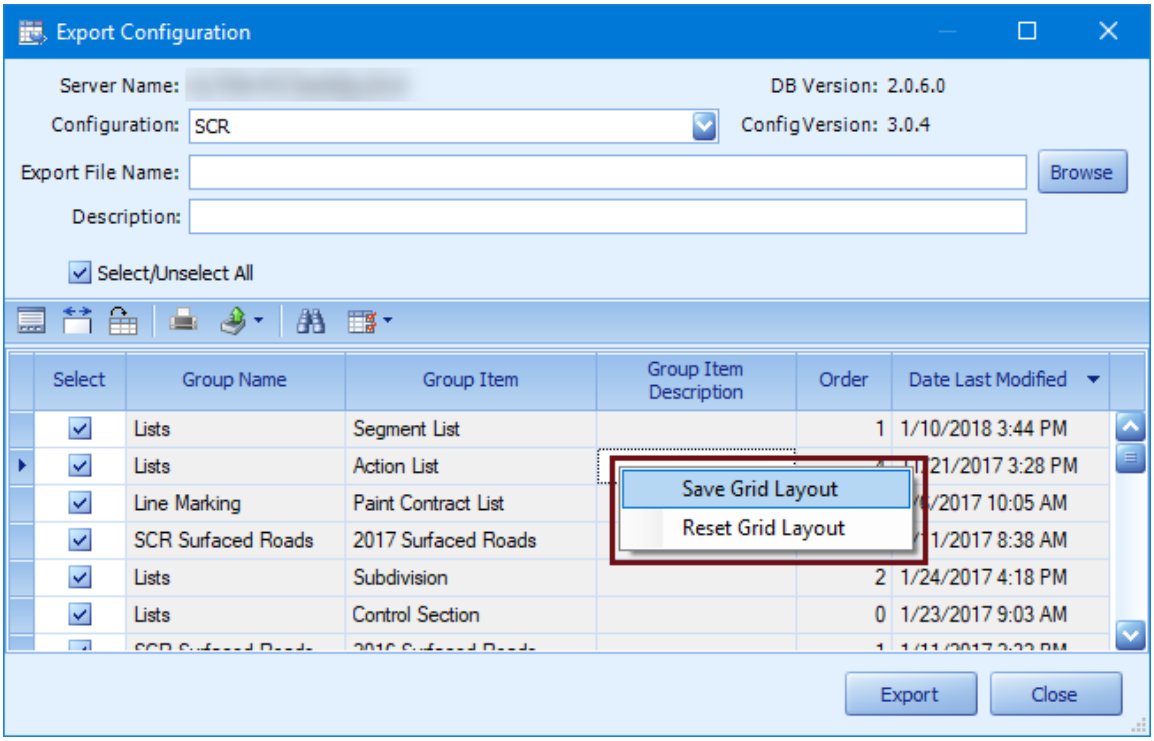


We recommend using **Tahoma**, **Regular**, size **9** for the default font as shown below:



11. **New Feature:** In the Export and Import Configuration dialogs, the size of the form is saved automatically.

12. **New Feature:** In the Export and Import Configuration dialogs, added the ability to Save and Restore the Grid Layouts in a right-click menu as shown below. This is handy if you want to sort the grid descending by Date Last Modified so the most resent changes are at the top.



13. **Issue:** An error can occur in the Export Form Data dialog when the Navigator Filter is used, and the Navigator Filter value contains replaceable text.
Issue: This issue has been resolved.

14. **Issue:** In the GIS Layers Editor dialog, some controls are not showing properly when the layer type is "MsSql Spatial".
    Issue: This issue has been resolved.

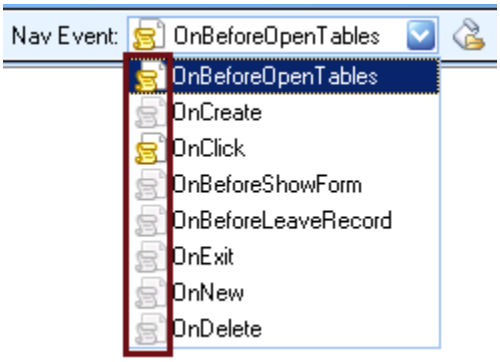
15. **Issue:** An error occurs when loading the GIS map for the "MsSql Spatial" layer with the spatial field containing null value.
    Issue: This issue has been resolved.

4.1 build 6

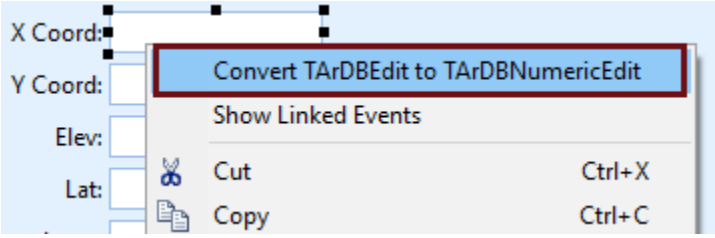
1. **New Feature:** Added a new navigator event "OnBeforeShowForm" for scripting. This event occurs just before the form is shown and after the OnClick event. This event can be used for certain scripting like initializing controls on the form the first time it is loaded. Note: if you need to do any processing each time the Navigator is clicked then use the OnClick event instead.

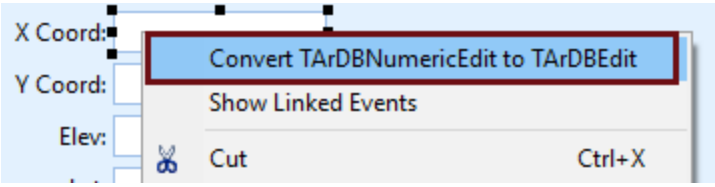
2. **Modification:** Changed the Navigator event combo box items to be listed in sequential order of the events being triggered as shown here:
    a. **OnBeforeOpenTables** – This event is great for setting initial properties such as: FilterSQL, OrderFields, or master/detail relationship before the tables are opened. Note: since this event occurs before the form is created, you have no access to the controls on the form at this point.
    b. **OnCreate** – This event can be used for initializing variables or controls on the form before going to the record in the Navigator.
    c. **OnClick** – This event occurs every time the users click on the Navigator.
    d. **OnBeforeShowForm** – This event can be used for certain scripting like initializing controls on the form the first time it is loaded after the users click on the Navigator.
    e. **OnBeforeLeaveRecord** – This event occurs when users are about to exit the application or go to another form (in the Sidebar).
    f. **OnExit** – This is the final event that occurs just before users are about to exit the application or go to another form.

3. **Enhancement:** Added icons to the Navigator event combo box to indicate which events contain scripting. The following screenshot shows that the "OnBeforeOpenTables" and "OnClick" events contain scripting.



4. **New Feature:** Added the ability to convert DB Edit controls on the form to DB Numeric Edit controls and visa versa using the right-click menu for these controls. All applicable properties will be set automatically. The DB Numeric Edit control is designed (and preferred) for numeric data so this is very handy new feature to convert DB Edit controls on existing forms.





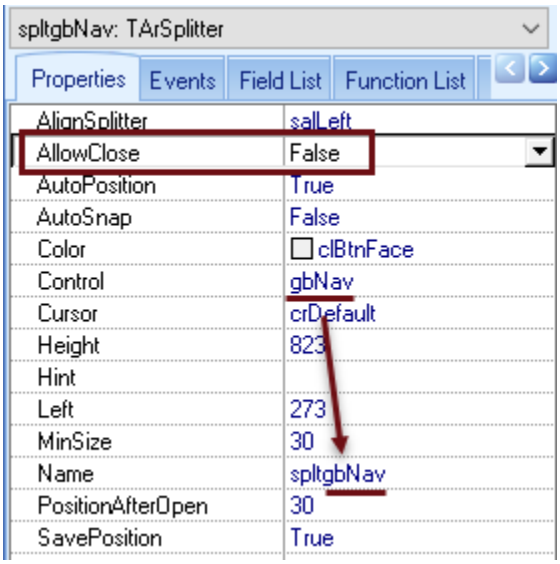
5. **Enhancement:** When the parameters hint is showing in the Code Editor while typing the script, you can press the Esc key to hide it immediately.

6. **Enhancement:** Added variable "AppInfo.IsNavigationEnabled" for use in scripting to check if the navigation is enabled or not.

7. **Enhancement:** Modified the Form Designer so the property: "Properties.IncrementalFiltering" for the TArDBComboBox and TArDBLookupComboBox controls is customizable and it retains the value that is set.

8. **Enhancement:** Added the Auto Naming feature to the Splitter control so now when the Control property is set, the Name of the splitter will be set accordingly as shown below in the next item.

9. **Enhancement:** Added a new property called: "AllowClose" to the "TArSplitter" control to determine whether to allow the splitter to be automatically closed when users click on it. The default value for this property is False.



10. **Enhancement:** Various performance and memory enhancements.

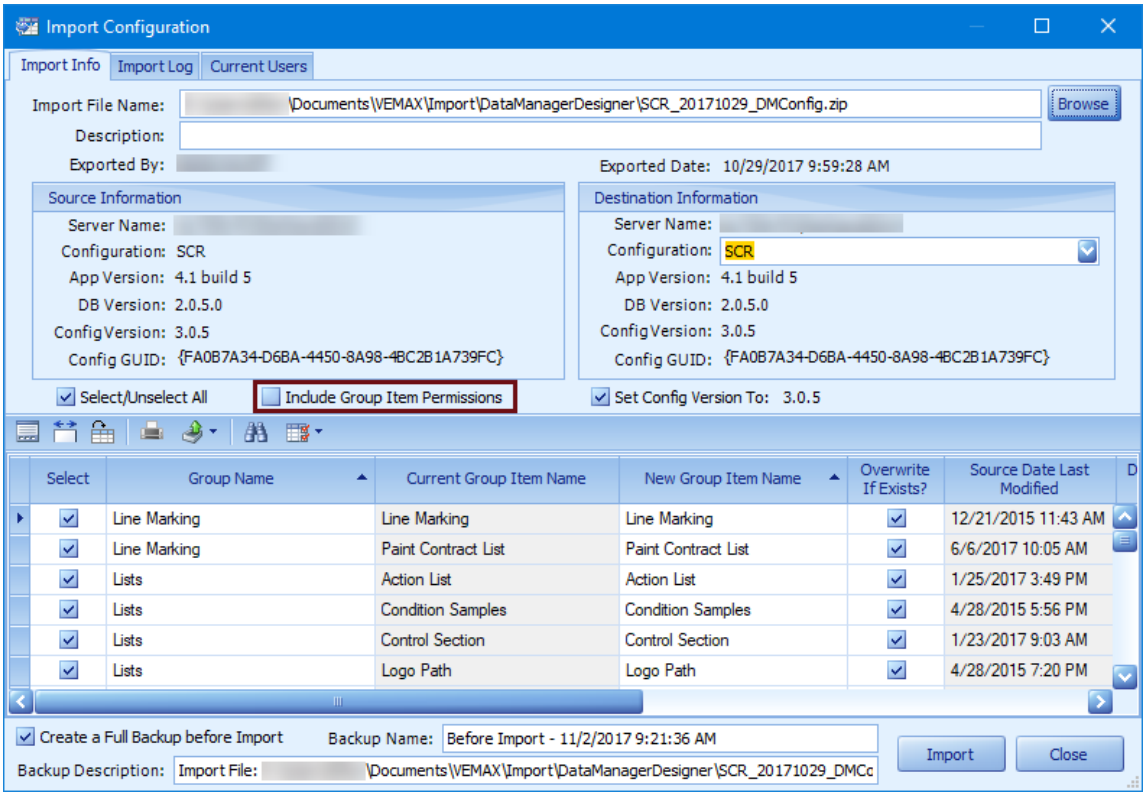
11. **Modification:** Disabled the mouse wheel on edit controls to avoid accidentally changing the data. Now spinning the mouse wheel is only for scrolling rather than editing data.

12. **Issue:** Running the Data Transfer import in silent mode may fail to recognize the import file. **Status:** This issue has been resolved.

4.1 build 5

1. **Upgrade:** Third-party control for the Google Map has been upgraded to the latest version.

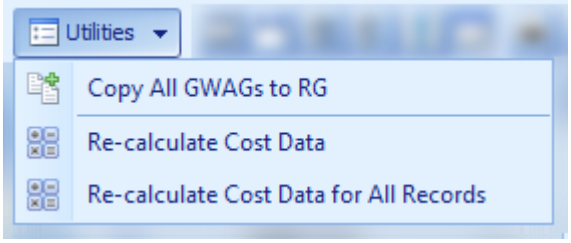
2. **Enhancement:** Added a new check box "Include Group Item Permissions" to the "Import Configuration" dialog so you can choose to include the group item permissions for the import. The default for this check box is unchecked.

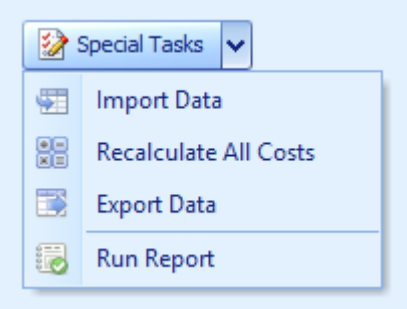


4.1 build 4

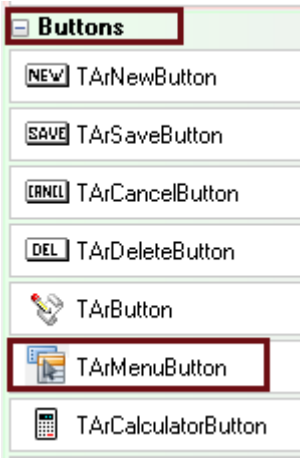
1. **New Component:** Added a new component in the Form Designer called: **TArMenuButton**
   Use this component to set up drop-down menu items for various functionalities. This special button greatly reduces the amount of real-estate needed to add multiple features on a form. Instead of using a separate button for each feature, use this Menu Button and save a lot of space on the screen.
   Set the property "MenuImages" if you want a customized image for each menu item. Set the property "MenuItems" to configure the menu items. The event "OnMenuItemClick" occurs when users click on the menu item. Use the parameter "intMenuItemID" to determine which menu item is being clicked. See the details below.

   Examples of the Menu Button:

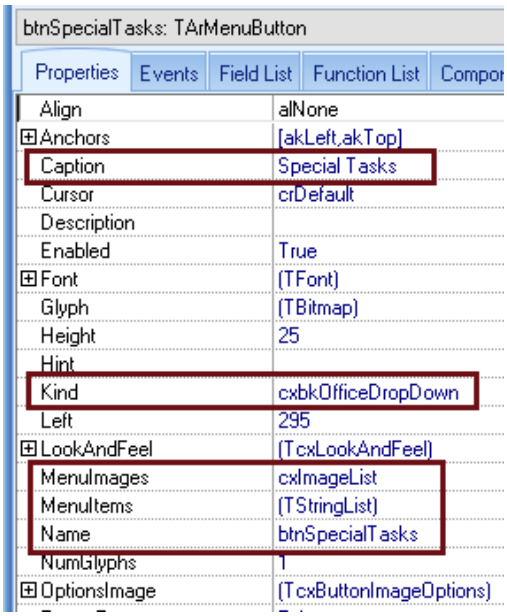




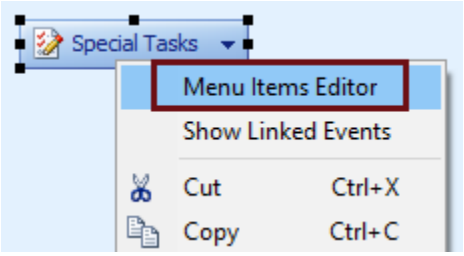
In the Components list, the new TArMenuButton is below the TArButton



Important properties of the Menu Button

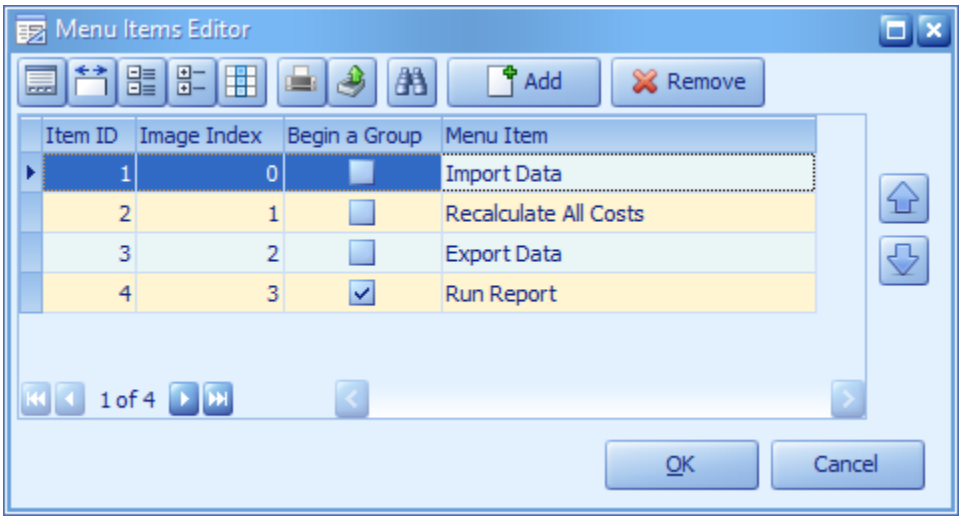


To add menu items, simply right-click on the Menu Button on a form then click on the **Menu Items Editor**.

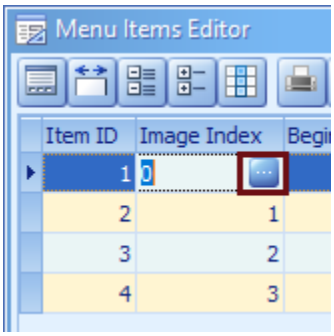


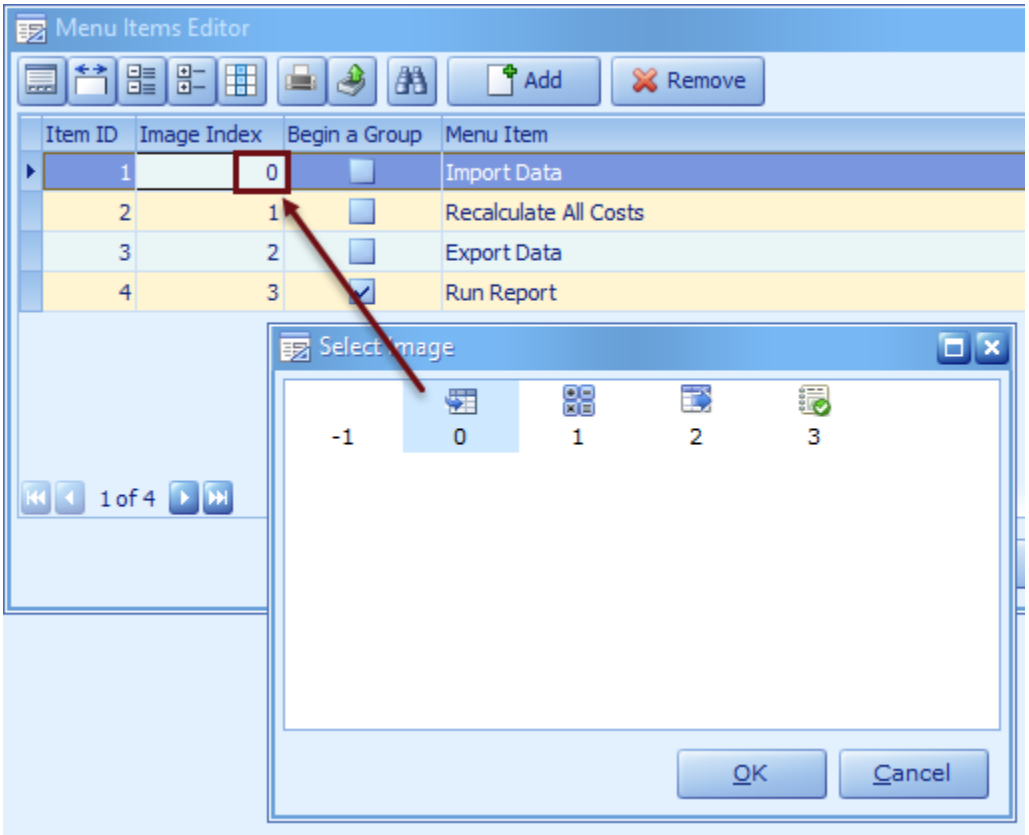
The **Menu Items Editor** looks like the following screen-shot.
Click on the **Add** button to add as many menu items as needed.
Click on the **Remove** button to remove the selected menu item.
Click on the up and down arrow buttons (on the right side of the grid) to move the menu items up or down respectively.

<u>Columns in the grid:</u>
Item ID: is the unique identifier for each menu item. This is used in the OnMenuItemClick event to identify which menu item users clicked on.
Image Index: is the index of the images in the Image List. As mentioned above, the Image List is identified with the MenuImages property.

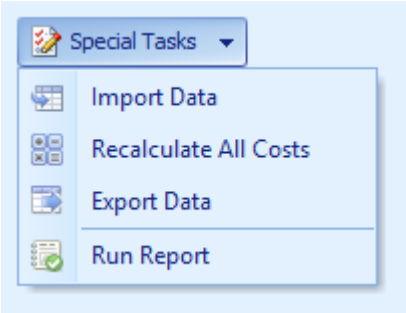


The ellipsis button in the Image Index column opens the list of images that were added to the Image List component.

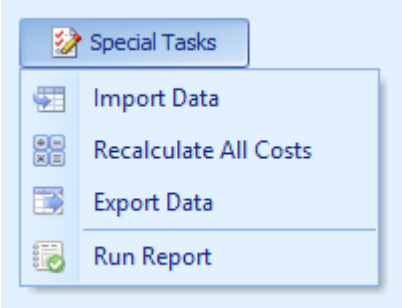




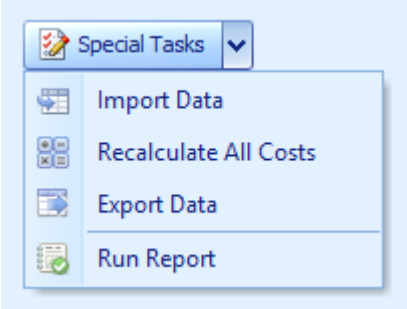
Begin a Group: adds a horizontal line above the menu item and is typically used as a logical separator of "like" items.
Menu Item: is the caption of the menu item that the users will see.



Kind property set to: **cxbkOfficeDropDown**
Clicking anywhere on the button will show the drop-down menu.
This is the default setting.

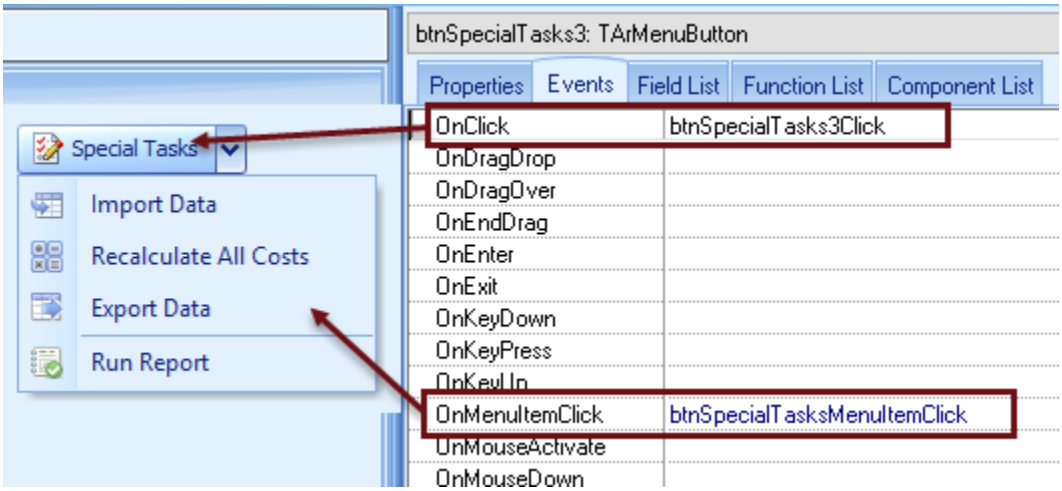


Kind property set to: **cxbkDropDown**
This option is the same as the one mentioned above except it hides the drop-down arrow button.
Clicking anywhere on the button will show the drop-down menu.



Kind property set to: **cxbkDropDownButton**
This enables a separate button to the left of the drop-down button.
Clicking on the button to the left of the drop-down button will execute the script in the OnClick event.
Clicking on the button on the right that shows the down arrow will display the menu.
See the screenshot below for details about this option.

When the **Kind** property set to: **cxbkDropDownButton**, the OnClick event can be used for the button to the left of the drop-down button.
In every "Kind" mode, the OnMenuItemClick is used to create a script for each menu item, as shown below.

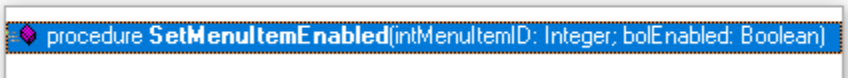


Here is an example of the **OnMenuItemClick** event.
We recommend using a case statement like this to simplify the scripting. The text to the right of each item number (e.g., "ImportData") is simply the name of a procedure that contains the script that you want to execute when the users click on that menu item.

```
procedure btnSpecialTasksMenuItemClick(Sender: TObject; intMenuItemID: Integer);
begin
  case intMenuItemID of
    1: ImportData;
    2: RecalculateAllCosts;
    3: ExportData;
    4: RunReport;
  end; //case
end;
```

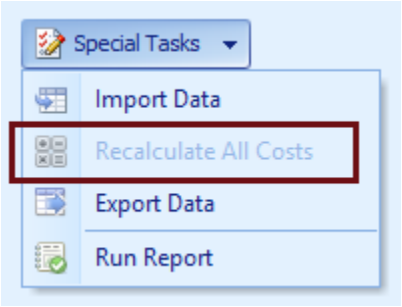
To disable / enable individual menu items, use the "SetMenuItemEnabled" procedure.
The parameters are the **Item ID** of the menu item and the Boolean value for **enabled** (
True = enabled, False = disabled)

```
btnSpecialTasks.SetMenuItemEnabled
```



e.g.,
btnSpecialTasks.SetMenuItemEnabled(**2**, False);

Here is the results of the above line of code:



The following line of code enables the second menu item:

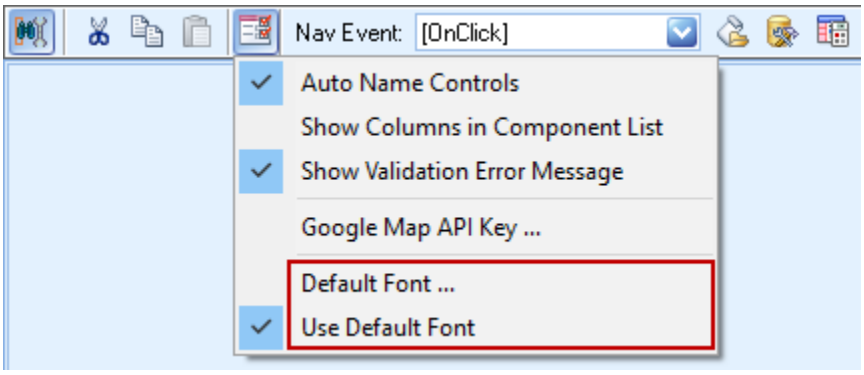
btnSpecialTasks.SetMenuItemEnabled(**2**, True);

Other applicable procedures with the same parameters as mentioned above:

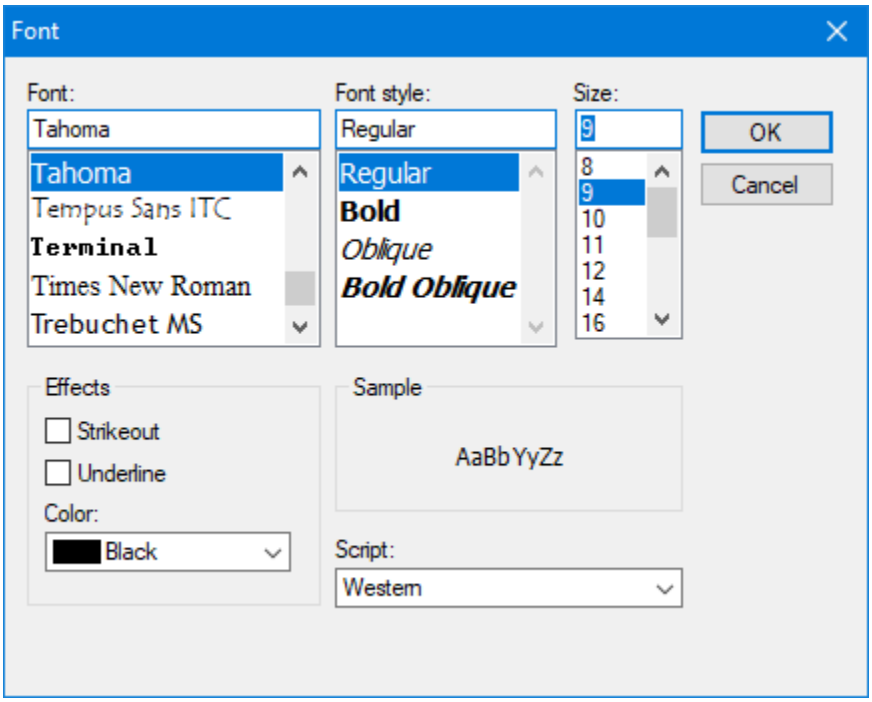
To show / hide a menu item:
btnSpecialTasks.SetMenuItemVisible(**2**, True);

To start or turn off a group (horizontal line above the menu item):
btnSpecialTasks.SetMenuItemBeginAGroup(**2**, True);

2. **New Feature:** Added two new menu items “Default Font” and “Use Default Font” to the “Form Options” tool bar button in the Form Designer. Use the “Default Font” menu item to set the default font for use when adding new components to the form. The “Use Default Font” check menu item must be checked for this to work. To ignore the default font, uncheck the “Use Default Font” menu item. The settings for these two menu items are saved for the entire application. These settings are not user-specific settings so an agency should decide what the standard font will be for all new forms.
   Note: the default font does not affect any controls that are already on a form. To change the font on existing controls, use the Font property.



   Recommend using **Tahoma, Regular** font, Size **9** as shown below.



3. **Enhancement:** Improved internal support for Statistical and other types of indexes.

4. **Enhancement:** Improved internal support for multiple Schemas. However, we still highly recommend that only dbo be used as the schema for all tables unless there is a very good reason to use multiple schemas. E.g. dbo.tblMyTable

4.1 build 3

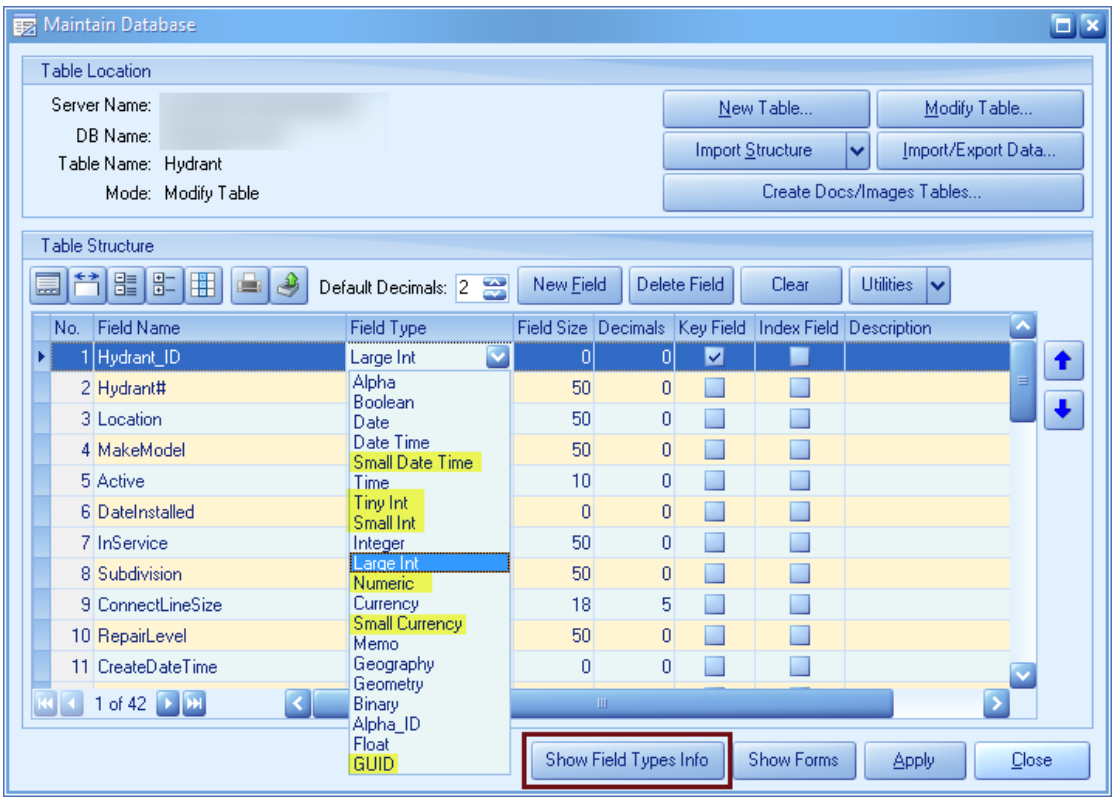
1. **Upgrade:** Upgraded the Google Map component.

4.1 build 2

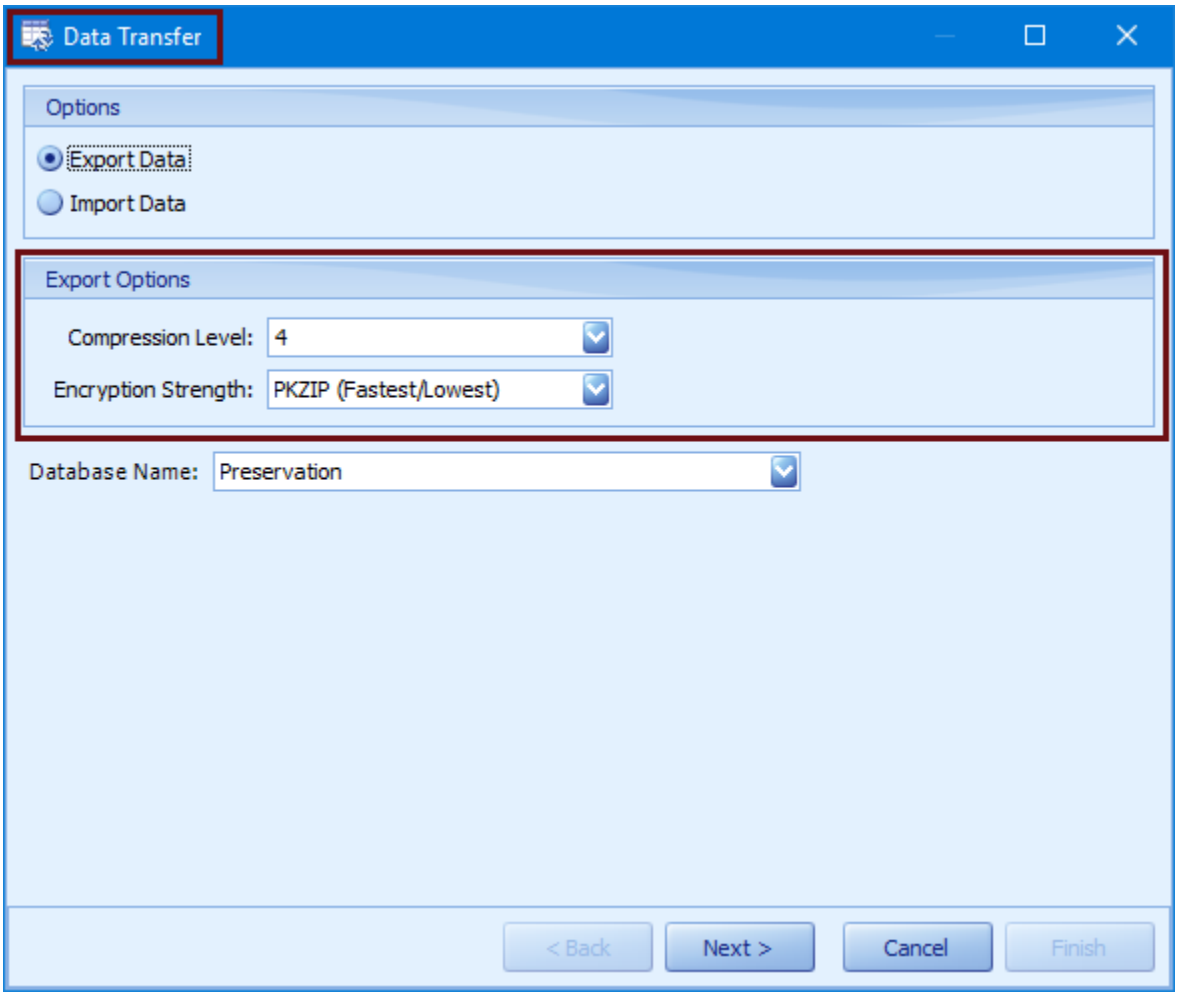
1. **Enhancement:** Added additional error checking in various functions to improve the quality of the scripting.

2. **Enhancement:** Changed the System ID for key generation to support 4 digits in the numeric System ID.

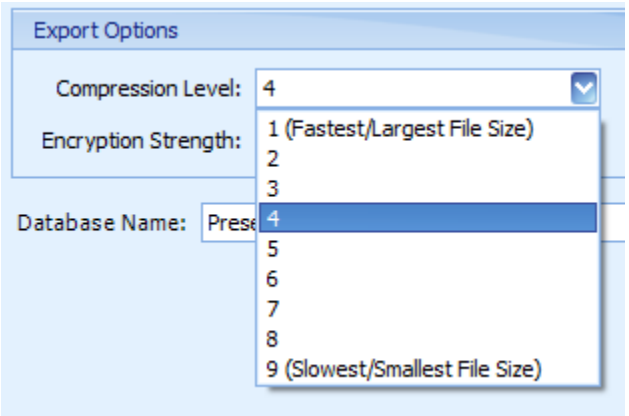
3. **Enhancement:** In the Maintain Database screen, added support for both Numeric and Float data types. It is recommended to use the Numeric data type instead of Float. The old Float data type has been changed to Numeric data type. The new Float data type is for storing approximate numerical values with no fixed field size and number of decimals. Also added support for Tiny Int, Small Int, Small Date Time, Small Currency and GUID.
   Added the "Show Field Types Info" button to open a document that describes all the Field Types.



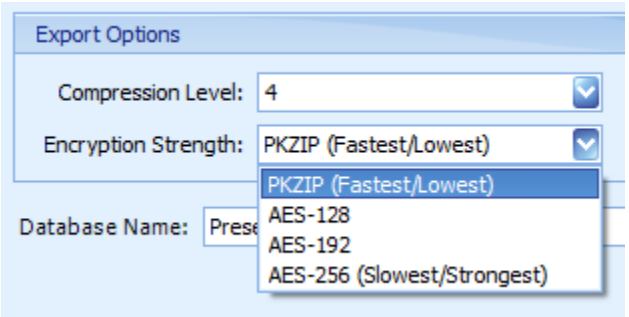
4. **Enhancement:** In Data Transfer, added the following Export Options:
   Compression Level – Used to determine the amount of compression.
   Encryption Strength – Used to determine how secure you want the zip files to be. See the details on the following page.



The **Compression Level** options are shown below.
The default level is: 4. The lower levels are much faster than the higher levels so you should only use the highest level (9) when the smallest size possible is needed.



The **Encryption Strength** options are shown below.
Unless the tables that you are exporting contain highly confidential information, we recommend using the PKZIP encryption strength because it is good enough for most situations and is usually more than 3 times faster than the stronger encryption options. If the tables contain highly confidential information, then we recommend using a higher Encryption Strength (like AES-128 at least).



4.1 build 1

1. **Enhancement:** Added the following new properties to the TDocImage component for filtering the files in the open file dialog:

   DocFileFilter
   DocFilterFilterIndex
   ImgFileFilter
   ImgFileFilterIndex

2. **Enhancement:** Added the new event "OnBeforePostImage" to the TDocImage component for processing the image before the image is posted to the database.

   Example:

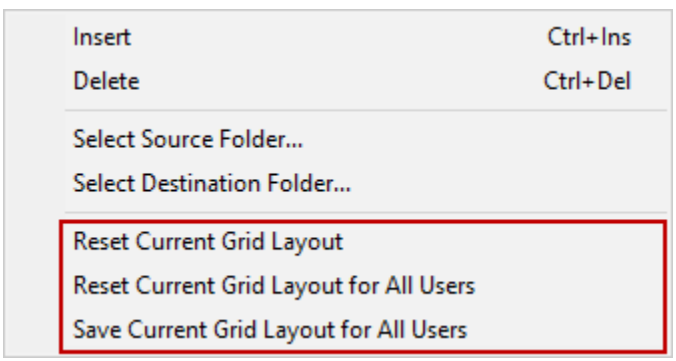
```
procedure DocImage1BeforePostImage(Sender: TObject; var bolCancel:
  Boolean);
begin
   DocImage1.CurrentImage.Proc.Resample(150, 150, rfLanczos3, True);
end;
```

3. **Enhancement:** Added the new event "OnBeforePostDocument" to the TDocImage component for processing the document before the document is posted to the database.

   Example:

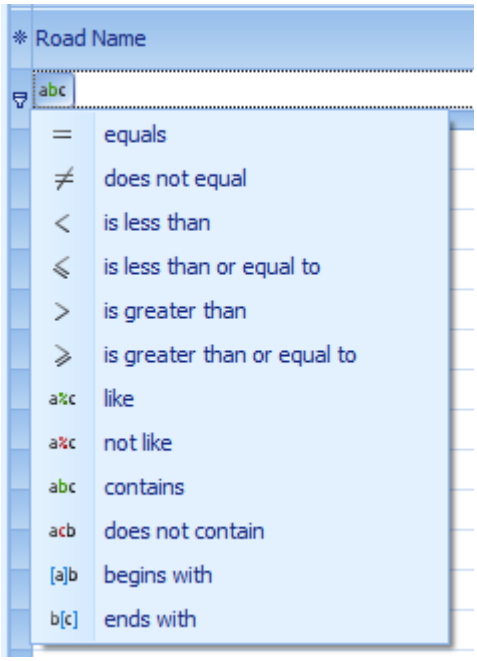
```
procedure DocImage1BeforePostDocument(Sender: TObject; var bolCancel:
 Boolean);
begin
    //File Size in Bytes, 1 MB = 1048576 Bytes
    //In this example, check if the file size is greater than 10 MB
    If  (DocImage1.CurrentDocument.FileSize > 10485760)  then
    begin
        if (MessageDlg('Do you want to save the file "' +
            DocImage1.CurrentDocument.FileName +
          '" to the database with the file size greater than 10 MB?',
          mtConfirmation, SetOf([mbYes, mbNo]), 0) <> mrYes)  then
           bolCancel := True;
    end;
end;
```

4. **Enhancement:** Added popup menu items to the Docs/Images grid for users who have Data Manager Designer access to reset and save the grid layout for all users of that form.

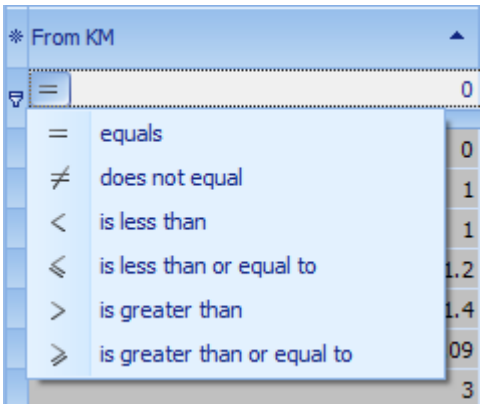


5. **New Feature:** Added several filtering options in the filter row of the grid.

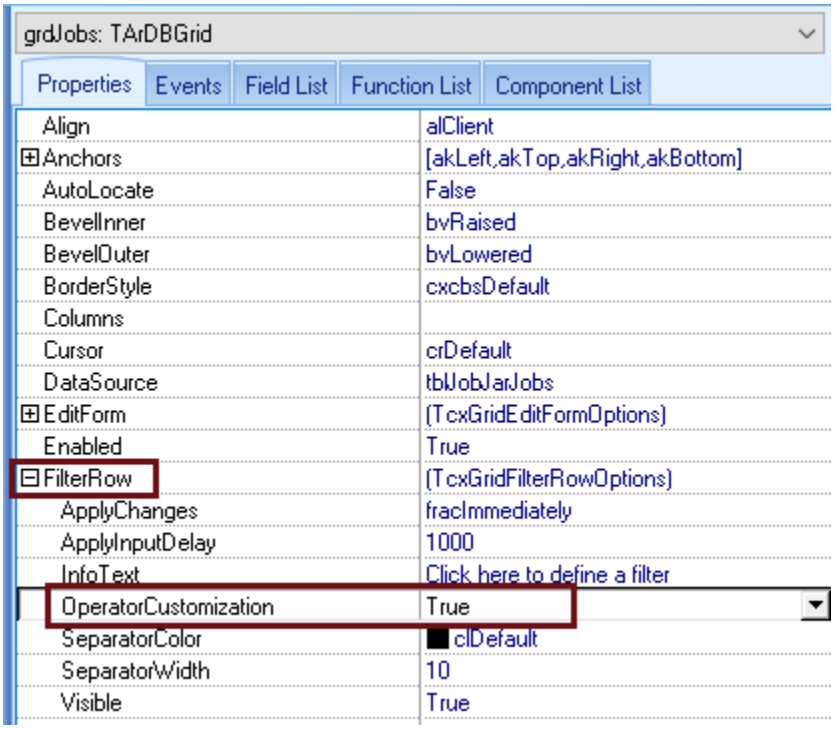
For text columns:



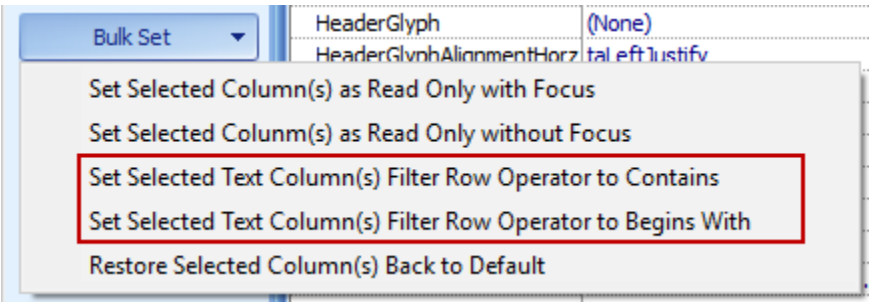
For numeric columns:



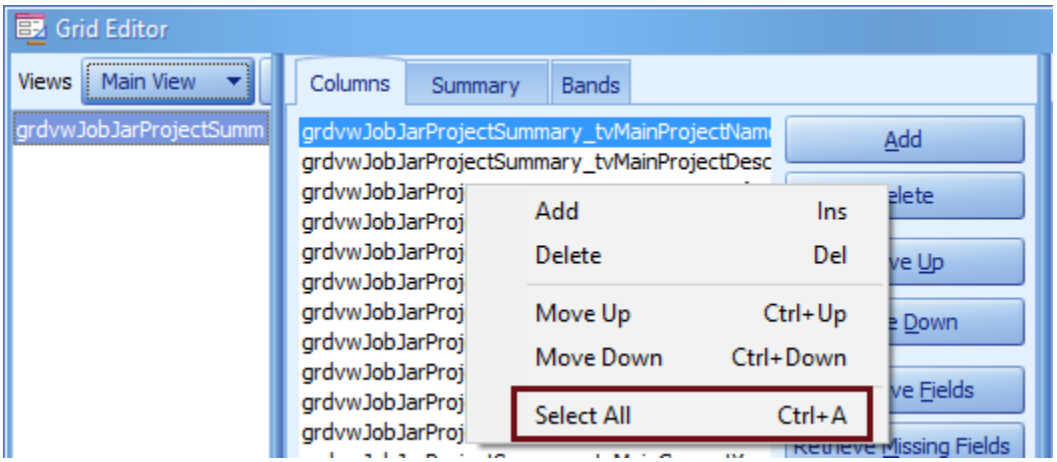
To enable this feature in Data Manager Designer, set the OperatorCustomization property to True. This is located under the FilterRow property group as shown below:



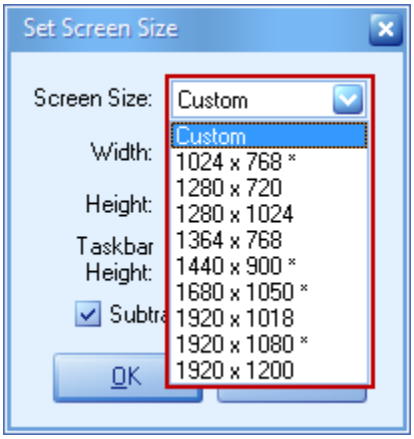
6. **Enhancement:** Added two new popup menu items to the “Bulk Set” button in the Grid Editor for setting the selected text column(s) filter row operator to “Contains” or “Begins With”.



To use this feature, select all applicable columns then click on one of the above Bulk Set options. To select all the columns, right-click on the columns list then click on the “Select All” menu item or press Ctrl+A.



7. **Enhancement:** Added more pre-defined screen sizes to the picklist in the Set Screen Size dialog. The items in this list that end with * are the most common screen sizes.

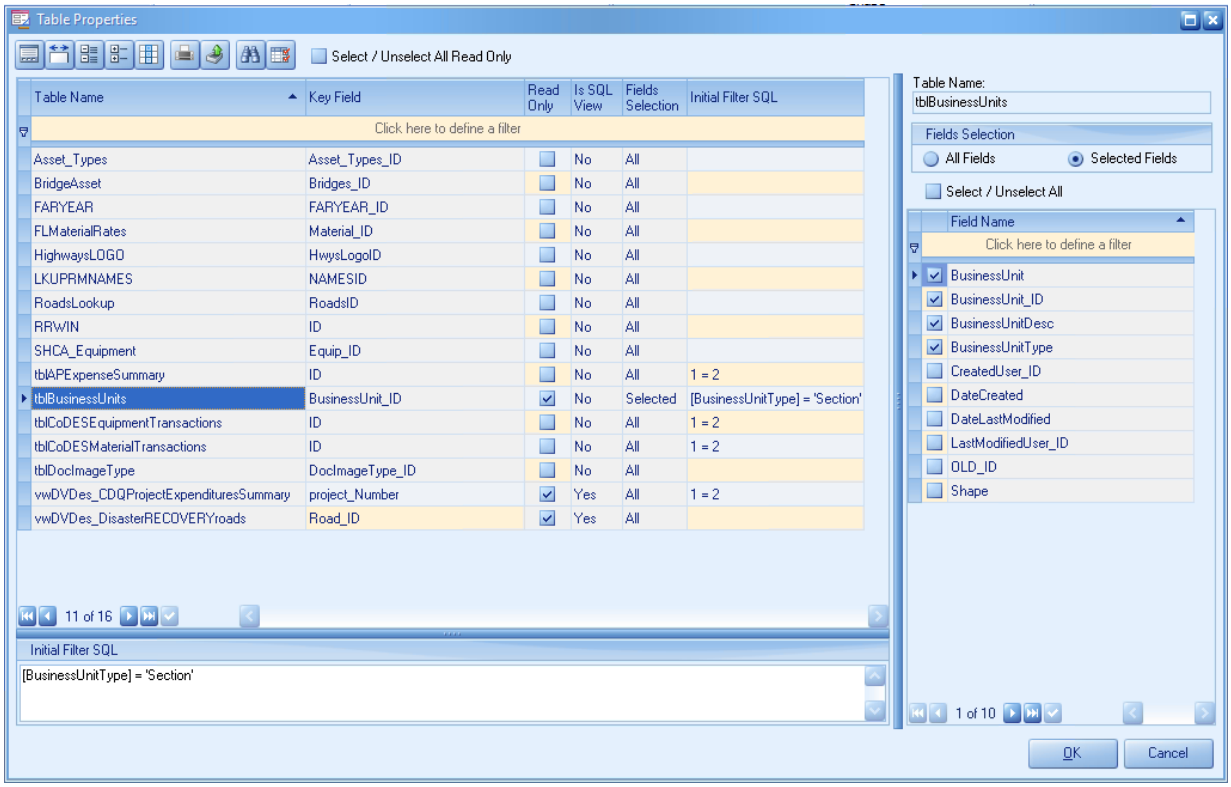


4.1 build 0

1. **Major New Feature:** Added the Table Properties button to the Main Connection and Lookup Lists Connection screens. This button opens the Table Properties screen which allows the user to set the table properties for each individual table.

   Key Field: Set the primary key field for the table.
   Read Only: Set the table as read only to prevent the user from editing the data; if "Is SQL View" is Yes, then Read Only check box will always be checked.
   Fields Selection: Set the fields selection to "All Fields" or "Selected Fields". Use the "Selected Fields" to limit the number of fields being loaded, which improves performance when loading the form.
   Initial Filter SQL: Set the initial filter SQL to filter the data being loaded initially for the table. Enter the text "1 = 2" if you want to load no data for the table. The table's filter can be overwritten by setting the table component's FilterSQL property value in scripting.

   For Example:
   tblMyTable.FilterSQL := 'RatingYear = 2017';
   tblMyTable.Filtered := True;

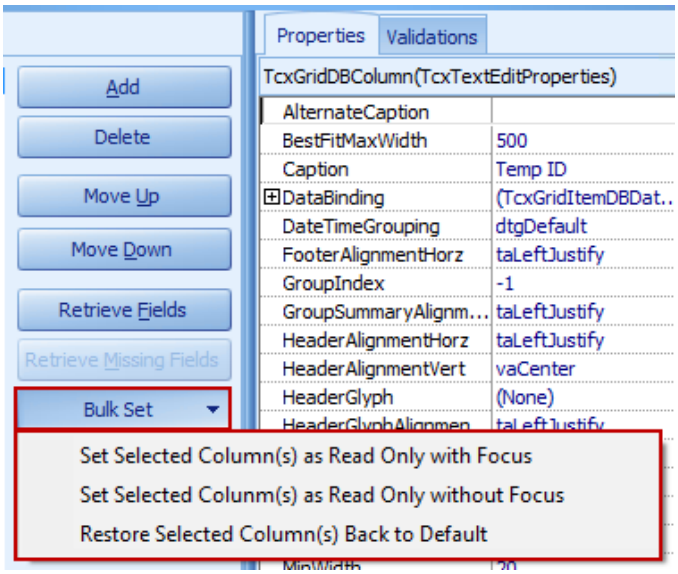


2. **Major Upgrade:** To improve compatibility with Windows 10 and SQL Server 2016, we have upgraded our development tools and third-party controls (such as the grids, edit controls, buttons, etc.) to the latest version.

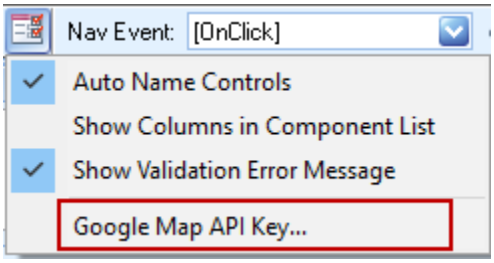
3. **New Feature:** Added new property AutoLocate to the component TArDBGrid. It indicates to automatically locate the last viewed record for the table when the form is loaded. To suppress the auto-locate process, put the following code in the OnClick event:

   AppInfo.SuppressAutoLocate := True;

4. **New Feature:** Added the button "Bulk Set" to the Grid Editor for setting the common properties for the selected column(s).



5. **New Feature:** Added the menu item "Google Map API Key" to the Form Designer's Form Options button for setting the Google Map API Key. The Google Map API Key is required for loading the Google Map and it will be applied to the component property "TArGoogleMap.APIKey" in all forms if this property has not already been set in the Form Designer.



6. **Enhancement:** Added two optional parameters "bolKeepNewRecords" and "strDateLastModifiedFieldName" to the function "DataTransferImport" for keeping the new records in the destination table by comparing the DateLastModified field values. The parameter "bolKeepNewRecords" is default to "True" if it is not specified. The parameter "strDateLastModifiedFieldName" is default to "DateLastModified" if it is not specified.

7. **Enhancement:** Improved error checking in the Form Designer.

8. **Issue:** The auto completion list in the code editor is showing duplicated property names.
   Status: This issue has been resolved.

9. **Issue:** In the Main Connection or Lookup Lists Connection dialog, without changing the existing Calculated Fields and pressing the OK button caused the existing Calculated Fields to be lost.
   Status: This issue has been resolved.

10. **Issue:** An error occurs when loading the Docs/Images grid that contains PDF file on the Images tab.
    Status: This issue has been resolved.

4.0 build 4

1. **Upgrade:** Third-party control for the Google Map has been upgraded to the latest version.

2. **Enhancement:** Added a new property "ReadOnlyRowFocus" to the TArDBGrid component. When this property is set to True, the read only row can have focus when the user clicks on the row cell.

3. **Enhancement:** Significantly improved Data Transfer to use memory more effectively when exporting or importing tables with large data size such as tables containing many photos or large documents.

4. **Enhancement:** Made some improvements to the TArDBImportData component.

5. **New Feature:** Added the following new functions to Data Manager Designer. The DataTransfer Import enables designers to add the ability to import Data Transfer files via a script button on the form rather than using the Data Transfer dialog. Doing so, makes importing these files much easer for the users.

Category	Function Name	Function Type	Description
Database	ClearMaxID	Procedure	Clears the Max ID record for the specified table.
Database	DataTransferImport	Function	To import data from the Data Transfer file with the extension of ".vdt" or ".zip". Set the parameter impImportMode to: impAppend, impAppendUpdate, impEmptyAppend. The optional parameters strVerifyTableName and strVerifyDatabaseName are for verifying if the import file contains the valid table name and/or valid database name.
Settings	SaveUserDefinedSettingDate	Procedure	Saves the date setting value for the current user.
Settings	GetUserDefinedSettingDate	Function	Gets the date value of the given user defined setting.
Settings	SelectFileInExplorer	Procedure	Opens the Windows Explorer and selects the specified file.

6. **New Feature:** Added non-data-aware control TArCalcEdit to Form Designer.

7. **Enhancement:** In Maintain Database screen, changed the Table Name box editable for new tables.

8. **Issue:** In Form Designer's Events Inspector, double clicking on an existing event may cause the code changes to be lost in certain situations.
   Status: This issue has been resolved.

4.0 build 3

1. **Enhancement:** Added a new style "cxStyleDefault" for use on the TArDBGrid component. When this style is used, it will draw the grid using the default color based on the selected theme. If no style is selected, the application will also use the selected theme.

2. **Modification:** Changed the component TArDBGridso it does not to automatically draw the entire grid by using the ReadOnlyRowColor property value when the grid is set as read only.

3. **Modification:** Changed the way that Data Manager is checking for the application license information.

4. **Modification:** Changed the table component to be loaded initially as read only if it is linked to a SQL view. Designer users can use the OnBeforeOpenTables event to set the read only property for any of the table components if necessary.

   For example:
   ```
   procedure OnBeforeOpenTables;
   begin
     tblMyTable.ReadOnly := True;
   end;
   ```

4.0 build 2

1. **New Feature:** Implemented the scripting event "OnCreate" which is being triggered after the form is created. Use this event to do initialization for the form.

2. **Enhancement:** Improved the process for Export Form Data so it's more responsive and is less likely to have the exporting files being locked by another process.

3. **Enhancement:** Improved the Export Form Data filter so it handles filtering blank values for the operators "Equals" and "Not equal".

4. **Enhancement:** Improved the Data Transfer export filter so it handles filtering blank values for the operators “Equals” and “Not equal”.

5. **Enhancement:** Improved the DataTransferExport() function to handle the table filter string more robustly.

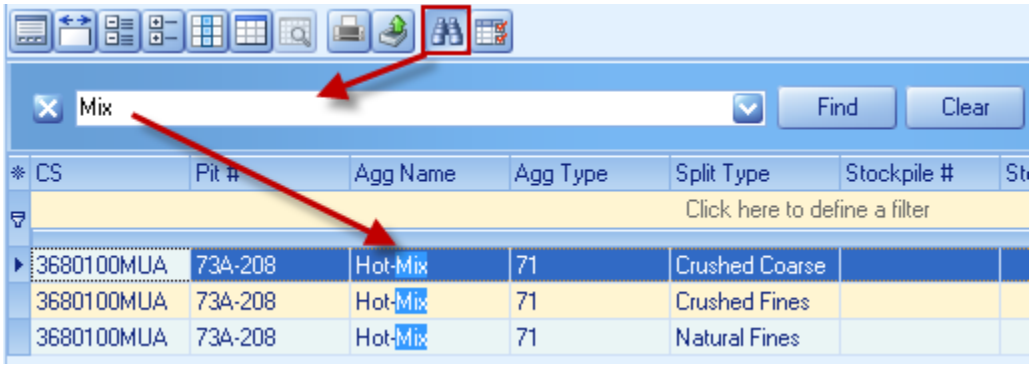
4.0 build 1

1. **Enhancement:** Enhanced Data Manager to launch other applications securely from the quick access buttons.

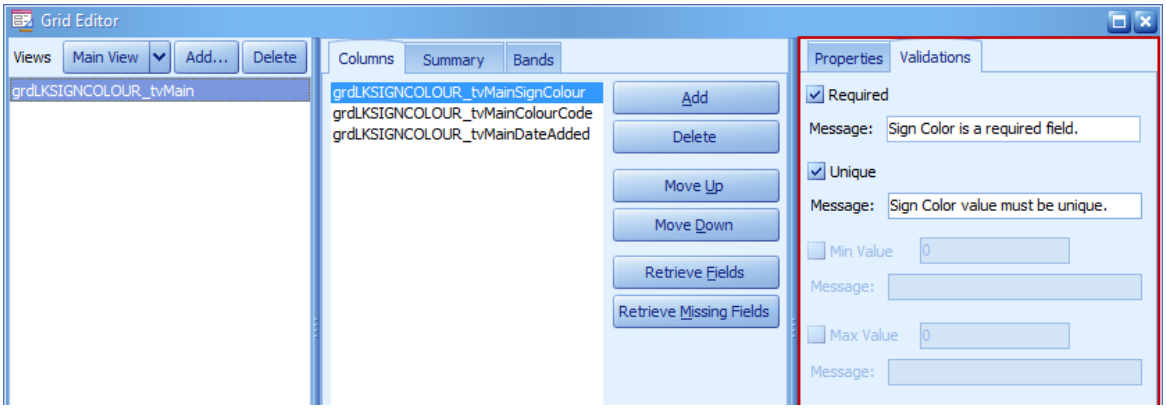
4.0 build 0

1. **Major Upgrade:** All third-party controls have been upgraded to the latest version.

2. **Major New Feature:** Added the “Show/Hide Find Panel” button to the grid toolbars. A Find Panel is an easy and straightforward way for end-users to locate information within the grid control. To execute a search, simply press the “Show/Hide Find Panel” button or press the keys “CTRL+F”, enter text within the Find box and the grid will display those records that have partially matching values in any of the columns in the grid. This is an extremely fast way to find something without knowing what column the data is in.
   This button is turned on by default when this toolbar is placed on a form however to turn it on for existing toolbars, simply set the ShowFindPanelButton to True.

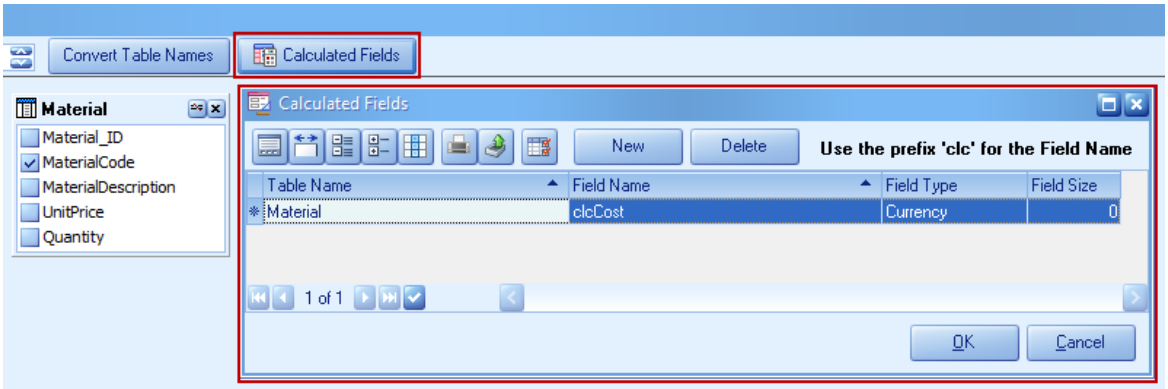


3. **Major New Feature:** In the Grid Editor dialog, added the ability to set the validations for the grid columns without scripting. Each validation can be turned on or off by the “Required” check box. A customized message can be entered for each validation, or the default message can be used. If the data type of the field is numeric then the Min and Max validation options are available.

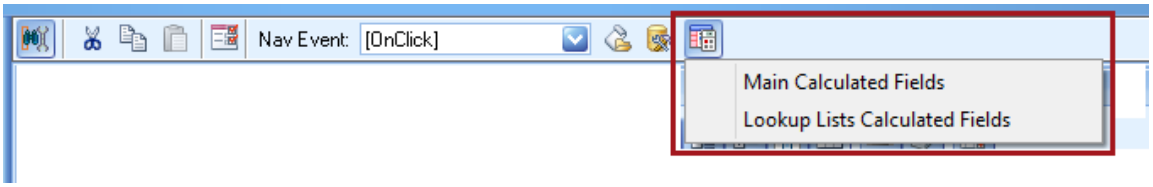


4. **Major New Feature:** Added the ability to create calculated fields for the data tables. The calculated field derives its data from the calculation of other fields. This is very useful in many situations where calculations need to be displayed in the grids and/or on the forms (using db controls) but not stored in the database.
To use this feature, follow these simple steps: in the Main Connection or Lookup Lists Connection dialog, click on the Calculated Fields button to create the calculated fields. Calculated fields can also be created in the Form Designer.
Next, in the Form Designer, link the table to the TArDBTableEvents component and double click on the OnCalcFields event to generate the event code. Then write your own code to do the calculation for the calculated field as shown below.

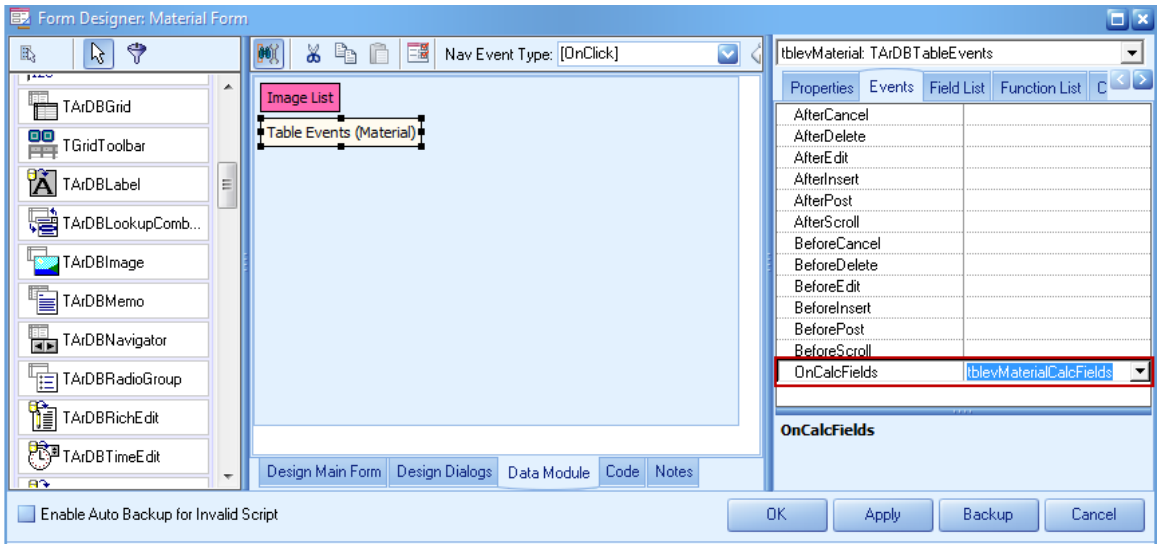
To create Calculated Fields in the Main Connection or Lookup Connection:



To create Calculated Fields in the Form Designer:

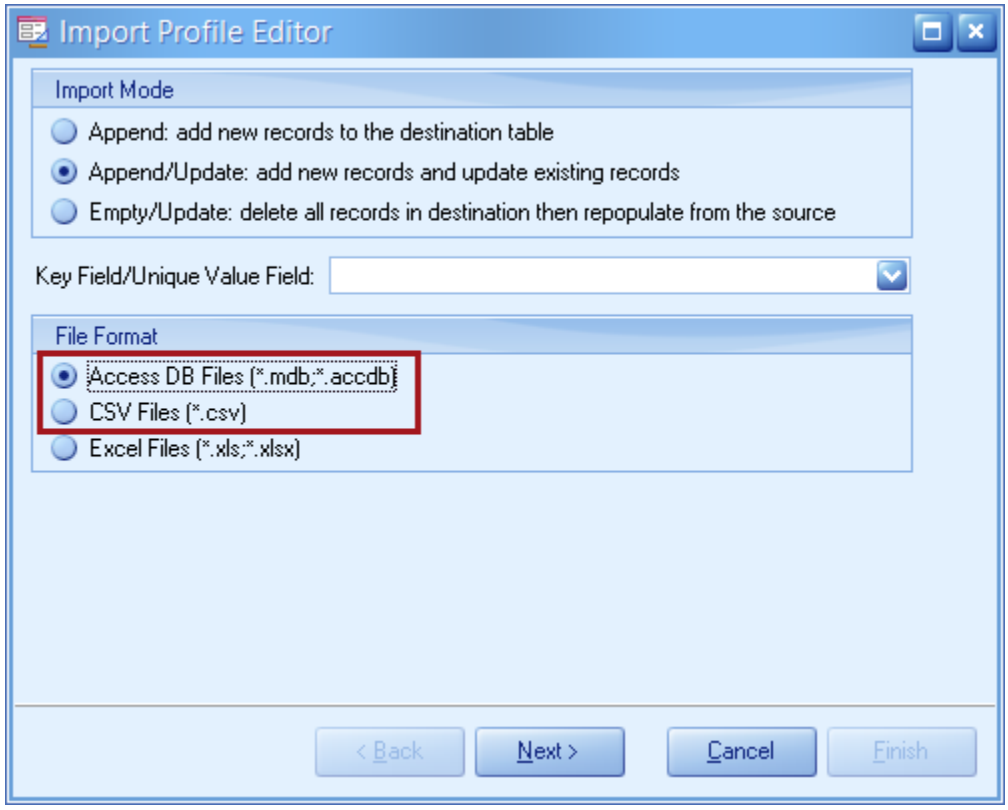


OnCalcFields Table Event:



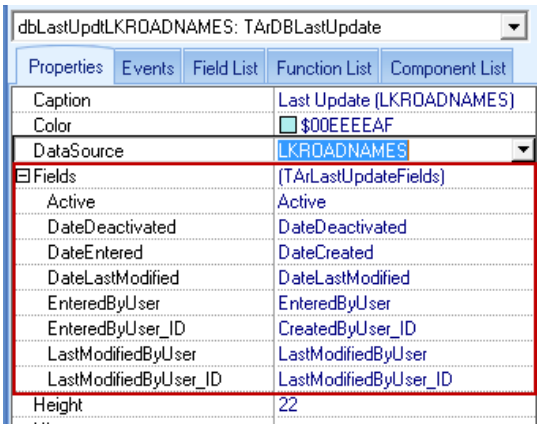

```
procedure tblevMaterialCalcFields(DataSet: TDataSet);
begin
  DataSet.FieldByName('clcCost').AsFloat := DataSet.FieldByName('UnitPrice').AsFloat *
     DataSet.FieldByName('Quantity').AsFloat;
end;
```

5. **Major New Feature:** Added support in the "TArDBImportData" component to import data from MS Access databases and CSV files.

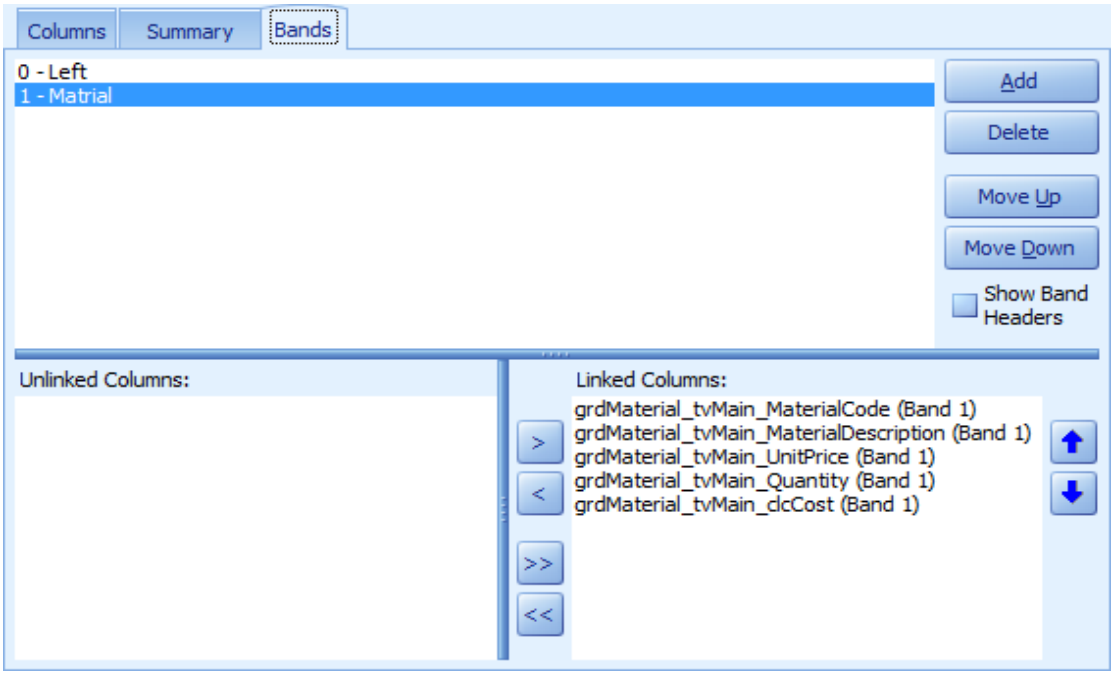


6. **Major New Feature:** Added the function "SendEmail" to the Function List which allows sending Email in scripting.

7. **Significant Enhancement:** Changed the "TArDBLastUpdate" component to include more common fields. The fields will be automatically populated when the selected DataSource has the matching field names. This enhancement eliminates the need to populate these fields in the Table Events scripts. Simply drop this component onto the Data Module and set these properties and Data Manager will take care of the rest.
   When the DataSource is selected, Data Manager will try to determine the names of these fields and automatically populate the field names however other fields can be selected from the pick lists. The "TArDBTableEvents" component also has the same feature.

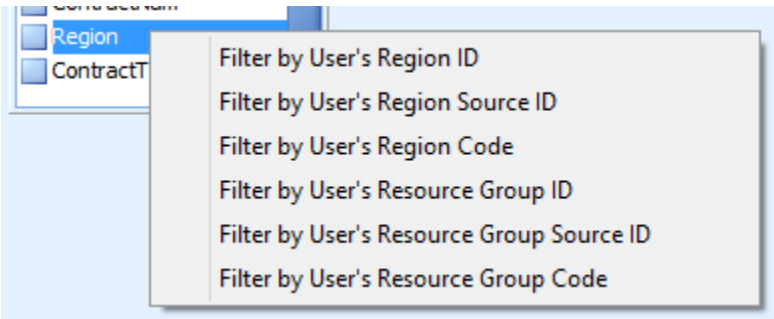


8. **Significant Enhancement:** Enhanced the Grid Editor's Bands tab to support adding and deleting bands, plus moving the band's position up and down. Also, added the dual lists for linking and unlinking the band columns.



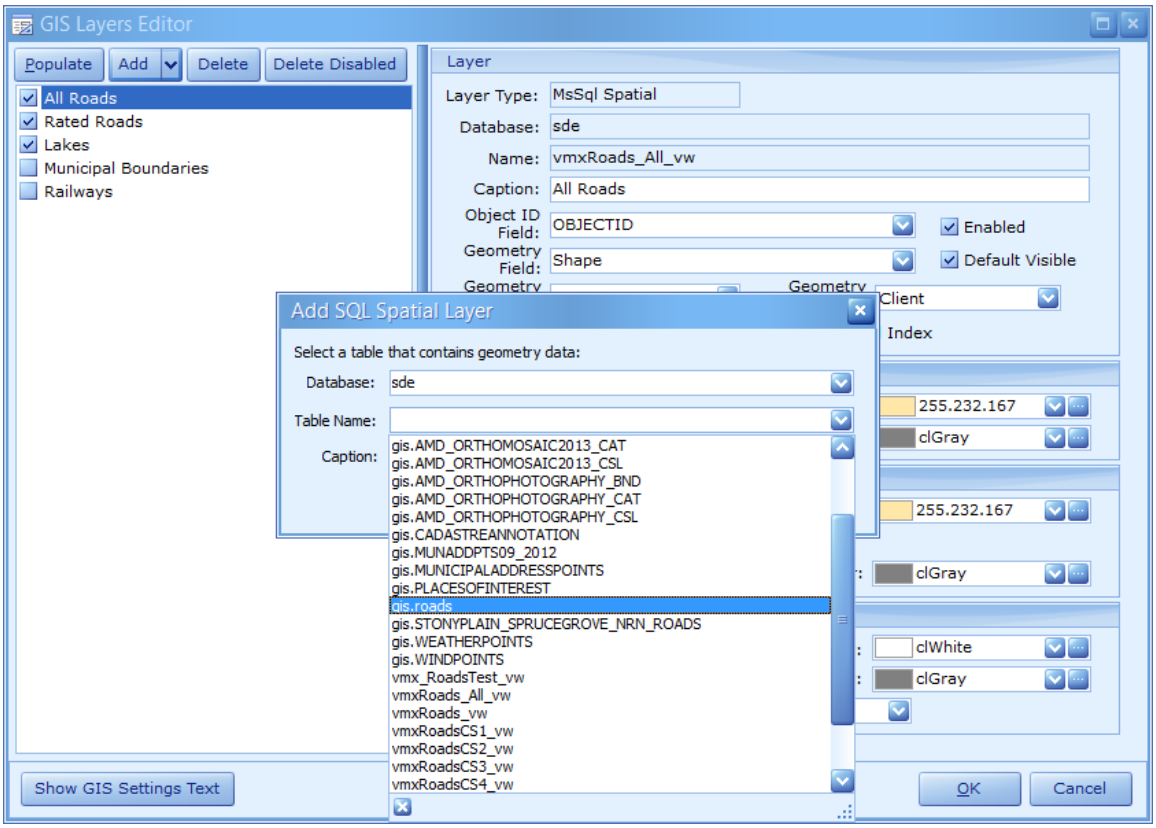
Here's an example of a grid with Bands.
Draging the band, moves all columns in that band with it. Same goes for hiding a band.

9. **Significant Enhancement:** Significantly improved the "Filter by User…" feature and added additional options as shown here:



Reminder: In version 3.0 build 0, this feature was added:
Added the ability in the Main Connection and Lookup Lists Connection screens to filter the table data by user's region or resource group. This filtering is based on the Regions and Resource Groups list in the Custom (and WRR) Control Panels. To tell Data Manager which field to use in the filtering, right click on the field to show the popup menu and check the appropriate option as shown below. Enhancements have also been made in the Regions and Resource Groups forms in the Custom (and WRR) Control Panels.

10. **Significant Enhancement:** Significantly improved all aspects of the GIS Layers Editor. Also, added support for each layer to be loaded from a separate database.

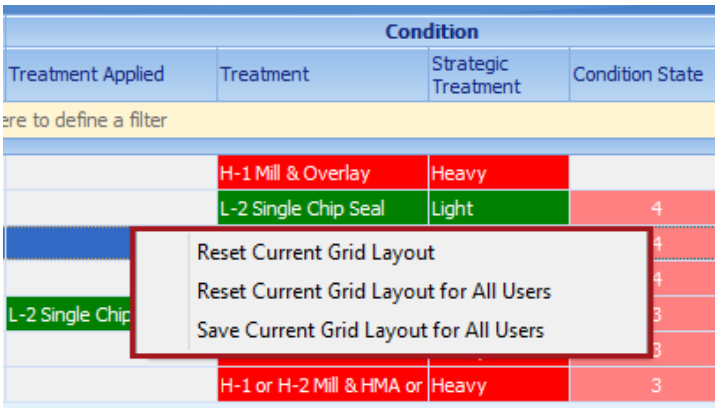


11. **New Feature:** In the Restore Configuration dialog, added the ability to pick and choose the Group Item(s) to restore from the "Full" or "Auto Full" backup.

12. **New Feature:** Added two new components to the Form Designer: TArDBToggleSwitch, TArToggleSwitch (non-data-aware). A toggle switch is a touch-friendly version of a traditional check box. A toggle switch replaces a check box with a "thumb", which can be positioned at one of available locations indicating the current checked state (on, off, or indeterminate). End-users can switch the checked state either by dragging the thumb, clicking the shaft next to the thumb, or pressing the Space key while the editor is focused.
Pressing the Space key immediately reverses the thumb position, while two other methods move the thumb with smooth animation.

StateIndicator	(TdxToggleSwitchStateIndicator)
Kind	sikGlyph
OffGlyph	(TdxSmartGlyph)
OffText	No
OnGlyph	(TdxSmartGlyph)
OnText	Yes
Position	sipInside

13. **New Feature:** Added popup menu items to the grid for users who have Data Manager Designer access to reset and save the grid layout for all users of that form.



14. **New Feature:** Added the ability to run Data Manager from any external process including Windows Scheduler. This feature is very handy if an automated process needs to run at a scheduled time.

    Here's an example of how to run Data Manager and go to a specific form and record.

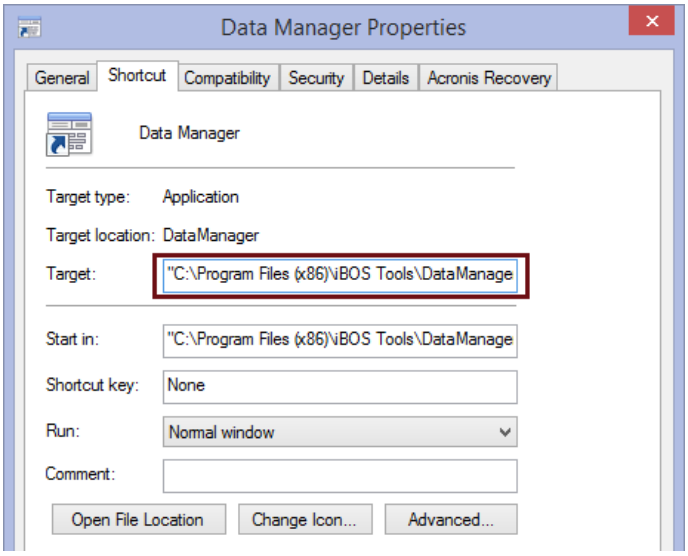
    "C:\Program Files (x86)\iBOS Tools\AR Ent 2\AREntViewer.exe" "SrcAppName=;UserName=John;Password=ABCD;ConfigName=SCR;GroupName=SCR Surfaced Roads;GroupItemName=2015 Surfaced Roads;NodeName=2015 SCR Form;NavKeyValue=00010000000000021151;TerminateApp=Yes"

See the list below for a brief description of all available parameters.

Parameter Names:

Parameter Name	Required	Description
SrcAppName=	No	Name of the application running Data Manager
UserName=	Yes	Username used to login to Data Manager
Password=	Yes	Password used to login to Data Manager
ConfigName=	Yes	Name of the Configuration to load.
GroupName=	Yes	Name of the Sidebar Group that contains the Group Item to load.
GroupItemName=	Yes	Name of the Sidebar Group Item to load.
GroupItemGUID=	No	GUID of the Sidebar Group Item to load. This is not needed.
NodeName=	Yes	Name of the System Node or Form Node to open.
NodeGUID=	No	GUID of the System Node or Form Node to open. This is not needed.
NavKeyValue=	No	The value of the Navigator Key that is used to locate the record to go to. If no key value is provided that the first record in the Navigator will be focused in a Form node.
FiscalYear=	No	Fiscal Year to go to. This is only needed for MMS 4 and PPT.
Region=	No	Region to go to. This is only needed for MMS 4 and PPT.
ResourceGroup=	No	Resource Group to go to. This is only needed for MMS 4.
ActiveTabPage=	No	Used to set focus to a specific tab on the form.
LocateTableName=	No	Table name to locate if needed.
LocateFieldName=	No	Fieldname to locate if needed.
LocateFieldValue=	No	Field value to locate if needed.
TerminateApp=	No	If set to **Yes**, then the application will terminate once the script in the OnClick event (and any functions called from the OnClick event) have completed. This is only needed if Data Manager is called from an automated process like Windows Scheduler.

The above example can be set in the “Target” property of a shortcut to Data Manager as shown here:

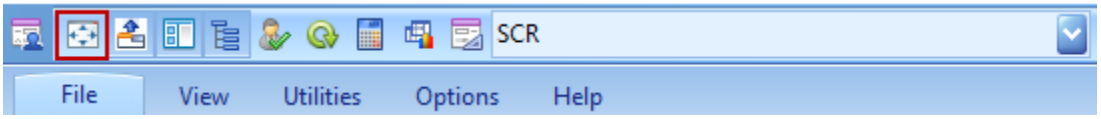


15. **New Feature:** Added the event "OnPaintShape" to the TArGISViewer component for customizing individual shapes such as color, width, outline color, outline width, etc.

    Example:

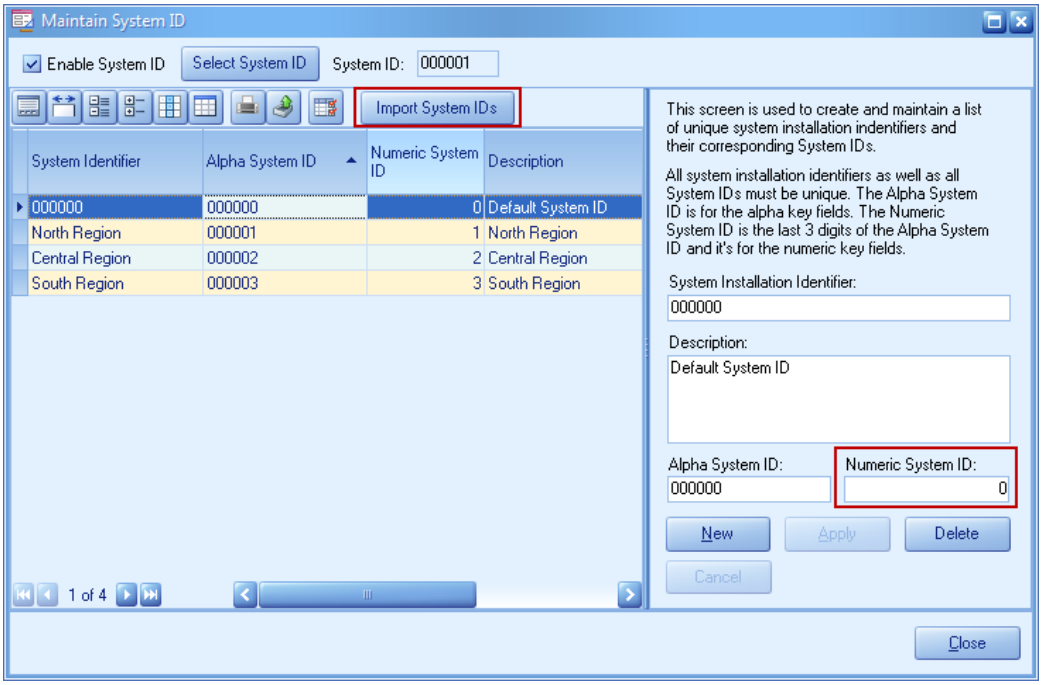
```
procedure ArGISViewer1PaintShape(_sender: TObject; _shape: TGIS_Shape);
var
  varCondState: Variant;
begin
  try
      varCondState := _shape.GetField('ConditionState');

      if (varCondState<>Null) and (varCondState = 1) then
      begin
          _shape.Params.Line.Color :=
          StringToColor(tbltblGISOptions.FieldByName('ConditionState1LineColor').AsString);
          _shape.Params.Line.OutlineColor := clWhite;
          _shape.Params.Line.Width :=
          tbltblGISOptions.FieldByName('ConditionState1LineSize').AsInteger;
      end
      elseif (varCondState<>Null) and (varCondState = 2) then
      begin
          _shape.Params.Line.Color :=
          StringToColor(tbltblGISOptions.FieldByName('ConditionState2LineColor').AsString);
          _shape.Params.Line.OutlineColor := clWhite;
          _shape.Params.Line.Width :=
          tbltblGISOptions.FieldByName('ConditionState2LineSize').AsInteger;
      end;
  except
  end;
end;
```

16. **New Feature:** Added a button to the quick access toolbar for hiding all navigation to maximize the main Form. Note: this button is disabled if a form is not loaded such as when the application is loaded. Once a form is loaded, this button is enabled automatically.



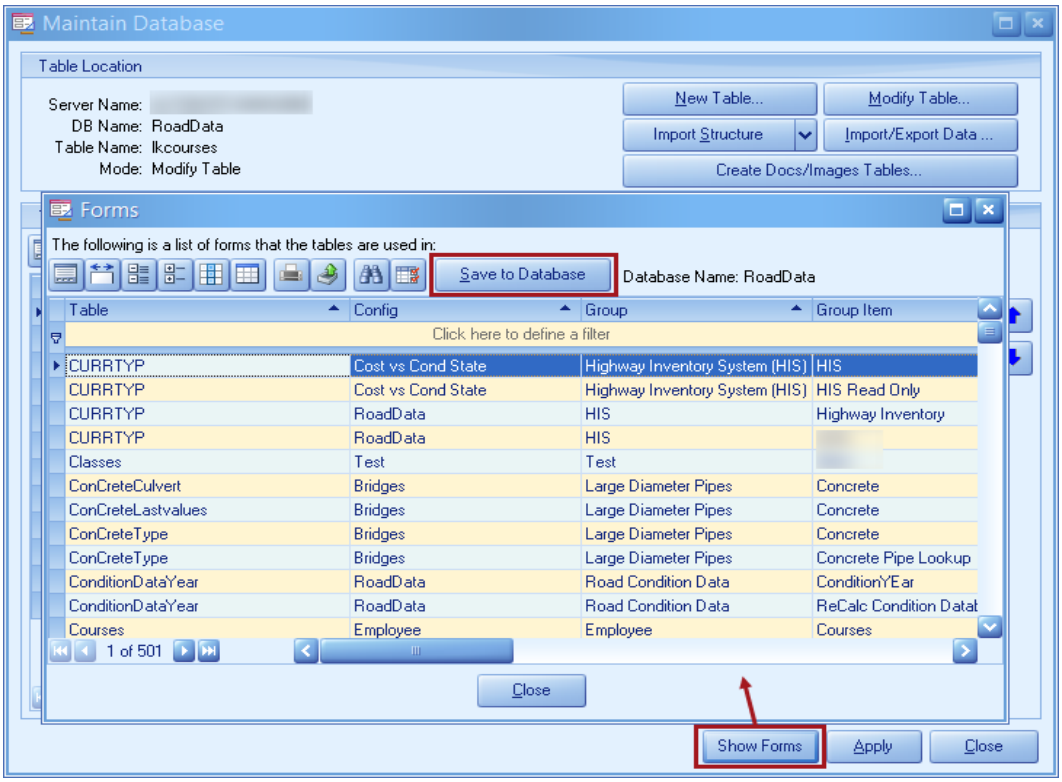
17. **New Feature:** Added a check box "Show Distinct" to the Main Connection dialog. If this box is checked and there are duplicates, only the first occurrence will show in the navigator.

18. **Enhancement:** Changed the Numeric System ID edit box to be editable and added the Import System IDs button as shown below. The Import System IDs button replaces all the data in the grid with data from an Excel file.



19. **Enhancement:** Significantly improved performance when loading images in the Select Image dialog. The Image selection dialog is now 10 times faster than the previous version.

20. **Enhancement:** Changed the "Forms" dialog in the Maintain Database utility to show a list of forms that the tables are used in for the selected database. Use the "Save to Database" to save the forms data to the table "tblARTablesUsed" in the database so that this information can be viewed in DataViewer for documenting and support purposes.



21. **Enhancement:** Added the F12 hot key to switch between the "Design Main Form" tab and the

"Code" tab in the Form Designer.

22. **Enhancement:** In the Form Designer, added popup menu items to the components TArDBImportData, TArDBDisplayFormat, TArDBEditMask, TArDBHistoryLog, and TArImageList for quick access the corresponding editors.

23. **Enhancement:** In the Grid Editor dialog, changed the buttons Retrieve Fields and Retrieve Missing Fields to automatically set the Caption for the columns added by parsing the field names with mixed case.

24. **Enhancement:** Added "Group Item Version" to the Group Item Properties dialog for versioning the Group Item.

25. **Enhancement:** Added the following two new events to the "TArDBGrid" component that are useful when the grids are displayed as Master/Detail grids or have more than one View.

    OnActiveTabChanged- occurs when a user switches to a detail level (defined by the ALevel parameter) by using tabs.

    OnFocusedViewChanged - occurs when the focused View identified by the FocusedView property is changed.

26. **Enhancement:** In the Import Configuration dialog, changed the column "Date Last Modified" to "Source Date Last Modified" and added the column "Destination Date Last Modified". Set the background to Yellow for both Date Last Modified columns when the "Destination Date Last Modified" is newer than "Source Date Last Modified". This is so that it is more obvious when the user is about to overwrite newer forms with older forms. Also added the columns "Source Group Item Version" and "Destination Group Item Version" for comparing the versions for the Group Item before importing.

27. **Enhancement:** The following functions have been added to the Functions list:

Category	Function Name	Function Type	Description
Date/Time	DateInRange	Function	Checks whether the specified TDate value is in a given range.
Date/Time	DayOfTheMonth	Function	Returns the day of the month represented by the specified TDateTime value.
Date/Time	DayOfTheWeek	Function	Returns the day of the week represented by the specified TDateTime value.
Date/Time	DayOfTheYear	Function	Returns the number of days between the specified TDateTime value and December 31 of the previous year.
Date/Time	DaysInMonth	Function	Returns the number of days in the month of the specified TDateTime value.
Date/Time	DaysInYear	Function	Returns the number of days in the year of the specified TDateTime value.

General	DebugMsg	Procedure	Show the message dialog inside a loop for debugging purposes and can abort the loop.

Date/Time	FormatDateTime	Function	The FormatDateTime function provides rich formatting of a TDateTime value DateTime into a string. Formatting is defined by the Formatting string.
			
			Examples of various formatting strings:
			
			strTimestamp := FormatDateTime('dddd d of mmmmyyyy', Now);
			
			strTimestamp := FormatDateTime('yyyy''/''mm''/''dd, hh:mm', Now);
			
			strTimestamp := FormatDateTime('yyyy''/''mm''/''dd, hh:mm:ss', Now);
			
			strTimestamp := FormatDateTime('yyyy''/''mm''/''dd, hh:mm:ss.zzz', Now);
			
			strTimestamp := FormatDateTime('yyyymmddhhmmsszzz', Now);
			
			strTimestamp := FormatDateTime('yyyy_mm_dd_hh_mm_ss_zzz', Now);
			
			strTimestamp := FormatDateTime('yyyy.mm.dd, hh:mm', Now);

Database	GetAppDBConnection	Function	Gets the application's database connection.
General	GetUserRegionCodeFilter	Function	Get the user region code filter containing the 'IN' clause with a list of the user region codes separated by comma. i.e.: IN ('1', '2')
General	GetUserRegionIDFilter	Function	Get the user region ID filter containing the 'IN' clause with a list of the user region IDs separated by comma. i.e.: IN (10000, 20000)
General	GetUserResourceGroupCodeFilter	Function	Get the user resource group code filter containing the 'IN' clause with a list of the user resource group codes separated by comma. i.e.: IN ('7101', '7102')
General	GetUserResourceGroupIDFilter	Function	Get the user resource group ID filter containing the 'IN' clause with a list of the user resource group IDs separated by comma. i.e.: IN (10000, 20000)

Date/Time	HoursBetween	Function	Returns the number of whole hours between two specified TDateTime values. Call HoursBetween to obtain the difference, in hours, between two TDateTime values. HoursBetween counts only entire hours. Thus, HoursBetween reports the difference between 9:00 A.M. and 9:59:59 A.M. as 0 because the difference is one second short of an entire hour. HoursBetween always returns a positive result and therefore the parameter values are interchangeable. Call HourSpan to obtain incomplete hours as a fraction of an entire hour.
Date/Time	IncDay	Function	Returns a date shifted by the specified number of days.
Date/Time	IncHour	Function	Returns a date shifted by the specified number of hours.
Date/Time	IncMinute	Function	Returns a date shifted by the specified number of minutes.
Date/Time	IncMonth	Function	Returns a date shifted by the specified number of months.
Date/Time	IncSecond	Function	Returns a date shifted by the specified number of seconds.
Date/Time	IncWeek	Function	Returns a date shifted by the specified number of weeks.
Date/Time	IncYear	Function	Returns a date shifted by the specified number of years.
General	MessageDlg Confirmation	Function	Displays a Confirmation message dialog box.
Database	OpenFileFromDBField	Function	Opens the file from the binary field.

General	SelectFolder	Function	Opens a dialog to select a folder. **Options:** **sdNewFolder** Displays the Make New Folder button, when using the sdNewUI option. **sdShowEdit** Display an edit box containing the currently selected folder/file. This also allows typing the name of the folder/file to select. **sdShowShares** Shareable resources on remote systems are also displayed inside the browse dialog box. This automatically enables the option sdNewUI. **sdNewUI** Use the new type of dialog, that is sizeable and includes a Make a New Folder button. The dialog also allows dragging and dropping and using the context menu on either folders or files. **sdShowFiles** The browse dialog box also displays files. **sdValidateDir** Validates the name of the folder/file inserted into the edit box, when using the bifEditBox option. The validation triggers an event of type TBrowseForFolderCallbackEvent.
General	SendEmail	Function	Sends an email using SMTP protocol. The Email Profile must be setup in Custom Control Panel before this function can be used. See the Code List for an example of how to use this function.
General	SetFieldValueWhere	Function	Set the field value for a field in the table based on the given where clause. Note: a where clause must be provided. i.e. the strWhereClause parameter cannot be blank. Caution must be taken when using this function because it updates the field in the table based on the where clause. If the where clause is not correct then the wrong data may be updated.

General	ShowYearlyCalendar	Function	Shows the Yearly Calendar for selecting a set of days within the 12-month period. See the Code List for a detailed example of how to use this function.
Database	StoreFileInDBField	Function	Shows the Select File dialog, then stores the selected file in the binary field. It also stores the file path and file name information if the corresponding parameters are provided. The parameters extWarningFileSize and extMaxFileSize are in MB (Mega Byte). To skip doing the check on file size, set these parameters to zero.
Date/Time	TryStrToDate	Function	Call TryStrToDate to parse a string that specifies a date. If strDateTime does not contain a valid date, TryStrToDate returns False. The conversion uses the format specified by the ShortDateFormat.
Date/Time	TryStrToTime	Function	Call TryStrToTime to parse a string that specifies a time value. If strDateTime does not contain a valid time, TryStrToTime returns False. The conversion uses the format specified by the ShortDateFormat.
Date/Time	WeeksInYear	Function	Returns the number of weeks in the year of the specified TDateTime value.

28. **Modification:** Added the function "ResetSQL" to the "TArSQLContainer" component. Must call this function to reset the SQL back to the original before calling the function "SetParamValue".

    Set parameter value example:

    ```
    sqlRoadNames.ResetSQL;
    sqlRoadNames.SetParamValue('@RoadName', '%MUA', False);
    sqlRoadNames.SetParamValue('@SignCrewBU', '6220', False);
    qryTemp.SQL.Text := sqlRoadNames.SQL.Text;
    ```

29. **Enhancement:** Added new functions to the MemList object. The MemList object is used to store unlimited number of variables in memory and have them available anywhere in the script. E.g. values can be stored in the Navigator's OnClick event then used in the main script. Simply use "Set…" to store a value then use "Get…" to get that value based on a Key. The Key can be anything as long as each variable uses a unique key for that form. When using the "Get…" functions, a default must be set in case the value has not been stored. The values stored in the MemList are only stored in memory and is automatically reset when Data Manager is restarted or when F5 is pressed or when the form is reloaded. The functions shown below can be used in scripting.
    MemList functions:

```
MemList.
```

```
procedure Clear
function  Delete(strKey: String)
function  GetBool(strKey: String; bolDefault: Boolean)
function  GetDate(strKey: String; dtDefault: TDateTime)
function  GetFloat(strKey: String; extDefault: Extended)
function  GetInteger(strKey: String; intDefault: Integer)
function  GetObject(strKey: String; objDefault: TObject)
function  GetString(strKey: String; strDefault: String)
function  SetBool(strKey: String; bolValue: Boolean)
function  SetDate(strKey: String; dtValue: TDateTime)
function  SetFloat(strKey: String; extValue: Extended)
function  SetInteger(strKey: String; intValue: Integer)
function  SetObject(strKey: String; objValue: TObject)
function  SetString(strKey: String; strValue: String)
```

30. **Enhancement:** Added the following scripting examples to the Code List:
    - Find First to search for files in a folder
    - Show Image List
    - Show Yearly Calendar

31. **Issue:** In the Form Designer, an error may occur when changed from one form to another.
    Status: This issue is resolved.

32. **Issue:** In the Form Designer, the Grid Editor cannot be shown after a chart has been created within the grid.
    Status: This issue is resolved.

33. **Issue:** The selected navigator node is not visible after the Refresh (F5) button is pressed.
    Status: This issue is resolved.

34. **Issue:** An error occurs when trying to import the configuration that contains data for the User Filters.
    Status: This issue is resolved.

35. **Issue:** The “TArDBRadioGroup” component doesn’t do the validation when the property Required is set to True.
    Status: This issue is resolved.

3.0 build 3

1. **Issue:** The Advanced Filter dialog shows an empty pick list for the Operator column when the selected field’s data type is Date.
   Status: This issue is resolved.

2. **Issue:** Some of the buttons in the Grid Editor are not working properly.
   Status: This issue is resolved.

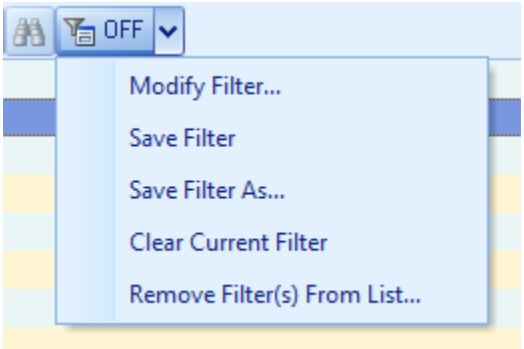
3.0 build 2

1. **Enhancement:** Added indexes to the Data Manager database to improve performance on loading Data Manager and Data Manager Designer.

2. **Issue:** Cannot export multiple large tables with more than one hundred thousand records all at once in Data Transfer.
   Status: This issue is resolved.

3.0 build 1

1. **Enhancement:** Replaced the Navigator Filter drop-down menu with a more advanced one for a more consistent look with the rest of the program.



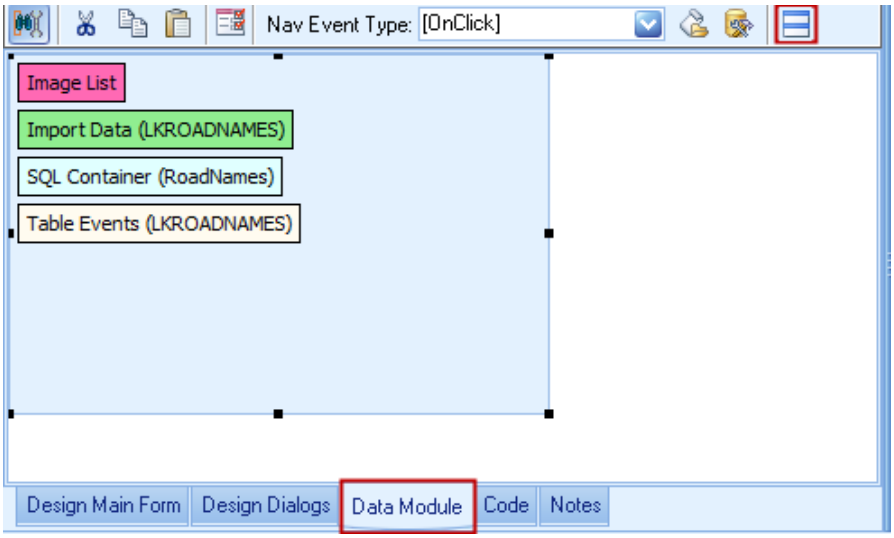
2. **Issue:** Data Manager required Full Access to the registry key for Component Object Model (COM) which is an essential part of Data Manager module for all the import/export functionalities.
   Status: This issue is resolved. Data Manager only needs Read Access to the registry key for COM.

3. **Issue:** The Navigator is shown when the Show Sidebar button is pressed even the Navigator is set as not to show.
   Status: This issue is resolved.

4. **Issue:** The Google Map is not showing the street view icon.
   Status: This issue is resolved.

3.0 build 0

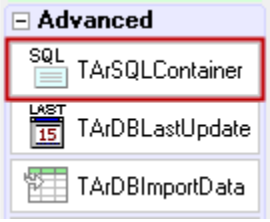
1. **Major New Feature:** In the Form Designer, we added the "Data Module" tab for placing the non-visual controls. Use the "Tile Controls" button on the toolbar to tile the controls on the "Data Module" tab in a nice organized list as shown below. If a non-visual control is placed on the "Design Main Form" tab or "Design Dialogs" tab, it will be automatically moved to the "Data Module" tab. The non-visual controls are TArTimer, TArGPS, TArImageList, TArSQLContainer, TArDBDisplayFormat, TArDBEditMask, TArDBHistoryLog, TArDBLastUpdate, TArDBTableEvents, and TArDBImportData.



2. **Major New Feature:** Added the "TArSQLContainer" component to the Form Designer for storing a SQL statement that can be assigned to a Query as shown below. The SQL statement can be edited in the SQL Editor dialog. The SQL Editor dialog can be accessed by right clicking on the component to show the popup menu and then left click on the "SQL Editor" menu item. The "Query Builder" button on the SQL Editor dialog is for showing the Query Builder tool. The Query Builder is for building the SQL statement with an interactive and user-friendly interface. This component also supports parameters in the SQL statement. Use the symbol "@" as the prefix for the parameter name. In scripting, call the "SetParamValue" function the set the parameter value in the SQL statement. This component minimizes the need to add each of the lines of SQL statement to a Query in scripting.
To assign the SQL Statement to a Query, simply do the following:

```
qryTemp.SQL.Text := sqlRoadNames.SQL.Text;
qryTemp.Open;
or
qryTemp.Execute;
```

The parameters of the SetParamValue function are:
SetParamValue(strParamName: String; strParamValue: String; bolIsNumeric: Boolean): Boolean;



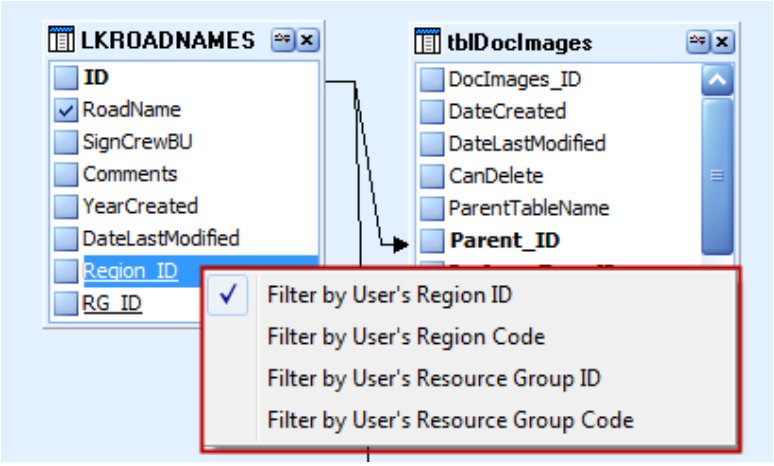
SQL statement example:

```
SELECT*FROM tblRoadNames
WHERE RoadName LIKE @RoadName AND Crew = @Crew
```

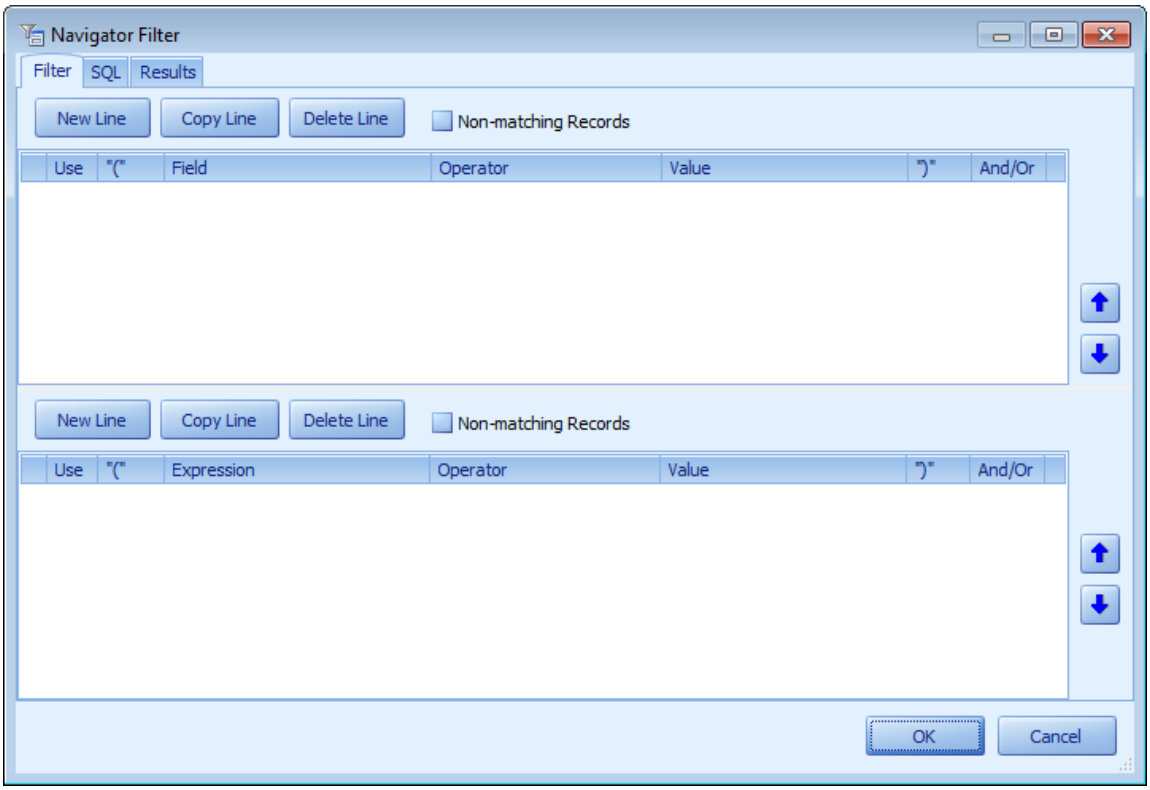
Set parameter value example:

```
sqlRoadNames.SetParamValue('@RoadName', '%MUA', False);
sqlRoadNames.SetParamValue('@SignCrewBU', '6220', False);

qryTemp.SQL.Text := sqlRoadNames.SQL.Text;
```

3. **Major New Feature:** Added the ability in the Main Connection and Lookup Lists Connection screens to filter the table data by user's region or resource group. This filtering is based on the Regions and Resource Groups list in the Custom (and WRR) Control Panels. To tell Data Manager which field to use in the filtering, right click on the field to show the popup menu and check the appropriate option as shown below.

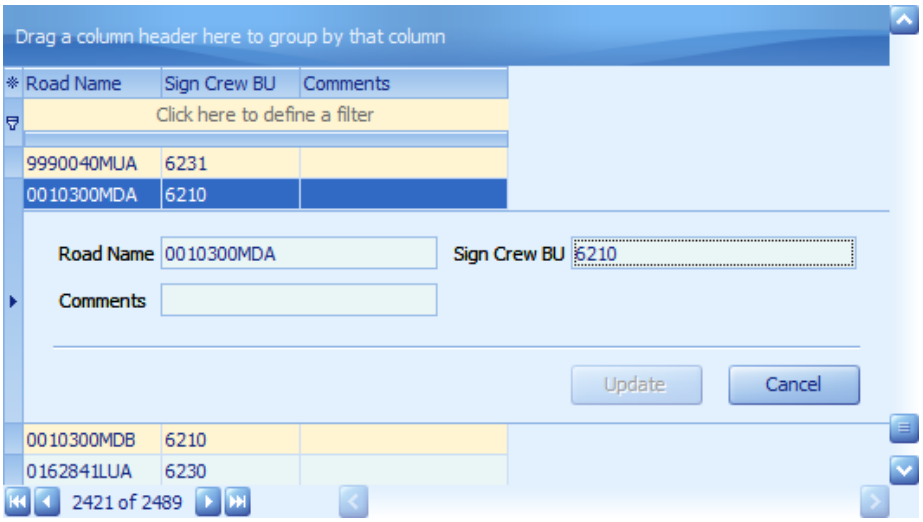


4. **Major Enhancement:** Changed the Navigator Filter dialog to the Advanced Filter dialog which has the same features as the one from DataViewer.



5. **Upgrade:** The Third-party controls have been upgraded to the latest version.

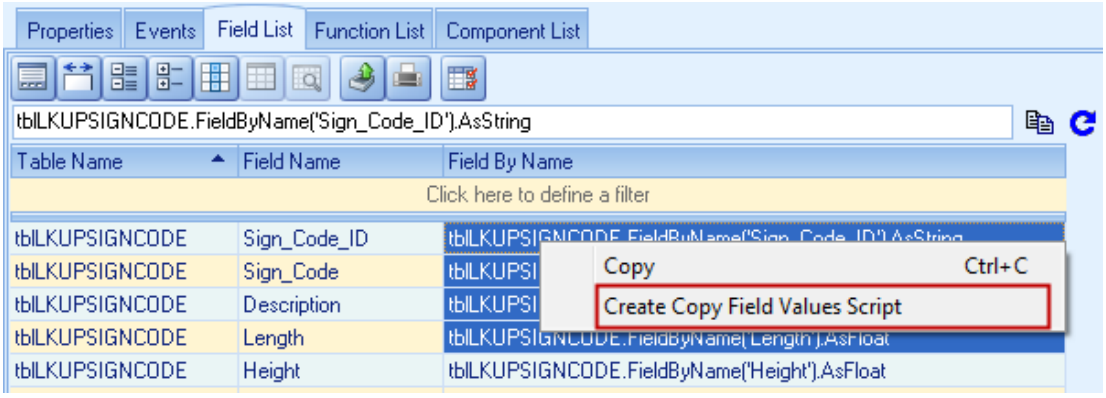
6. **New Feature:** The TArDBGrid component now allows data editing with built-in edit form. To enable this feature, simply set the property OptionsBehavior\EditMode to emInplaceEditForm. In Viewer mode, if the users want to edit the data in this form layout, they can simply double click on the row to show the edit form for the selected record or press the Enter key or press the F2 key. This is handy in editable grids with a lot of columns. When the Edit Mode is set to use the edit form, the grid cells are read only. You can only edit the values in the edit form.



7. **New Feature:** Added the Show Forms button to the Maintain Database dialog for showing all the forms that the selected table is used in.

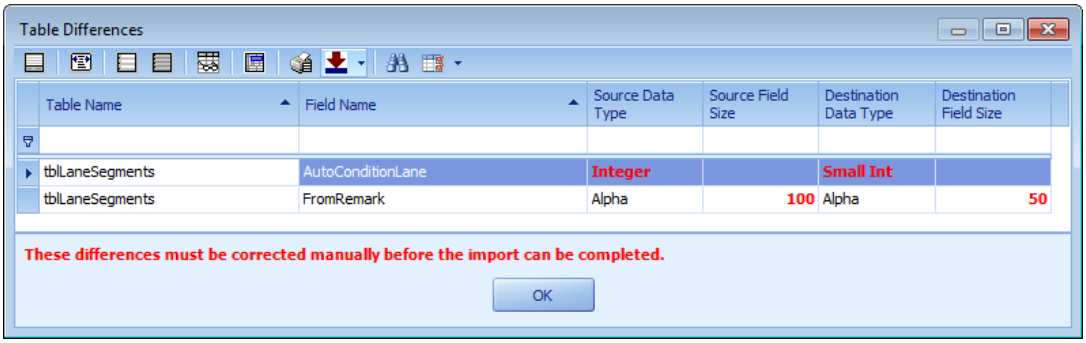
8. **New Feature:** Added the Refresh button to the Form Designer for refreshing the current form. This button is only shown when the current form is being locked by another user. The "Go to Line #" edit box beside the Refresh button is for going to the line number in code after the refresh. If the number is greater than zero, it will go to the specified line number; otherwise, it will stay on the current line number.

9. **New Feature:** Added the "Create Copy Field Values Script" menu item to the Field List popup menu for generating a script like the one shown below.



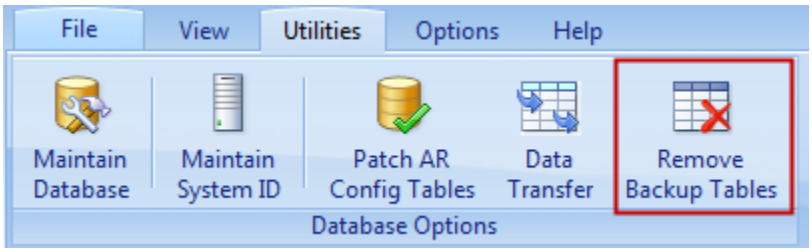

```
tblLKUPSIGNCODE.FieldByName('Sign_Code_ID').Value := qryTemp.FieldByName('Sign_Code_ID').Value;
tblLKUPSIGNCODE.FieldByName('Sign_Code').Value := qryTemp.FieldByName('Sign_Code').Value;
tblLKUPSIGNCODE.FieldByName('Description').Value := qryTemp.FieldByName('Description').Value;
tblLKUPSIGNCODE.FieldByName('Length').Value := qryTemp.FieldByName('Length').Value;
```

10. **New Feature:** Added the "Table Differences" report to the Data Transfer utility to display all the differences between the source tables and the destination tables.

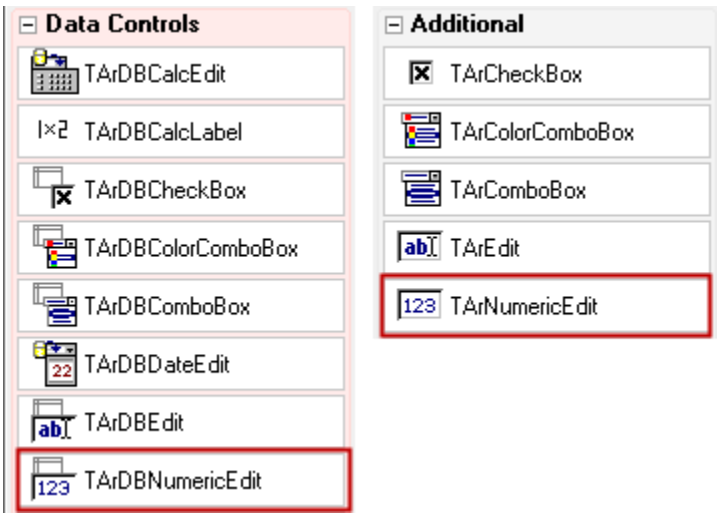


11. **New Feature:** Data Manager now automatically backs up tables (structure and data) in various situations. These backup tables are prefixed with: zzvmxBAK_. The prefix is to ensure the backup table names are unique. The "zz" part is simply to ensure that the backup tables appear at the bottom of the list of tables so they are out of the way.

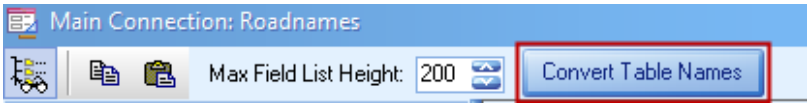
12. **New Feature:** In Data Manager Designer, added the "Remove Backup Tables" button to the ribbon toolbar to cleanup old backup tables if they are no longer needed. This button is enabled when the setting "User Can Remove Backup Tables" is set to Yes.

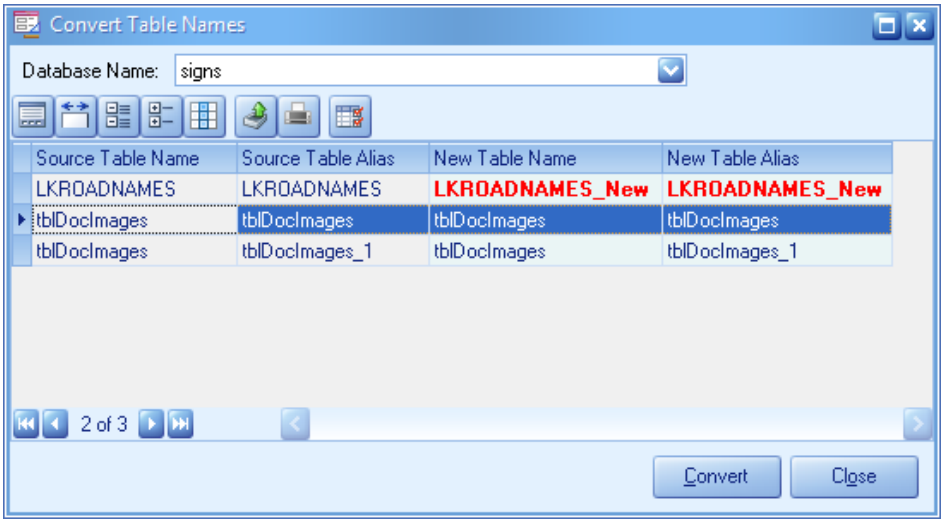


13. **New Feature:** Added the "TArDBNumericEdit" data-aware control and "TArNumericEdit" non-data-aware control for editing and displaying numeric values.



14. **New Feature:** Added the "Convert Table Names" button to the Main Connection and Lookup Lists Connection toolbars for converting table names for the selected node, including all the references in the form and the script.





15. **Enhancement:** The Export and Import process in the Data Transfer utility has been enhanced to overcome a 32-bit limitation on large tables.

16. **Enhancement:** In Maintain Database screen, added the menu items "Utilities\Rename SQL View" for renaming SQL view in the database and "Utilities\Delete SQL View" for deleting SQL view in the database.

17. **Issue:** Copy and paste the TArImageList component doesn't retain the images in the image list.
    Status: This issue is resolved.

18. **Issue:** The SQL statement for the navigator doesn't contain the correct field sort order when multiple fields are selected for the same tree level.
    Status: This issue is resolved.

19. **Enhancement:** Added two new settings for the Data Transfer utility:
    - User Can Do Empty & Update in Data Transfer
    - User Can Delete Tables in Data Transfer

    When emptying or deleting tables in Data Manager, the entire tables are automatically backed up with the prefix: zzvmxBAK_.

20. **Enhancement:** Changed the TGridToolbar component to use more modern images for the buttons. Set the property **ShowNewSize** to True to make the buttons fully fit the images. The ShowNewSize defaults to False for existing grid toolbars so that the toolbar does not overlap with other controls. The ShowNewSize defaults to True for new grid toolbars that are added to the form. We recommend that the ShowNewSize property be set to **True** for all grid toolbars so that the images look better and are not cut off.

21. **Enhancement:** Improved the display of the required field validation messages on the Message Panel instead of popping up one message dialog per field.

22. **Enhancement:** In the Grid Editor dialog, using the menu item "Create Banded Table View" or "Create Table View" will keep all the existing columns and their properties.

23. **Issue:** The function IntToStr() doesn't work when passing in the Int64 parameter value.
    Status: This issue is resolved.

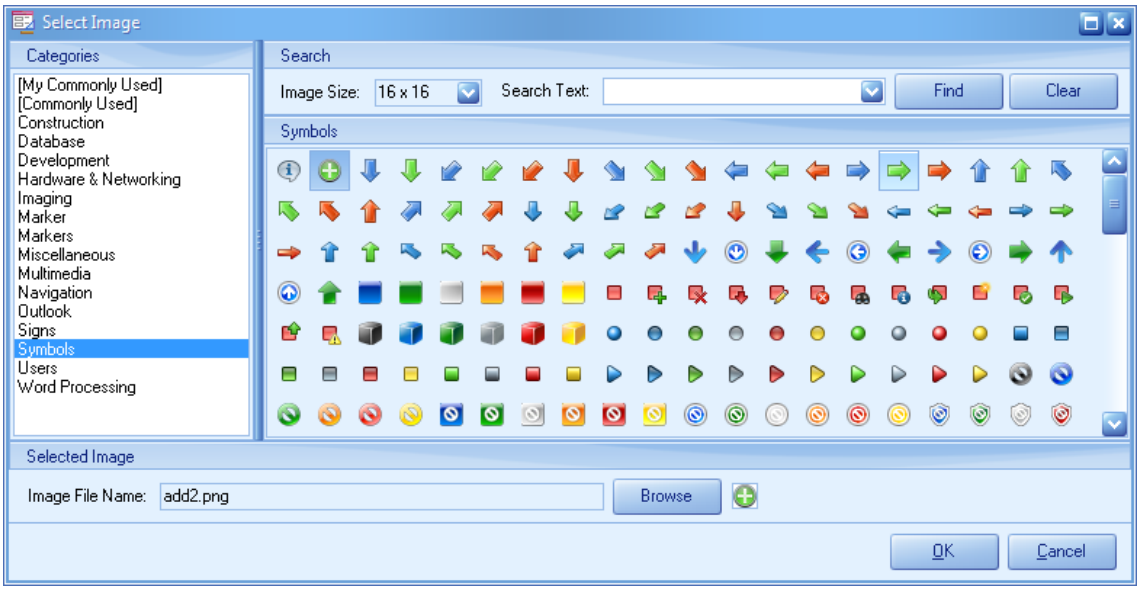
24. **Issue:** The Maintain Database utility causes error when trying to delete a field that has an index with non-standard index name linked to it.
    Status: This issue is resolved.

25. **Issue:** The Maintain Database utility doesn't work on renaming a field when the new field name is the same as the old field name except the character case is different.
    Status: This issue is resolved.

26. **Issue:** The info in the status icon tab did not copy over when copying a form from one node to another.
    Status: This issue is resolved.

2.7 build 3

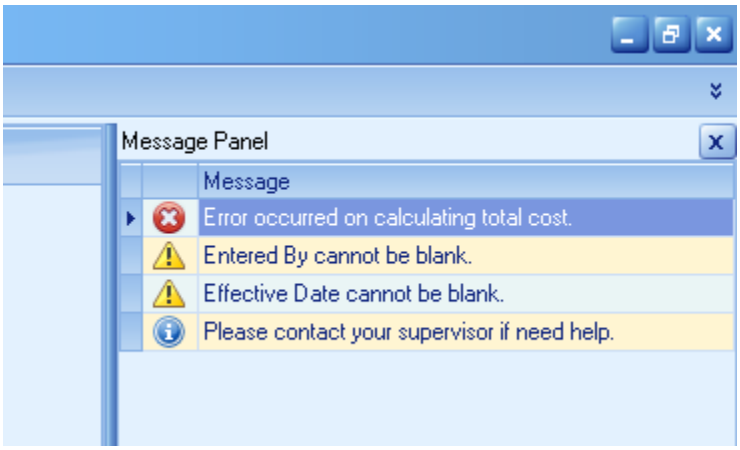
1. **Major New Feature:** The Select Image dialog now supports a library of high quality images which are organized in categories and the images are searchable.



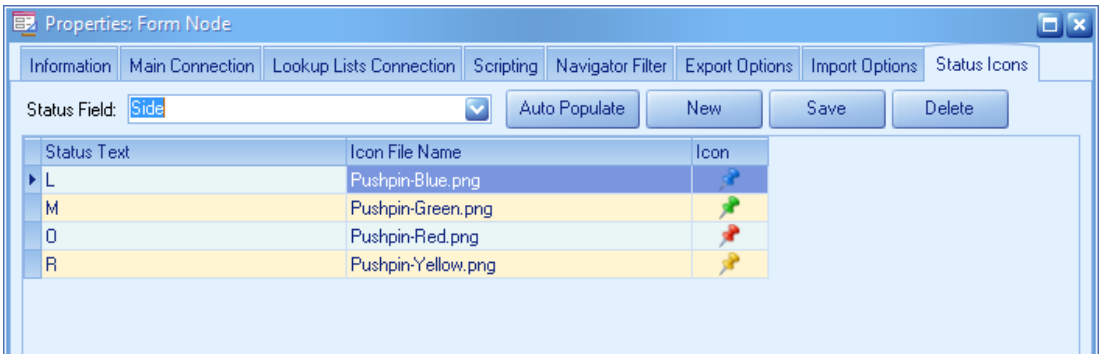
2. **Major New Feature:** In Data Manager, added the Message Panel to the right side of the main screen for displaying custom messages. This feature is very handy for providing user validation information in a list rather than displaying one message at a time. Plus, the users can access the form and display the messages at the same time.

   Function/procedures for the Message Panel:

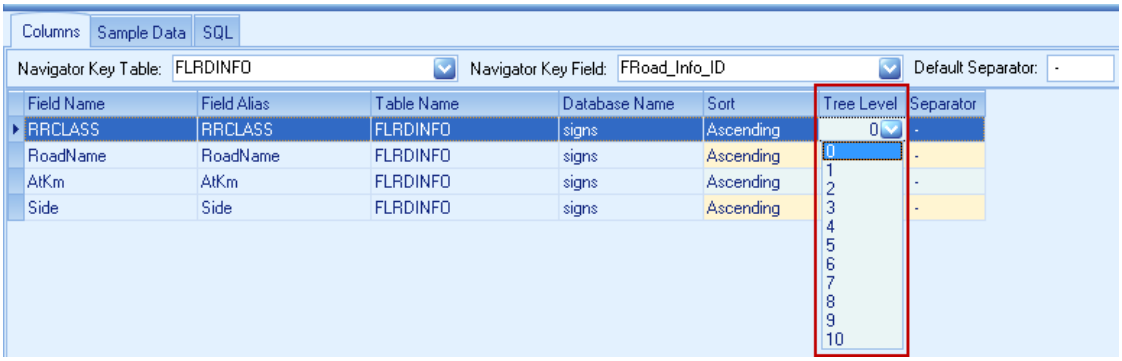
   procedure ShowMessagePanel;
   procedure HideMessagePanel;
   procedure ClearMessages();
   procedure ClearMessages(messageType: TMessageType);
   procedure AddMessage(messageType: TMessageType; strMessageIdentifier: String; strMessage: String);
   function MessagesExist(messageType: TMessageType): Boolean;



3. **New Feature:** Added the Status Icons tab to the Form Node Properties dialog for specifying icons for each status text value. The Status Field pick list contains a list of selected fields for the navigator.

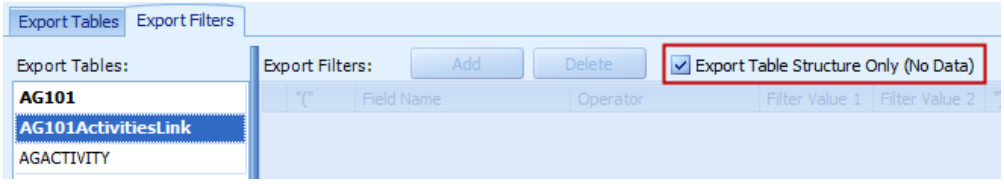


4. **Enhancement:** In the Main Connection dialog for the Form Node, you can specify the selected field as a visible field in the tree by selecting the tree level value from 1 to 10, or specify the selected field as hidden field by selecting the tree level value of 0. The purpose of hidden fields is to be able to select one in the Status Field pick list on the Status Icons tab.

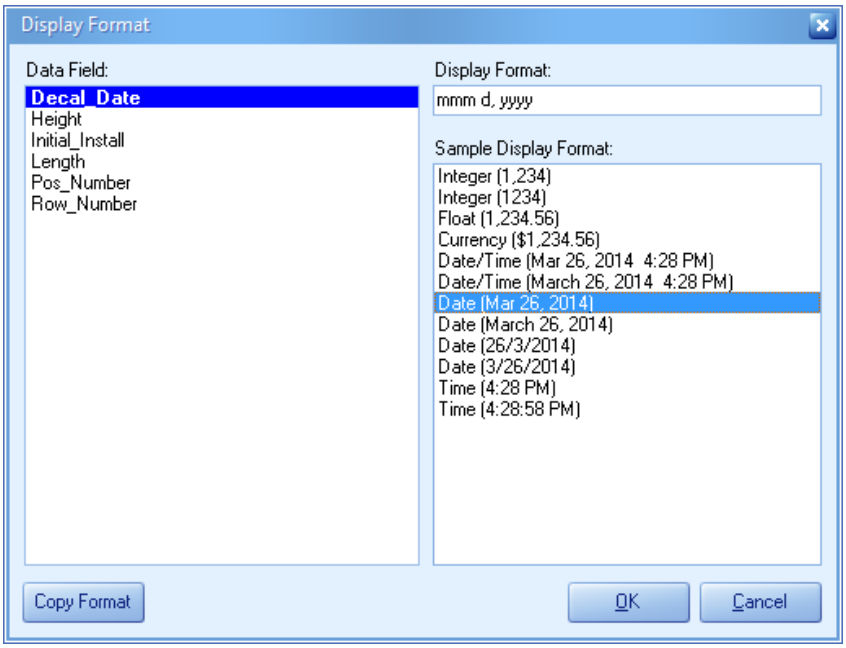


5. **Enhancement:** Improved the usability of the TArGISViewer component. The map settings are being saved automatically for each user. Right click on the map to show the popup menu "Reset Map Settings". Click on this menu to reset the map settings for the current user.

6. **New Feature:** Added the "Export Table Structure Only (No Data)" check box to the DataTransfer Export screen for exporting the table structure only without including any data.



7. **Enhancement:** Added the "Copy Format" button to the Display Format dialog in Form Designer for copying the selected format to other fields that have the same data type. Additional Display Formats were also added.



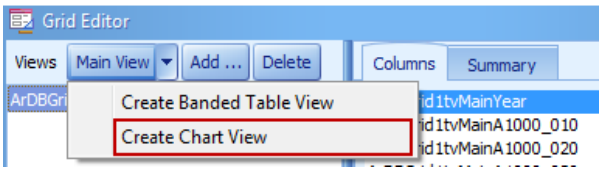
8. **Enhancement:** In the Maintain Database dialog, the "Create Docs/Images Tables" allows the user to create a new docs/images table with a different name if the table tblDocImages already exists.

2.7 build 2

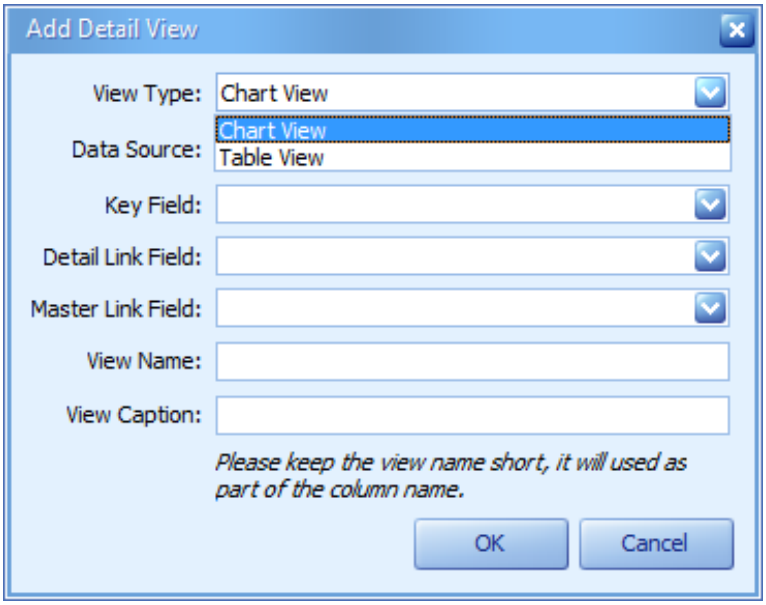
1. **Upgrade:** The Third-party controls have been upgraded to the latest version.

2. **New Feature:** The TArDBGrid component now supports Chart View which allows the users to display complex numerical data in an extremely easy to understand but very informative manner. A chart diagram represents a way of displaying data in Chart Views. There are five diagram types available: Column, Bar, Area, Line, and Pie. Both simple and multiple (stacked) diagrams can be drawn.

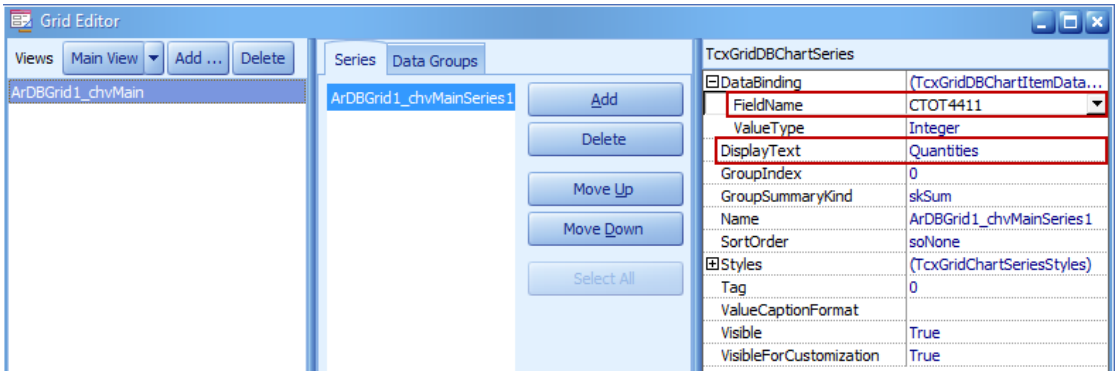
   To create a Chart View for the Main View, press the MainView\Create Chart View menu from the Grid Editor.



   To create a Chart View for the Detail View, press the Add button to show the Add Detail View dialog and then pick the view type "Chart View".



For each series, set the DataBinding\FieldName property for the Y axis value and set the DisplayText property for the Y axis title.



In Form Designer, select the TcxGridDBChartView component from the Component List then set the Categories\DataBinding\FieldName property for the X axis value and set the Categories\DisplayText property for the X axis title. Finally, set the Title\Text property for the chart title.

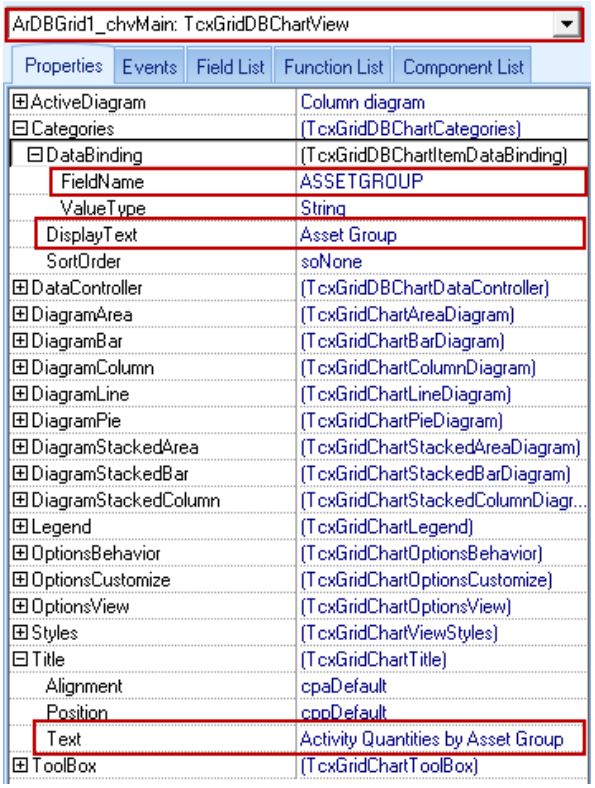
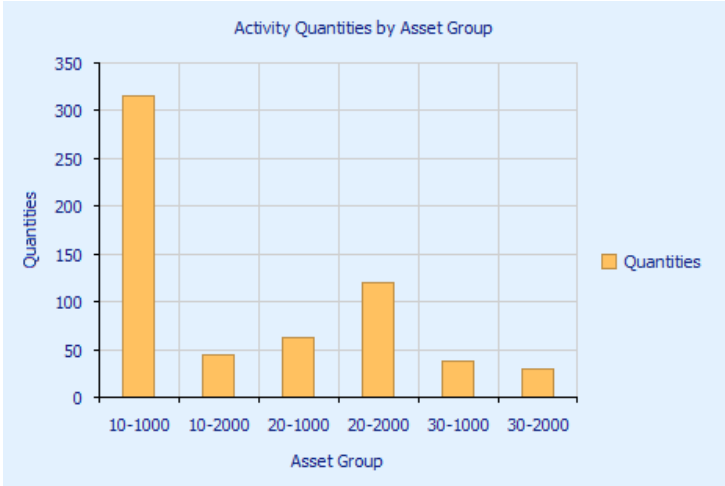
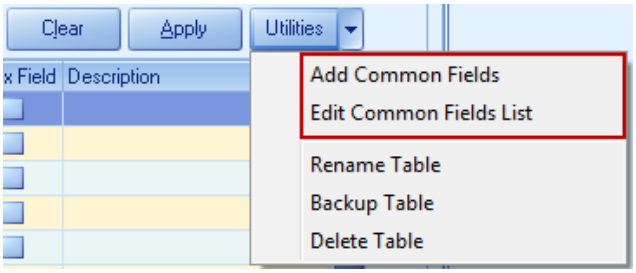


Chart display in Data Manager:

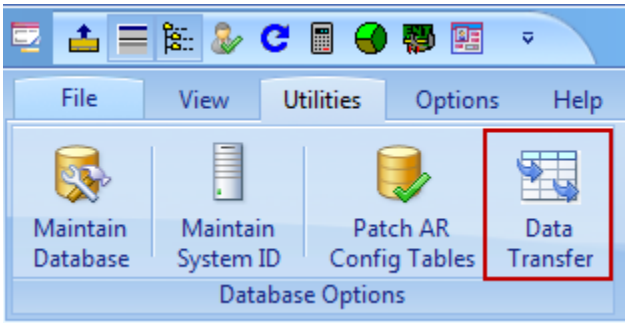


3. **New Feature:** In the Maintain Database screen, added two new menu items: “Add Common Fields” and “Edit Common Fields List” to the Utilities menu. The Add Common Fields feature shows a check-list which allows the user to multi-select which ones to add. The common fields will be added to the bottom of the field list. The Edit Common Fields List feature shows a dialog for editing\adding categories and editing\adding common fields for each category.



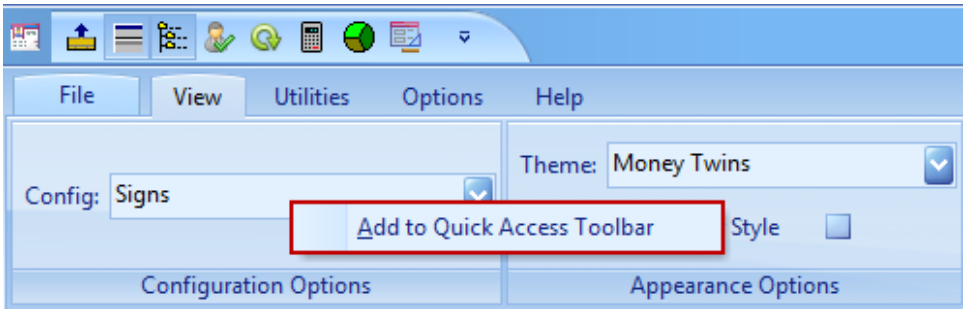
4. **New Feature:** Added the "TArDBImportData" component for importing data for a specific table from the Excel file. Set the TArDBImportData.ImportProfile property in Form Designer and call the function TArDBImportData.RunImport in the script to show the Import Data dialog in viewer mode.

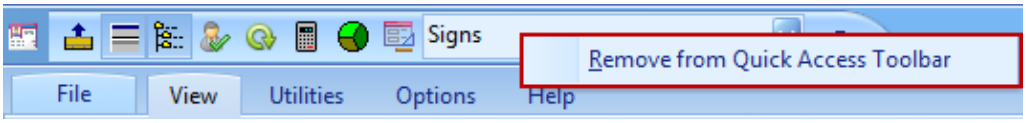
5. **New Feature:** Added the "Data Transfer" button to the Utilities tab in the ribbon toolbar. This button is enabled when the security setting "User Can Run Data Transfer" is set to Yes for the current user. Data Transfer is a utility allowing users to export or import data for the tables. The Export option supports filtering for individual table. The Import option supports creating the tables if they don't exist and also adding new fields if they don't exist.

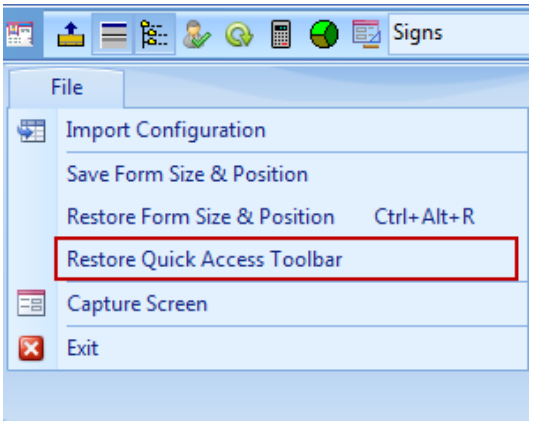


6. **New Feature:** Added the "CurrentUserInfo" global object to the Code Editor. In scripting, it has access to the information of the current user such as CurrentUserInfo.UserName, CurrentUserInfo.FullName, CurrentUserInfo.EmailAddress, etc. Also, there is a function *GetUserInfoByID(intUserID: LargeInt): TUserInfo* for getting the user information for a specific user.

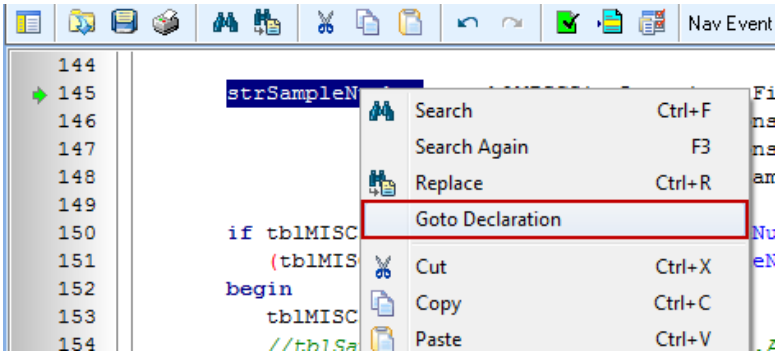
7. **New Feature:** User can add any item from the Ribbon to the Quick Access Toolbar by right-clicking on the item and press the "Add to Quick Access Toolbar" menu. To remove an item from the Quick Access Toolbar, right click on the item and press the "Remove from Quick Access Toolbar" menu. To restore the Quick Access Toolbar to default settings, press the "File\Restore Quick Access Toolbar" menu.



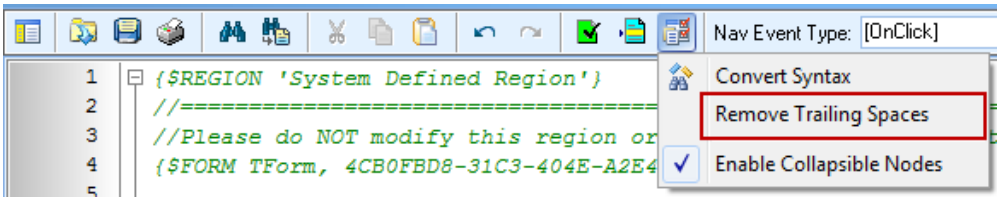




8. **New Feature:** Added the "Goto Declaration" menu item to the code editor popup menu. If the selected text is a variable name, then press the "Goto Declaration" menu item to go to the variable declaration. If the selected text is a procedure or function name, then press the "Goto Declaration" menu item to go to the procedure or function declaration.



9. **New Feature:** Added the "Remove Trailing Spaces" menu item to the code editor toolbar. Press the "Remove Trailing Spaces" menu item to remove the trailing spaces in the code.



10. **Enhancement:** Data Manager will automatically log the exception error providing that the form hasn't handled the exception in the script by using the "try...except" code. If you need to manually log the exception error in the script, call the method*LogError(strFunctionName, strErrorCode, strErrorType, strErrorDescription, strSystemError: String; bolDisplayMessage: Boolean)*.

11. **Enhancement:** The Maintain Database screen is now resizable and the screen size is being saved and restored for the user.

12. **Enhancement:** Added function *TArGISViewer.AddSQLLayer(strTableName: String; strLayerCaption: String; strFilter: String; strParams: String)* to the TArGISViewer component for adding SQL layer at runtime. The parameters strFilter and strParams are optional.

13. **Enhancement:** The Export Form Data and Import Form Data features now support the table containing column data type "geometry".

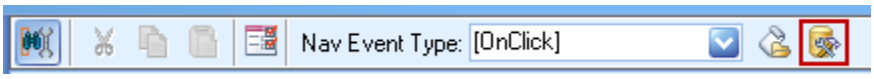
14. **Enhancement:** The Main Connection dialog draws the table links with black arrows for the left joins and white arrows for the right joins. You can now visually see what kind of join is assigned to each link which makes it much easier to verify if the joins are correct or not. Also for creating a new link between two tables, always drag the link field from the master table to the detail table. If you do it in this order, it will automatically figure out the correct join type and set it for the new link.

15. **Enhancement:** Added the Active column to the User Filters screen which allows the user to de-activate the user filters by un-checking the Active value.

16. **Enhancement:** In scripting, you can use the LastExceptionMessage value to display the error message in the try/except code block.

```
try
        //Do something ...
  except
        MessageDlg('Error occurred when processing data.' + #13 +
        LastExceptionMessage, mtError, SetOf([mbOK]), 0);
  end;
```

17. **Enhancement:** In Form Designer, added the Maintain Database button to the toolbar for showing the Maintain Database dialog.



18. **Issue:** The Import Form Data screen raises an error "System.OutOfMemoryException" when trying to import a table with very large amount of data (greater than 400,000 records) and the grid data is loaded.
    Status: This issue is resolved.

19. **Issue:** The Import Configuration screen doesn't set the Date Last Modified value for the configuration when it's imported.
    Status: This issue is resolved.

20. **Issue:** The component TArDBTableEvents causes "Access Violation" error in Form Designer when the linked table is removed from the Main Connection or Lookup Lists Connection.
    Status: This issue is resolved.

21. **Issue:** Renaming the button controls in Form Designer may cause the linked events to be unlinked.
    Status: This issue is resolved.

22. **Issue:** Cut and paste the controls in Form Designer will cause the linked events to be unlinked.
    Status: This issue is resolved.

23. **Enhancement:** Improved performance when adding controls on a form in the Form Designer.

24. **Issue:** In Form Designer, an access violation error occurs when adding a control to the form and immediately moving the control.
    Status: This issue is resolved.

25. **New Feature:** Added the following new functions to Data Manager Designer.
    Some of these are mentioned above.

Category	Function Name	Function Type	Description
Database	DataTransferExport	Function	To export data for a list of tables to a file with the extension of ".vdt" or ".zip". Set the parameter strExportTables with a list of table names separated by "||". Table filtering is also supported in the export. Just add the symbol "~~" after the table name, then add the filter string for the table. Example: strExportTables := 'tblTable1|| tblTable2~~MyField = ''MyData''|| tblTable3~~AnotherField> 2006';
GIS	DegreesToUTMZ13	Function	Convert GIS degrees to UTM zone 13.
General	GetGeneralInfoValue	Function	Gets the info value from the general info table in the system database.
General	GetUserInfoByID	Function	Get the user information by User ID.
Date/Time	IncrementDateByYear	Function	Increment the date by a number of year.
General	IsFieldValueBlank	Function	Checks if the field value is blank or not.
General	ListFileDir	Procedure	Get a list of files in a directory.
Database	LocateNavigatorRecord	Function	Locates the navigator record by passing the navigator key value.
General	LogError	Procedure	Add an error message to the error log table.
Controls	SetBandedColumnReadOnlyStyle	Procedure	Sets the banded grid column read only style.
Controls	SetColumnsReadOnlyByTag	Procedure	Sets the grid columns read only by each column's tag number.
General	SetMasterDetailLink	Procedure	Sets the master detail link for master and detail tables.

2.7 build 1

1. **Enhancement:** When the TArImageList component is placed on the form, it will include a set of predefined common images in the image list called "cxCommonImages". You can use this image list in the components that support image list such as the buttons.
2. **Enhancement:** Changed the property editor for the ImageIndex property to show a dialog box

containing all the images for the selected image list. You can press the OK button or double click on the image to select and close the dialog box.

3. **Enhancement:** Added the "Max Field List Height" spin edit box to the Main Connection and Lookup Lists Connection dialogs for setting the maximum field list height for all the selected tables or views on the panel.

4. **Issue:** The function UpdateCoordinatesFromAddress for the component TArGoogleMap stopped working. Also, the same problem occurred for the GIS function GetCoordinatesFromAddress.
   Status: This issue is resolved.

5. **Issue:** The F5 function key for refreshing the screen is not working properly if the Data Manager application is run from the Data Manager Designer's toolbar icon.
   Status: This issue is resolved.

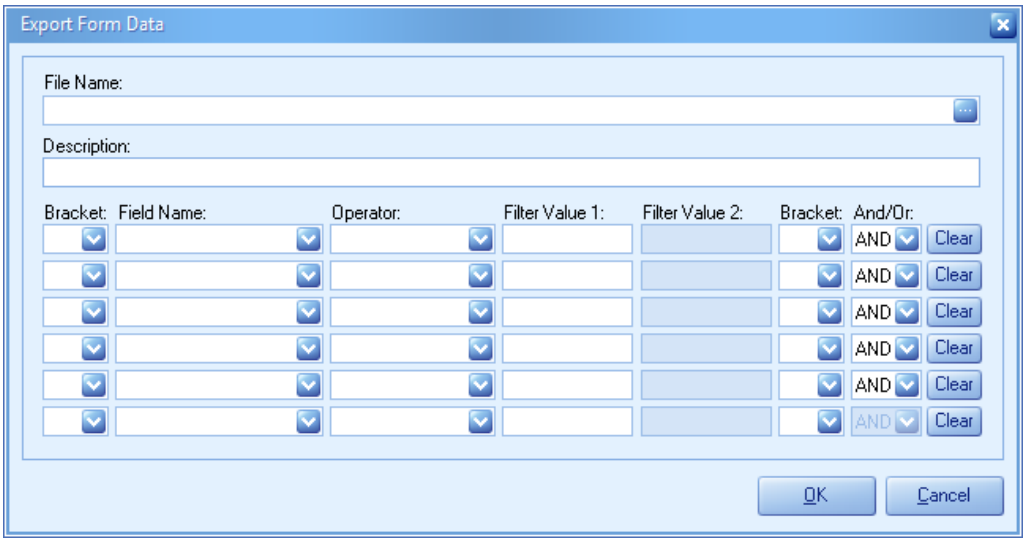
6. **Issue:** An error occurs when loading a second SQL layer for the TArGISViewer component.
   Status: This issue is resolved.

2.7 build 0

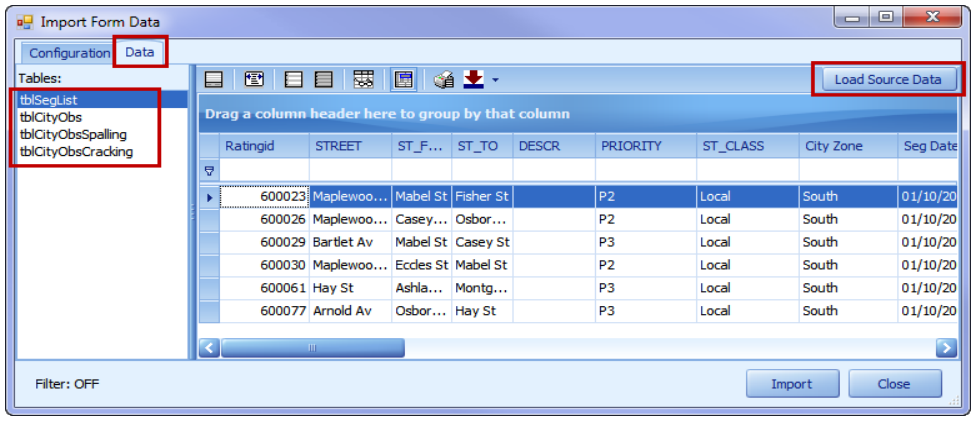
1. **Major New Feature:** Data Manager now supports GIS with a new component called: TArGISViewer. Currently this component can load map formats of GeoMedia, ESRI SDE Binary and MS SQL Spatial in a SQL Server database.

2. **Upgrade:** The Third-party controls have been upgraded to the latest version.

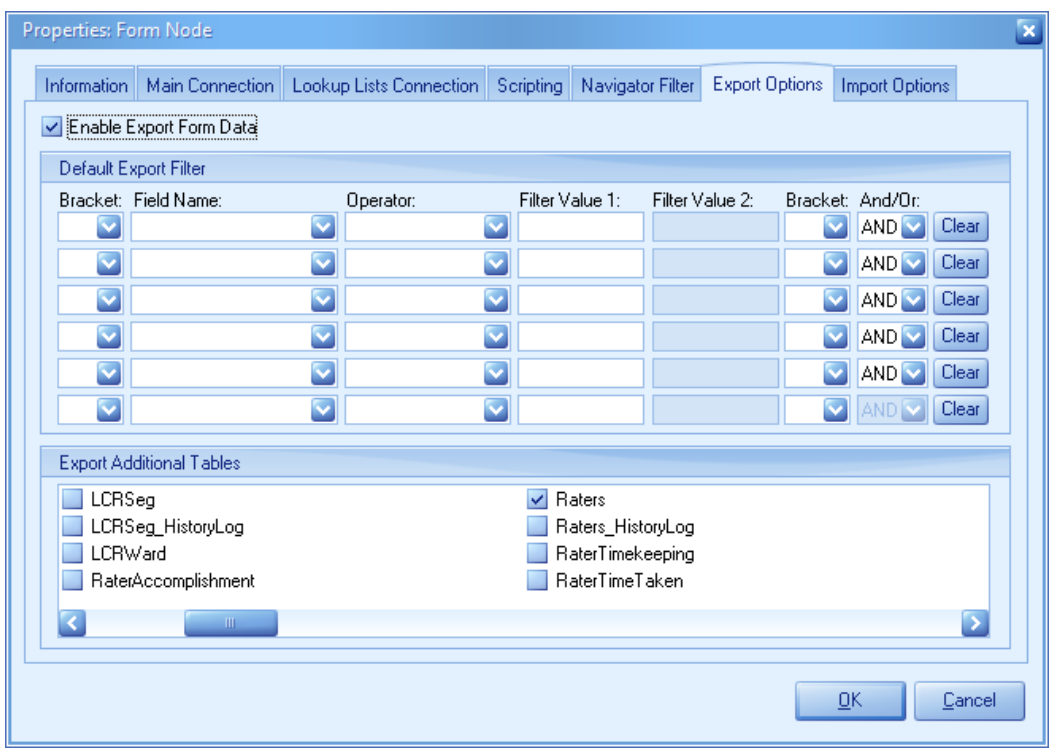
3. **Enhancement:** The Export Form Data dialog now has up to 6 fields for filtering the export data. This works much like the Advanced Filter in DataViewer.



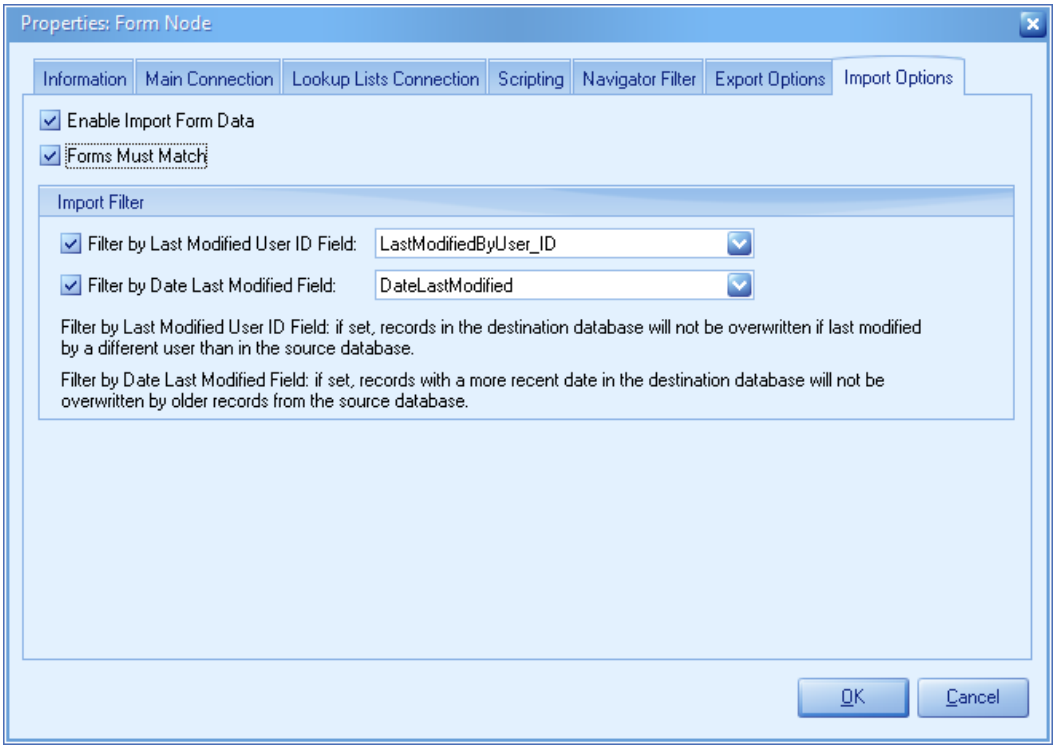
4. **Enhancement:** The Import Form Data dialog now has a navigator on the Data tab for selecting a table and displaying the corresponding data on the grid. The grid data is not loaded initially. Press the "Load Source Data" button to load the data if needed.



5. **Enhancement:** Added an "Export Options" tab to the Form Properties dialog in Data Manager Designer with the following features (as shown in the screen below):
   a. Enable Export Form Data – allows the configuration manager to control if data can be exported from certain forms or not.
   b. The configuration manager can specify a default filter to be used with the Export Form Data dialog mentioned above.
   c. The configuration manager can specify additional tables for export. This is handy, for example, if lookup lists need to also be exported rather than asking the users to go to the lookup screen and create a separate export.

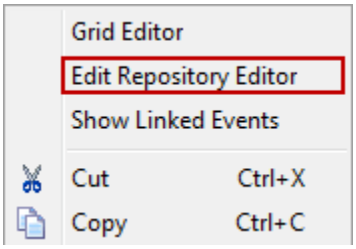


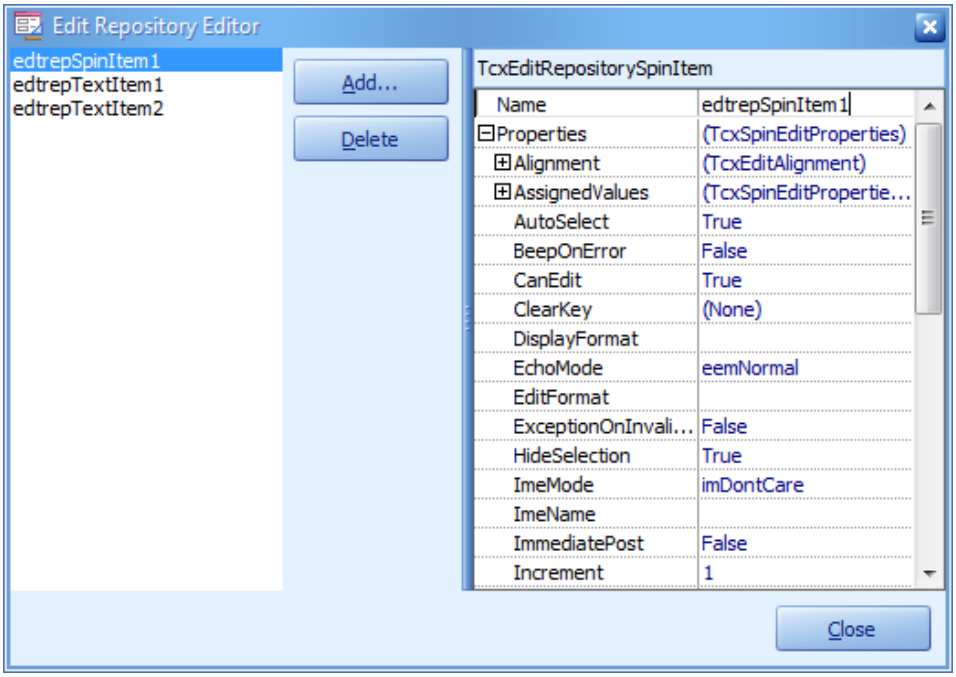
6. **Enhancement:** Added an “Import Options” tab to the Form Properties dialog in Data Manager Designer with the following features (as shown in the screen below):
    a. Enable Import Form Data – allows the configuration manager to control if data can be imported into certain forms or not.
    b. Forms Must Match – if checked only data that was exported from the same form can be imported. This prevents users from importing 2012 data into a 2013 form for example.
    c. Filter by Last Modified User ID – if set, records in the destination database will not be overwritten if the last modified by a different user than in the source database.
    d. Filter by Date Last Modified – if set, records with a more recent date in the destination database will not be overwritten by older records from the source database.



7. **New Feature:** Data Manager now supports connecting to tables with a table schema other than “dbo”. This is a bit technical but it is necessary if a database administrator sets up a database with a non-standard schema.

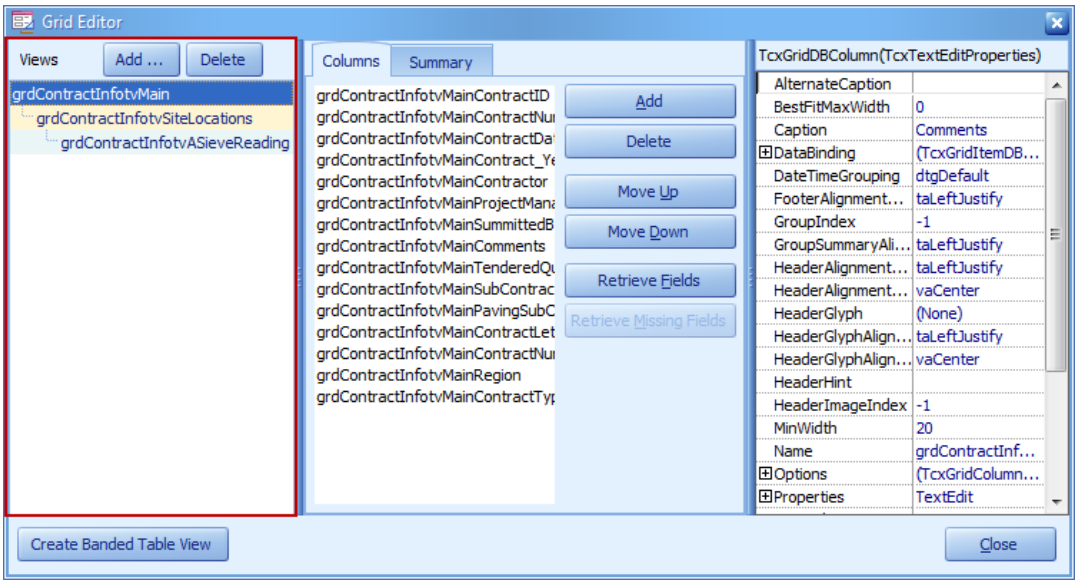
8. **New Feature:** In the Form Designer, added a new popup menu item "Edit Repository Editor" to the TArDBGrid component. The repository provides a convenient way to support a collection of repository items. It is easy to add, delete repository items and customize their properties at design time via a repository dialog. A repository item can be used in the OnGetProperties event for the grid column to specify properties common to several editors. The OnGetProperties and OnGetPropertiesForEdit events allow you to implement the MultiEditors feature, i.e. provide different editors to display and edit the cells of a single item.



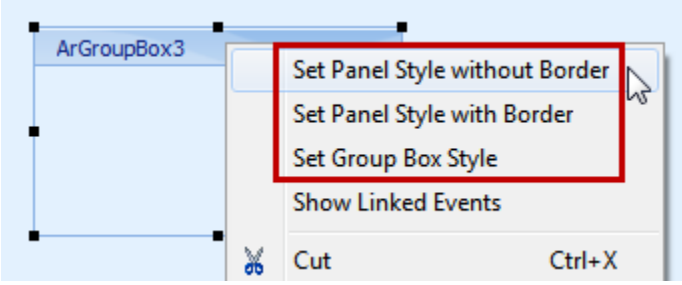


9. **New Feature:** In the Form Designer, added a new popup menu item "Show Columns in Component List" to the Form Options toolbar button. When this menu item is checked, the columns will be shown in the component list so Configuration Managers have access to the events for those columns.

10. **New Feature:** In the Form Designer, added support for multiple views to the Grid Editor dialog. The grid can display data from several datasets. Hierarchical data structures such as <u>master-detail relationships</u> are implemented via multiple views.

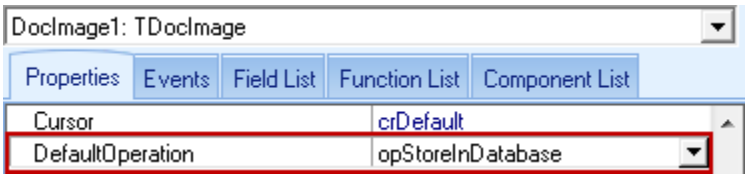


11. **New Feature:** In the Form Designer, added the following right-click menus to the Group Box control:
    - Set Panel Style without Border: is used for creating invisible toolbar panels
    - Set Panel Style with Border: is used for creating simple panels with a border but no caption.
    - Set Group Box Style: re-sets the group box back to the original settings.

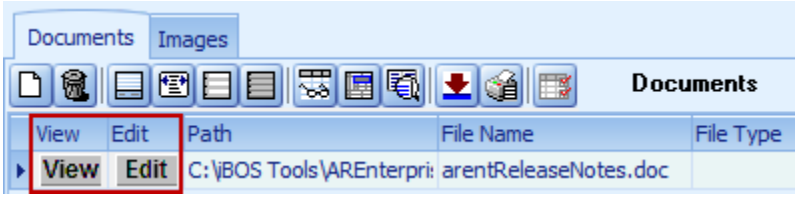


12. **Enhancement:** In the Form Designer, the size and position of the Grid Editor and the new Edit Repository Editor are saved for each user of Data Manager Designer.

13. **Enhancement:** The TDocImage component now supports storing the documents and images into the database. To enable this feature, set the DefaultOperation property value to opStoreInDatabase.



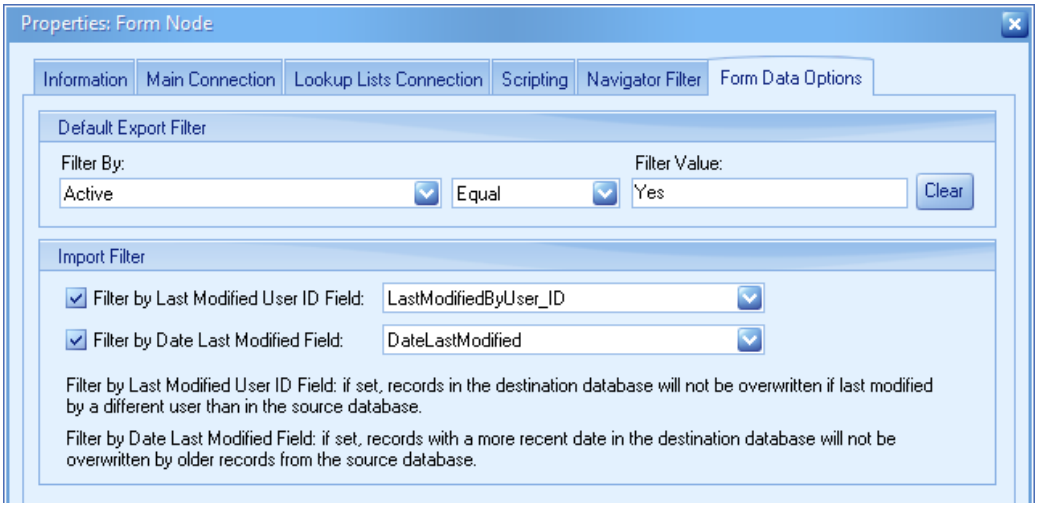
Press the View button to view the document or image and press the Edit button to edit the document or image. This component enables the record locking mechanism when the DefaultOperation property value is set to opStoreInDatabase. The current record will be locked for the current user if the Edit is pressed or other record values being edited. For existing records, changing the DefaultOperation property value to opStoreInDatabase doesn't automatically transfer the files to the database. To manually transfer a file to the database, just re-pick the file and save the current record or alternately call the function DocImage1.TransferFilesToDatabase in scripting.



14. **New Feature:** Added two new components to the Form Designer: TArFormImage, TArImageList. The TArFormImage is for displaying a graphical image on the form. The TArImageListrepresents a collection of same-sized images, each of which can be referred to by its index.

2.6 build 6

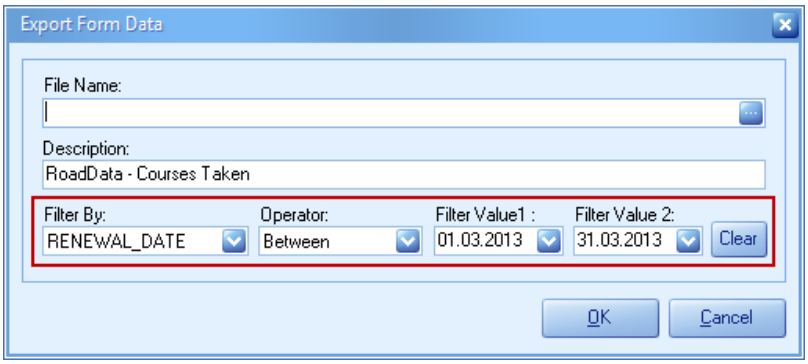
1. **Enhancement:** Added Form Data Options tab to the Node Properties dialog for setting the Default Export Filter and Import Filter.



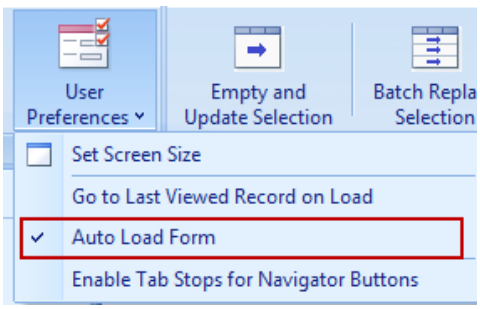
2. **Enhancement:** The "Save Group Item As" feature now supports replacing the group item if it exists.

2.6 build 5

1. **Enhancement:** Added filtering capability for the Export Form Data dialog.



2. **Enhancement:** Added "Auto Load Form" check item to the User Preferences list to automatically load the form when the group item is selected.



3. **Issue:** In the Form Designer, the shortcut Ctrl+V doesn't work on the Properties tab for the Caption and Hint properties. It causes the text being pasted into the Notes memo box.
   Status: This issue is resolved.

4. **Issue:** In the Form Designer, the typed text is being cut off on the Properties tab for the Caption and Hints properties when you type in a fast pace.
   Status: This issue is resolved.

5. **Issue:** In Code Editor, pasting text from the clipboard without clicking on the memo box first causes the text being pasted in the wrong position.
   Status: This issue is resolved.

2.6 build 4

1. **Upgrade:** The Third-party controls have been upgraded to the latest version.

2. **Enhancement:** The Main Connection dialog now supports adding views. The view can only be used a master table or lookup table.

3. **Enhancement:** Added new field types "Date", "Time", "Geometry", and "Geography" to the Maintain Database dialog for SQL Server version 2008 or greater.

4. **Issue:** In the Form Designer, the screen flickers when selecting the control inside the TArGroupBox.
   Status: This issue is resolved.

5. **Issue:** In Maintain Database, the values for the Description column are lost when the Up/Down button is pressed and then press the Apply button.
   Status: This issue is resolved.

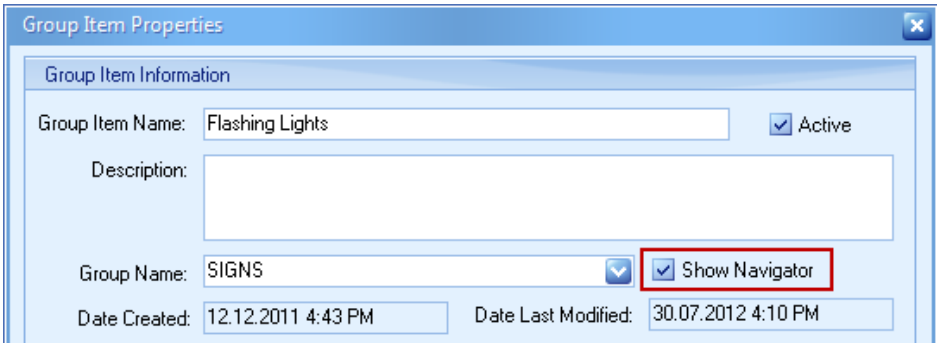
2.6 build 3

1. **Enhancement:** The primary key generation for various types of key fields is improved significantly for multiple users doing data entry at the same time on the same form.

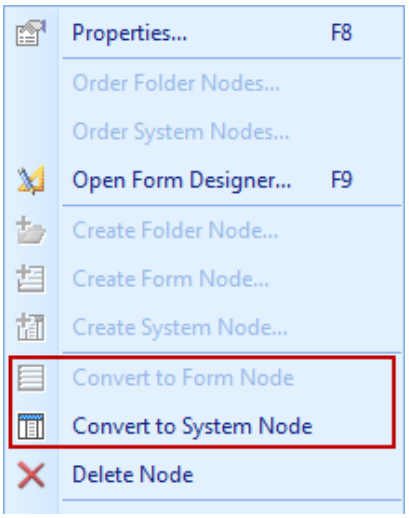
2.6 build 2

1. **Upgrade:** The Third-party controls have been upgraded to the latest version.

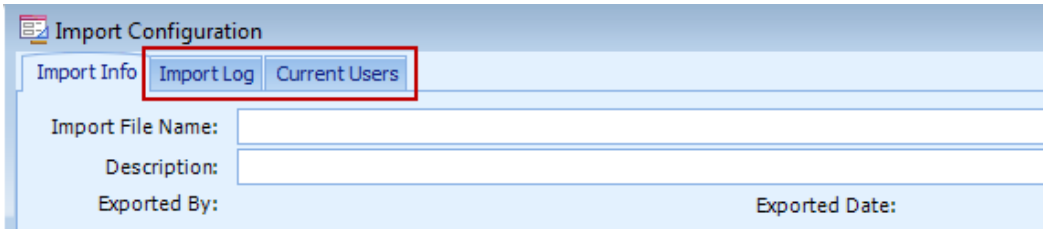
2. **New Feature:** Added the "Show Navigator" checkbox to the Group Item Properties dialog. If it's unchecked, the navigator will be hidden for the selected group item in view mode. The right-click popup menu for the Navigator will then appear on the Group Item node. So things like Grid and Splitter settings can still be reset.



3. **New Feature:** Added two menu items to the navigator popup menu called "Change to Form Node" and "Convert to System Node". The "Convert to Form Node" menu item is for converting the Node Type for the selected System Node to Form Node. The "Convert to System Node" menu item is for converting the Node Type for the selected Form Node to System Node.



4. **New Feature:** Added the "Import Log" and "Current Users" tabs to the Import Configuration dialog. The "Import Log" is for showing the information about the previous imports. The "Current Users" is for showing a list of users currently logged into the Data Manager application.



5. **New Feature:** Added a new control called "TArDBRichEdit" which is an edit control that allows rich text to be edited. In Data Manager, just double click on the Rich Edit control to show the Rich Edit Editor.

6. **New Feature:** Data Manager now supports automatic spell checking for any text edit controls in the application.

7. **Enhancement:** The Notes memo box In the Form Designer now supports rich text.

8. **Enhancement:** Improved performance on initial loading of the form.

9. **Enhancement:** The Grid Editor now supports setting the common property for multiple columns at once for the selected columns.

10. **Enhancement:** The Grid Editor now automatically sets the column style to read only when the column's ReadOnly property is set to True.

11. **Issue:** In the Code Editor dialog, the error "Unknown Identifier or variable is not declared: 'tbl…'" occurs for a specific table even the table actually exists.
    Status: This issue is resolved.

12. **Issue:** The error "Cannot locate record in table" occurs when the Navigator Key Field defined in the Main Connection dialog is not the primary key.
    Status: This issue is resolved.

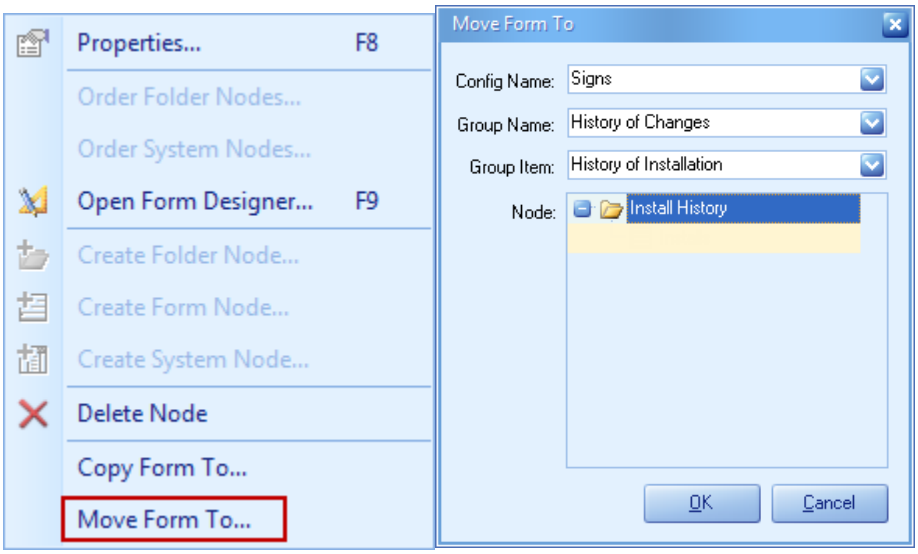
13. **Issue:** In the Form Node Properties dialog, cannot open the Main Connection dialog when a table or a linked field is missing.
    Status: This issue is resolved.

2.6 build 1

1. **Upgrade:** The Third-party controls have been upgraded to the latest version.

2. **Issue:** The tab stop for the TArDBRadioGroup component doesn't work.
   Status: This issue is resolved.

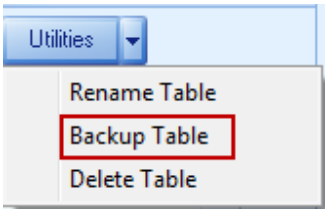
3. **New Feature:** Added the “Move Form To” menu item to the navigator popup menu in Data Manager Designer. Use the “Move Form To” dialog to move the selected form to any folder that not yet has a form created.



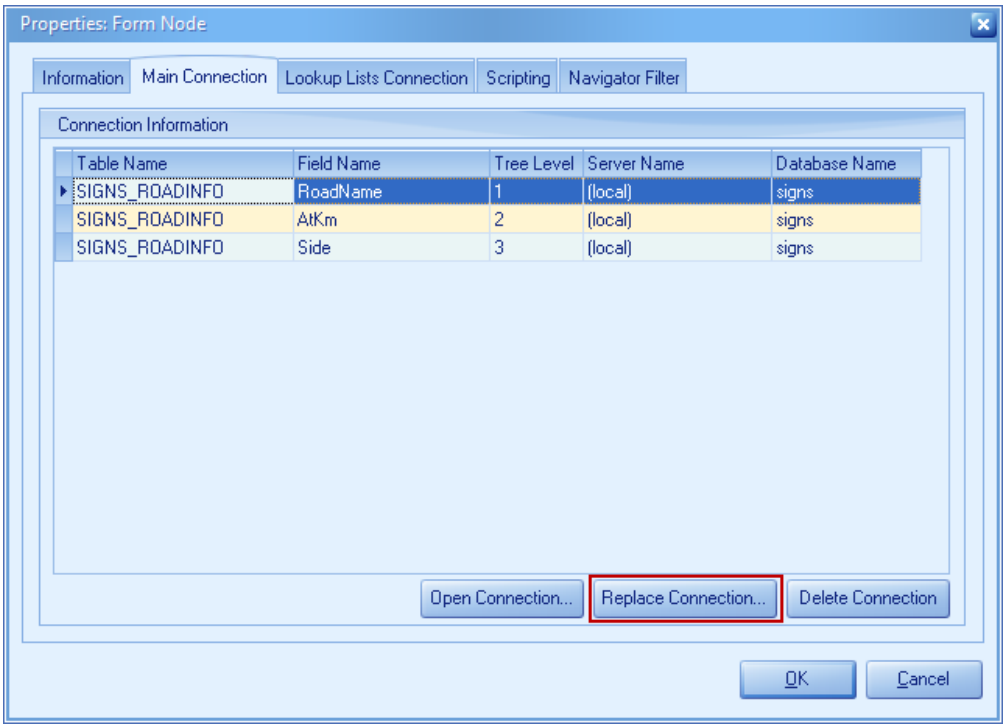
4. **New Feature:** Added the “Cascade” and “Tiles” icons to the toolbar in the Lookup Lists Connection dialog. Use these icons to arrange the tables that are added to the lookup lists.

5. **New Feature:** Added new controls called “TArTrackBar” and “TArDBTrackBar” which represent the track bar control that is used to edit values using a slider.

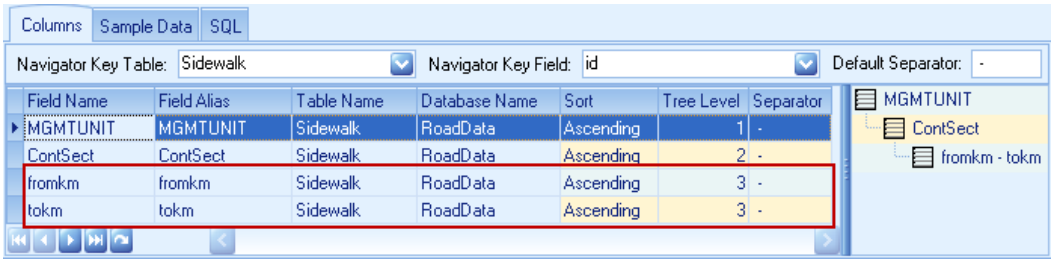
6. **New Feature:** Added new menu item “Backup Table” to the Maintain Database dialog for backing up a table including all the data.

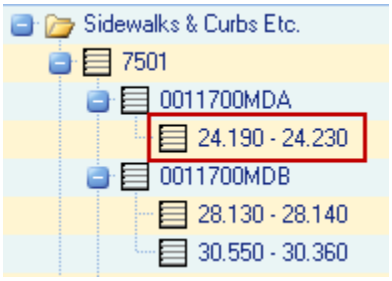


7. **New Feature:** Added the "Replace Connection" button to the Main Connection and Lookup Lists Connection tabs on the Form Node Properties dialog. The purpose of the "Replace Connection" button is to replace the database connection with a compatible database that has the same table structures.

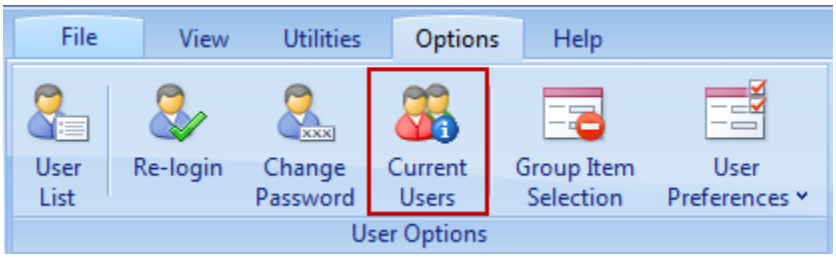


8. **New Feature:** The "Columns" tab in the Main Connection dialog now supports combining two or more fields for each level to be display in the navigator. Each field can have a different separator. Each row (field) in the grid can be dragged and dropped into the desired position. The Tree Level values for all rows will be automatically re-ordered if the Tree Level values for the dragged row and the dropped row are different.





9. **New Feature:** Added the "Current Users" icon to the Options tab on the ribbon to show a list of users currently logged into the Data Manager application.



2.6 build 0

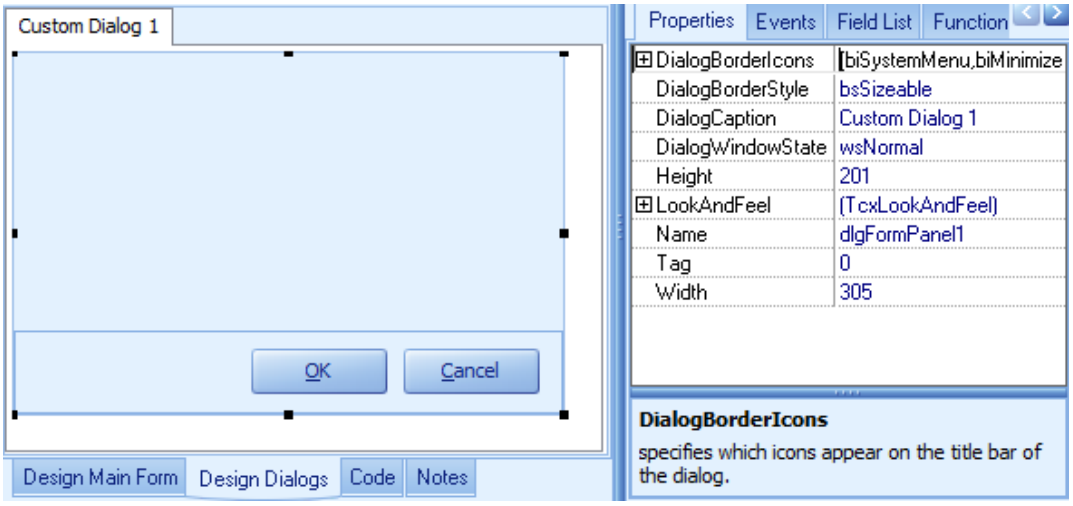
1. **Major New Feature:** Data Manager now supports custom dialogs. In Data Manager Form Designer, press the Design Dialogs tab to show the design panel for the dialogs.

   The following scripts are generated automatically when you double click on the OK and Cancel buttons:

```
Procedurebtn DlgOK1Click(Sender: TObject);
begin
    frmCustomDialog.ModalResult := mrOK;
end;

procedurebtnDlgCancel1Click(Sender: TObject);
begin
    frmCustomDialog.ModalResult := mrCancel;
end;
```

   To show the dialog, call the function ShowCustomDialog(strDialogCaption: String):

```
procedure btnShowDialogClick(Sender: TObject);
begin
    if (ShowCustomDialog('Custom Dialog 1') = mrOK) then
    begin
        //do some action when the OK button on the dialog is pressed

    end;
end;
```



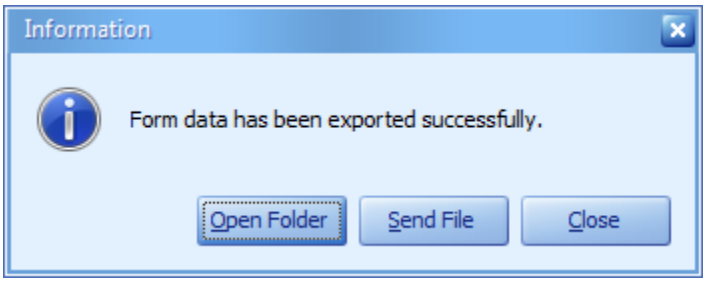
2. **New Feature:** Added the TArTimer component to the Advanced tab on the Component Palette. Use this component to trigger an event, either one time or repeatedly, after a measured interval. Write the code that you want to occur at the specified time inside the timer component's OnTimer event.

3. **Major New Feature:** Added the TArGPS component to the Advanced tab on the Component Palette. Use this component for reading the GPS information from a NMEA-compatible GPS Receiver. Contact VEMAX for details on how to use this component.
VEMAX has tested this component on the following GPS: **Holux GR-213 USB GPS** that we purchased from Canada GPS:
http://www.canadagps.com/mousegpsgr213usb.html

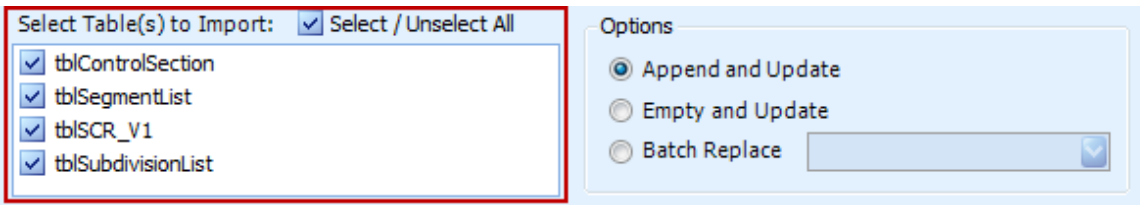
   Here's a link to the technical spec of the Holux GR-213 USB GPS:
http://www.canadagps.com/Downloads/GR-213U-spec-E.pdf

   Here is an image of the GPS we used:

4. **New Feature:** Added the Open Folder and Send File buttons to the message dialog for the Export Configuration and Export Form Data screens. The Open Folder button is for opening the folder containing the exported file. The Open Folder button is particularly handy if you need to copy the exported file. The Send File button is for sending the exported file by email.



5. **New Feature:** Now you can select or unselected the table(s) to import in the Import Form Data screen. It remembers the unselected tables so you don't have to unselect them again the next time you do import form data.

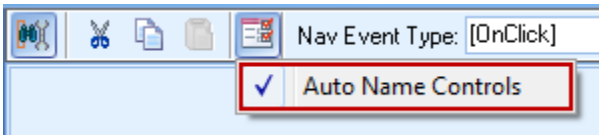


6. **Enhancement:** The Import Form Data screen automatically creates the tables and/or fields if they don't exist in the destination database.

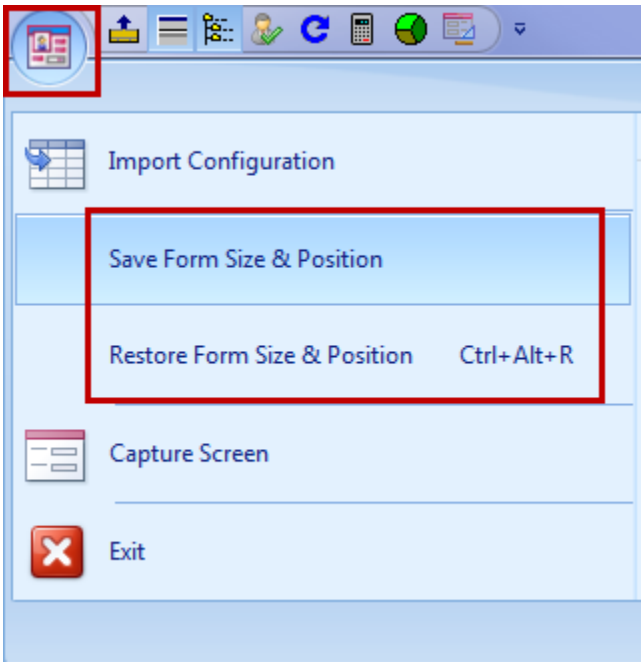
7. **New Feature:** The Maintain Database screen now supports adding a description for each field. The Description field is not required but it is useful if you want to document specifically what the fields are used for in a table.

No.	Field Name	Field Type	Field Size	Decimals	Key Field	Index Field	Description
1	ID	Large Int	0	0	☑	☐	
2	SegNo	Alpha	20	0	☐	☐	Segment Number
3	MGMTUNIT	Alpha	5	0	☐	☐	Management Unit

8. **New Feature:** Added the Auto Name Controls check box to the Form Designer Toolbar for enabling or disabling auto renaming the controls when the property is changed for Caption, DataField, DataSource, etc. This is so that if you have already setup the names of the controls you want to keep and have added code to the events of some controls, you can turn off the auto naming feature so the name of the controls doesn't get out of sync with the events. Data Manager remembers this setting on each form for each user of Data Manager Designer so whatever you had set last time, it will be set that way the next time you go into Form Designer for that form.



9. **New Feature:** Added the ability to save and restore the size and position of the application. This is to overcome problems some users who have two monitors experience in a Citrix environment. To use this feature, simply move Data Manager to where you want it to always load then click on the "Save Form Size & Position" item under the Application menu item as shown below. If you want to clear these settings and go back to the default behaviour, click on the "Restore Form Size & Position" menu item or press Ctrl + Alt + R on the keyboard.



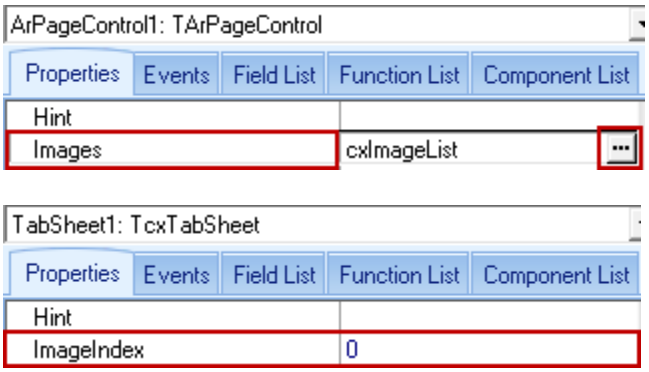
2.5 build 1

1. **New Feature:** Data Manager now supports being called from DataViewer Enterprise's Edit button and directly goes to the form and record defined from DataViewer Designer.

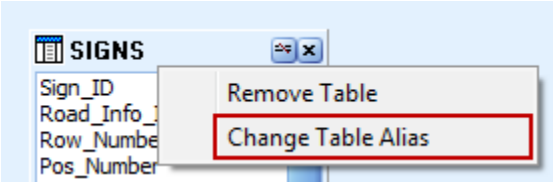
2. **New Feature:** Added the function *ShowConfirmMessage(strMessage: String): Word* for showing the confirmation message dialog which requires the user to type the word "Yes" to confirm. The return value is either mrYes or mrNo depending on which button the user presses.

3. **New Feature:** Added OnAfterNew and OnAfterSave events to the New button, OnAfterSave event to the Save button, OnAfterCancel event to the Cancel button, and OnAfterDelete event to the Delete button.

4. **New Feature:** Added the Images property to the TArPageControl component for supporting an image on each tab. To add images to the image list, select the TArPageControl component and click on the ellipse button on the Properties editor. Then select the TcxTabSheet component and set the ImageIndex to the desire image in the image list. The ImageIndex for the first image is 0.

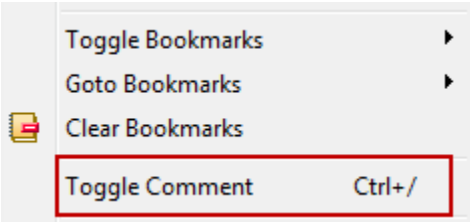


5. **New Feature:** The Main Connection and Lookup Lists Connection dialogs now support changing the table alias. Right click on the list box header to show the popup menu.



6. **New Feature:** Added the Font Size combo box to the Code Editor Toolbar for setting the font size for the Code Editor. This setting is automatically saved for each user of Data Manager Designer.

7. **New Feature:** The Code Editor now supports toggling comment for a block of selected lines by pressing Ctrl + / combined keys. This feature can also be accessed via the popup menu.



8. **New Feature:** The Field List and Component List grids in the Form Designer now support selecting multiple cells for doing Copy (Ctrl + C) and Paste (Ctrl + V).

9. **New Feature:** Data Manager now supports saving and restoring the splitter Positions on the form. Use the "Reset Splitter Positions" popup menu item on the Navigator to reset the splitter positions back to default.

10. **New Feature:** Data Manager Designer now supports setting the System Nodes order. Use the "Order System Nodes" popup menu item on the Navigator to set the order for the System Nodes.

11. **New Feature:** Added the "Node Expanded" check box to the Node Properties dialog for setting the node as expanded.

12. **New Feature:** Added the "Grid Options" button to the TGridToolbar component. To show this button, set the "ShowGridOptionsButton" property to True. When the "Grid Options" button is pressed in Data Manager, it will show the popup menu containing the "Show Auto Filter Row" menu item. The "Show Auto Filter Row" menu item is for showing or hiding the Auto Filter Row on the grid.

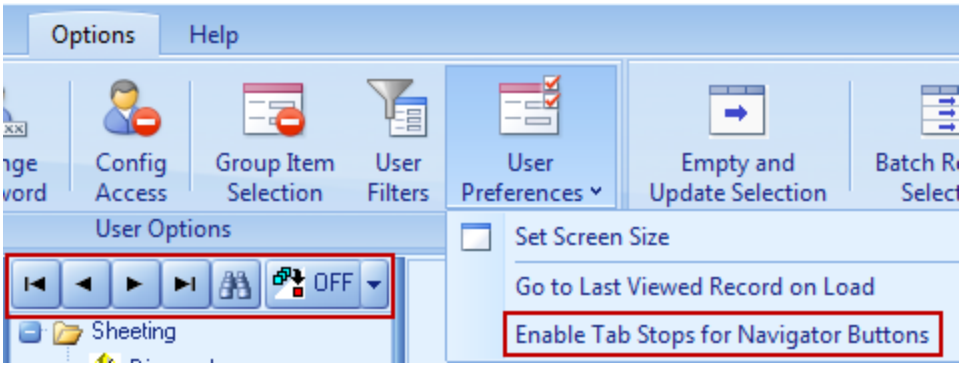
13. **New Feature:** Added the following events to the TArDBGrid component:

    OnCustomDrawCell
    OnEditValueChanged
    OnEditing
    OnFocusedItemChanged
    OnFocusedRecordChanged

14. **New Feature:** The cxStyleReadOnly component now supports changing the color dynamically with the current theme if the Color property is set as clDefault.

15. **New Feature:** Added the "Cleanup Backup" button to the Backup Configuration dialog for cleaning up the existing backups. The number of backups is growing as more forms are being built. That makes it harder to maintain the backup list and it also requires more disk storage space. That's why the system administrator should cleanup the backups regularly.

16. **Enhancement:** The tab stops for navigator buttons are now disabled by default. This makes it quicker to tab from the navigator to the form. They can be set as enabled by checking the "Enable Tab Stops for Navigator Buttons" menu item.



17. **Issue:** The TArDBGrid component doesn't save the grid footer summaries.
    Status: This issue is resolved.

18. **Issue:** The Navigator Filters dialog doesn't work properly when specifying Starting Range and Ending Range for the date field.
    Status: This issue is resolved.

19. **Issue:** The Maintain Database utility doesn't process the deleted fields when modifying a table and the Apply button is pressed.
    Status: This issue is resolved.

2.5 build 0

1. **Major Release:** The internal workings of Data Manager 2 have been enhanced significantly to make PPT 2 possible. PPT is our Performance Prediction Technology and it is built, for the most part, using Data Manager.

2. **New Feature:** Added the TArColorComboBox and TArDBColorComboBox (data-aware) controls to the Tool Palette on the Form Designer. Use the TArColorComboBox or TArDBColorComboBox control to add a color combo box to your form. With it, you can display a list of colors (with text descriptions), which can be selected via a dropdown window. There are four predefined sets of colors provided and you can also create color sets manually. Additionally, end-users can select custom colors using a standard or a custom color editing dialog. The control also provides a 'most recently used' feature so that end-users can easily reuse recently selected colors.

3. **New Feature:** Added the ability to automatically name controls in Data Manager based on the following:

   For data-aware controls, Data Manager sets the name to our standard prefix + the DataField when the DataField is set.
   For non-data-aware controls, Data Manager sets the name to the prefix + the Caption without spaces when the Caption is set.

   i.e.
   ArLabel: lbl + the Caption without spaces
   ArDBEdit: dbe + the DataField selected

4. **New Feature:** Added the function *GetCustomColor: TColor*that returns the color from a popup color dialog. It returns the color value clNone if the user cancels from the dialog.

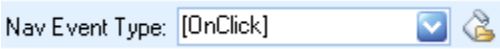
5. **New Feature:** Added the procedure *RefreshForm*for refreshing the current form without rebuilding the navigator. This may be needed to force the form to refresh after visual changes are made to the controls on the form.

6. **New Feature:** Added the procedure *SetChildControlsEnabledByTag(ctlParent: TWinControl; intTag: Integer; bolEnabled: Boolean)*for setting the Enabled property of child controls of a certain tag number.

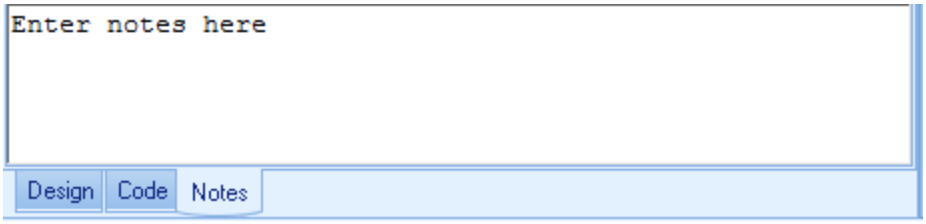
7. **New Feature:** Added the procedure *SetChildControlsReadOnlyByTag(ctlParent: TWinControl; intTag: Integer; bolReadOnly: Boolean)*for setting the ReadOnly property of child controls of a certain tag number.

8. **New Feature:** Added new property called: "Version" to the TArDBGridcontrol for detecting the compatibility of the saved grid layout with the latest version of the grid. If the saved grid layout has a different version number from the one set in the Form Designer, the saved grid layout will not be loaded. This is so that the developer of the form can set the grid version to a new number when he or she makes a change to the grid columns that the users need to be aware of. E.g. If the current version of the grid is: 1 and the developer changes columns that need to be updated for the users, the developer will set the version to 2. The next time the users open the form, Data Manager will detect the changed version and ignore the automatically saved layout for that grid and load the grid the way the developer set it in Data Manager Designer. This eliminates the need for the developer to tell all the users to reset their grid settings to see the new changes in the grid.

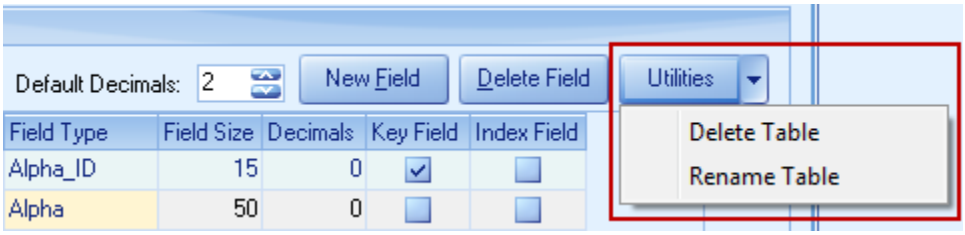
9. **New Feature:** Added the Navigator Event Type combo box and the Open Script button to the tool bar In the Form Designer for opening the Navigator Scripting dialog from Form Designer. This means that if you are in the Form Designer and you need to make a change to the OnClick event for the navigator, for example, you can simply select that event in the pick-list and click the Open Script button beside it without leaving the Form Designer.



10. **New Feature:** Added the Notes tab to the Form Designer for adding notes for the current form. This is intended to for the developer of a form to add any notes about that form that they see fit like an explanation, To-Do list, etc. The amount of text that can be entered per form is unlimited.



11. **New Feature:** Added the Utilities button to the Maintain Database screen for deleting and renaming a table. This button is only enabled when a table is in Modify mode.



12. **Enhancement:** The properties to show the Filter Row and Navigator on the grid control are now turned on by default when a grid is added to form.

13. **Enhancement:** Changed Data Manager so that if a user clicks on a Group Item and there is only a single System node or one Folder node with one System node then Data Manager should go directly to that System node without requiring the user to click on it.

14. **Enhancement:** Filters the Config Pick List to only show the configurations that the current user has access to.

15. **Enhancement:** Added several Date and String functions to the Functions List in Data Manager Designer.

16. **Enhancement:** Improved the Convert Script button to look for many additional code fragments to convert. This button is now driven by a table so additional code fragments can easily be added in the future.

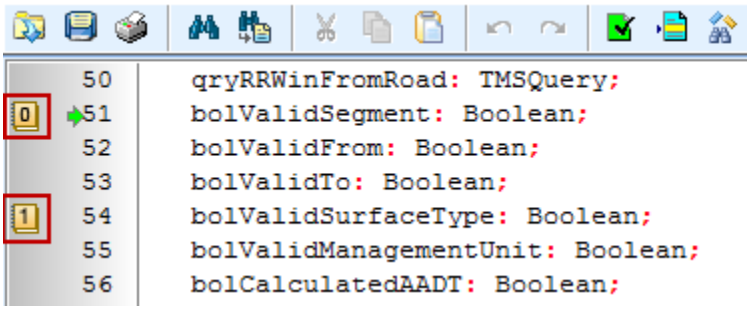
17. **Modification:** Changed the default for the ParentFont property to False for all controls.

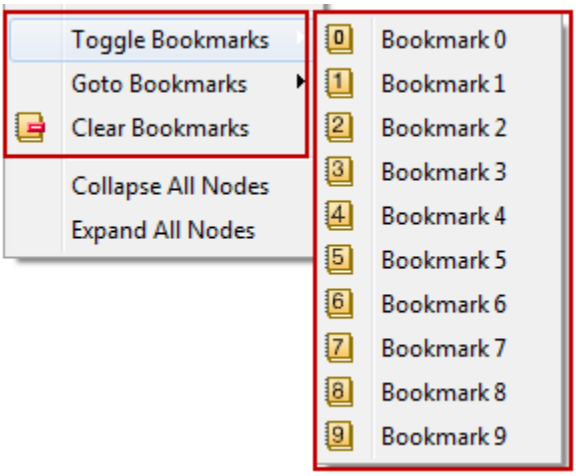
18. **Issue:** The Import/Export Form Data feature doesn't work for System Nodes.
    Status: This issue is resolved.

19. **Issue:** The Search and Replace dialog doesn't work properly in Code Editor.
    Status: This issue is resolved.

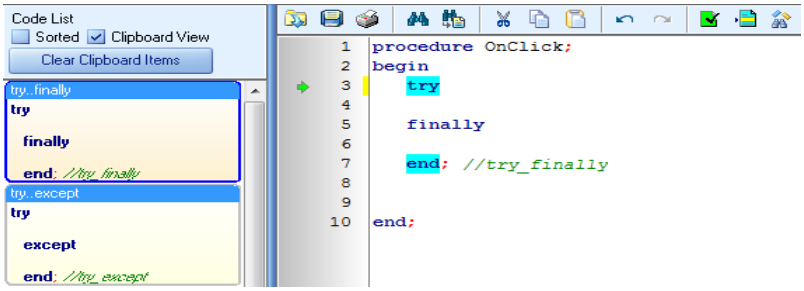
2.1 build 0

1. **New Feature:** The Code Editor now supports adding up to 10 bookmarks. Press Ctrl + Shift + Number (0 to 9) to toggle the bookmark on and off for the current line. Press Ctrl + Number (0 to 9) to go to the bookmark. This feature can also be accessed via the popup menu.

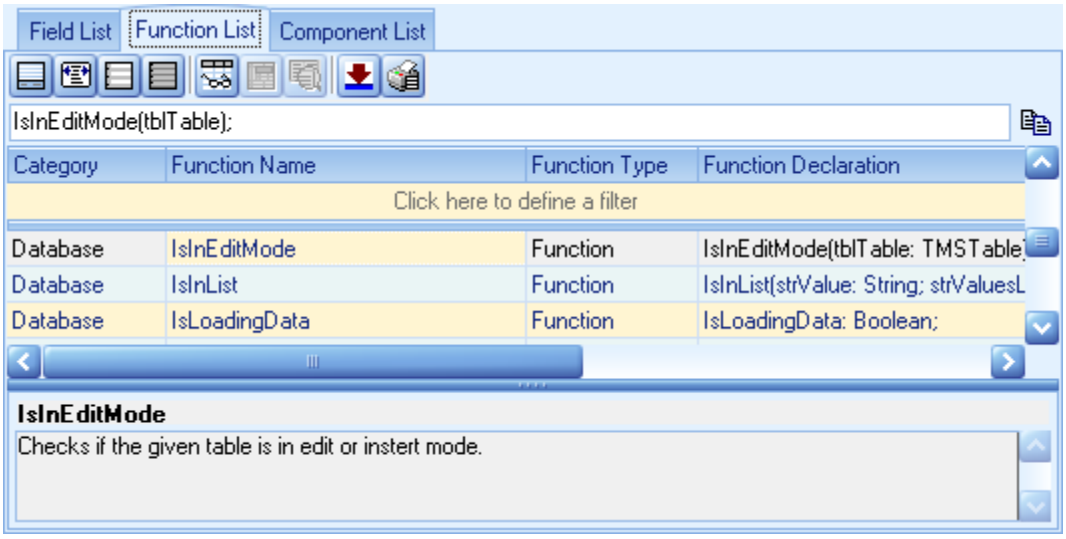




2. **New Feature:** Added the Code List panel to the left side of the Code Editor. The Code List is a list of code snippets that are shown with syntax highlighting in a listbox control. It can be used to drag the code from the Code List and drop it to the Code Editor. It can also be used as a clipboard monitor, displaying all code snippets that are being cut or copied to the clipboard from the Code Editor (if the "Clipboard View" box is checked).



3. **Enhancement:** The Function List grid In the Form Designer is now searchable and contains more columns: Category, Function Name, Function Type, Function Declaration, Version Added, and Date Added. It also shows the description for the selected function at the description panel below the grid.



4. **Enhancement:** Added the "Re-login" icon to the Quick Access toolbar which allows the current user to re-login as a different user.

5. **Enhancement:** The System ID feature now supports the primary key with the data type of "Integer", "Large Int", or "Float". It appends the last three digits of the System ID to the right side of the primary key value when a new record is created. The System ID must be a compatible number for this to work. When creating a new table from the Maintain Database screen, it is recommended to use the data type of "**Large Int**" for the primary key for all new tables that do not need to be backward compatible with the old Alpha IDs.

6. **Issue:** When the option "Go to Last Viewed Record on Load" is checked and the navigator record cannot be found, it causes the application to stop responding.
   Status: This issue is resolved.

7. **Issue:** The Import Configuration dialog doesn't keep existing group item selection records after the group items being re-imported.
   Status: This issue is resolved.

8. **Enhancement:** Copying and pasting script source code into an email (or Word) will now show the syntax highlighting (provided the email is not set to plain text).
   This makes the script much easier to read for the recipient.

2.0 build 7

1. **New Feature:** Added the function *IsInEditMode(tblTable: TMSTable): Boolean;* This function is very useful when you want to know if a table is in edit mode.
   e.g. if **IsInEditMode(tblSigns)** then
   tblSigns.Post;

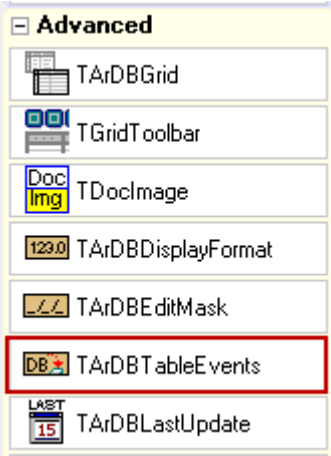
2. **Enhancement:** The Import Configuration dialog now supports importing the original configuration to a different configuration and creating a full backup before import to the existing configuration.

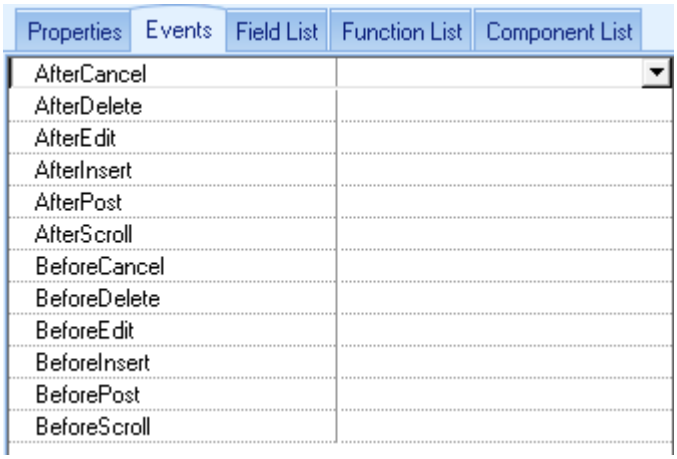
3. **Issue:** The icons for the group items were not loaded correctly after the user imported the group items.
   Status: This issue is resolved.

4. **Enhancement:** Improved performance when loading a form with thousands of records in the Navigator.

2.0 build 6

1. **New Feature:** Added the TArDBTableEvents component to the Tool Palette on the Form Designer. This component is useful for writing custom code on the table events. For example if you need to set a value in a field before the table is posted (saved), use the BeforePost event.





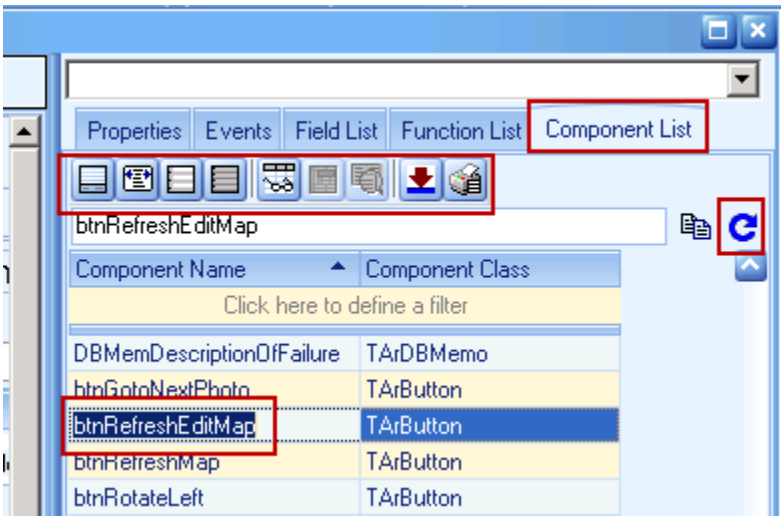
2. **Enhancement:** The Group Item Properties and Node Properties dialogs now support loading PNG image file for the Group Item icon and the Navigator Node icon.

3. **New Feature:** Added the function *IsInList(strValue: String; strValuesList: String; bolCaseSensitive: Boolean): Boolean;* This function is very useful when you want to know if a value is one of many items.
   e.g. if **IsInList(strScore, 'S1,S2,S3', False)** then
   ShowMessage('Score is Slight');

4. **Enhancement:** Improved the importing of configurations. Now if a configuration being imported does not exist, the import will create it automatically.

2.0 build 5

1. **New Feature:** Added the Component List grid to the Form Designer and Code Editor screens as shown below to show a list of all components on the form. This is so that it is easier to copy and paste a component name into code editor. This is especially handy in the Code Editor in the Properties screen. You can also export or print this list.



2. **New Feature:** Added the grid tool bar to the Field List Function List and Component List tabs in the Form Designer and Code Editor screens.

3. **New Feature:** Added a refresh button  to the Field List. This is so that if the structure of the tables is changed by another user or other software while you are in the designer, you do not have to exit the form to refresh this list.

4. **New Feature:** The code editor now supports parameter hints. Parameter hints is activated when you type a method/function name followed by "(". This will display the list of parameters for the specified method, so you can properly type the parameters.

5. **New Feature:** Added the Backup button to the Form Designer and Code Editor screens for creating manual backups for the current form (last saved). If you want the latest changes to be backed up, be sure to click the Apply button before clicking the Backup button. This is handy if you want to backup the current form before making major changes. Since these are manual backups, they will not be overwritten by automatic backups.

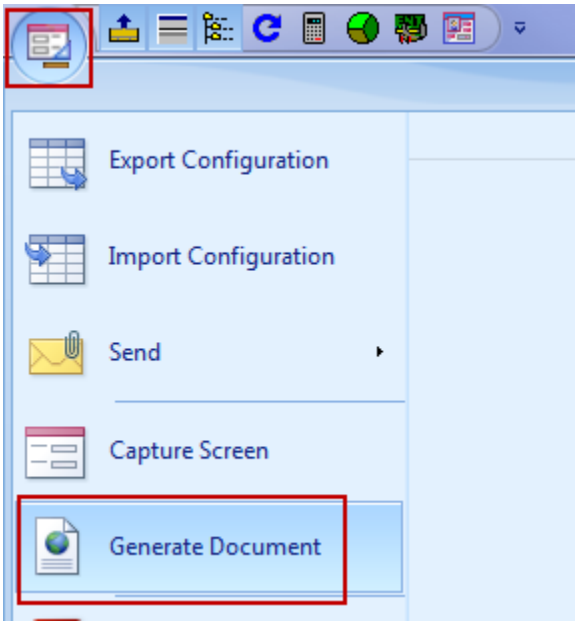
6. **New Feature:** Added a tool bar icon  to the code editor for converting the code to the new syntax.

7. **New Feature:** Added the procedure "*OpenFileAR(strFileName: string; strParams: String)*" for opening a specified file. The file can be an executable file or a document file. The parameter "*strParams*" is for passing parameters to an executable file if needed, otherwise set it to blank with two single quotes.

8. **Enhancement:** Significantly improved the ability to copy and paste controls from Data Manager 1.x to Data Manager 2.x. This includes the Page (tab) control.

9. **New Feature:** Added Copy and Paste tool bar buttons to the Main Connection and Lookup Lists Connection screens. The Copy button is for copying the query structure to the clipboard and the Paste button is for pasting the query structure from the clipboard. This is useful when you want to copy the query structure from one form and paste it to another form. It even supports the query structure from Data Manager version 1. The Copy button has also been added to version 1.1 build 28 so that you can copy the query structure from the old Data Manager to the new one.

10. **Enhancement:** Added the ability to save the Generated Document where ever you want. When you click on the Generate Document menu item (as shown below) a Save dialog will be displayed. The default folder will be your My Documents folder however you can browse where you want to save the file.

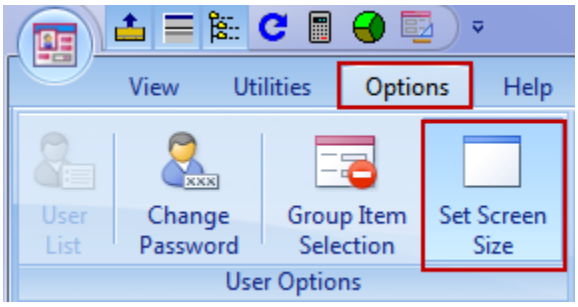


11. **Enhancement:** Added a Comments memo box to the Add Doc/Image dialog.

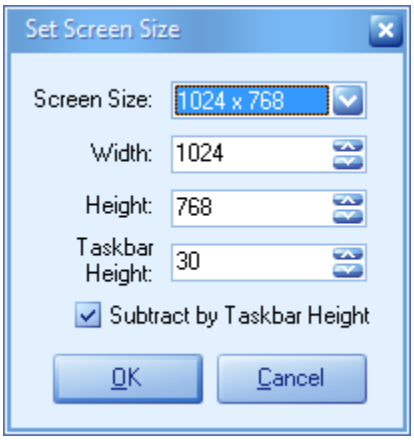
12. **Enhancement:** The Comments memo column in the Doc/Image grid now shows the text and a drop down to seem all comments.

13. **New Feature:** Added the following new functions:
    - GetMyDocumentsFolder: Used to get the user's My Documents folder. This is useful if you want the File Open Dialog to default to the user's My Documents folder.
    - GetMyPicturesFolder: Used to get the user's My Pictures folder. This is useful if you want the Open Image Dialog to default to the user's My Pictures folder.
    - GetMyAppDataFolder: Used to get the user's App Data folder. This is useful if you want save or retrieve Application specific information to or from the user's App Data folder.

14. **New Feature:** In Data Manager, added a button in the Options Ribbon to set the screen to a selected screen resolution. This is very handy when designing and testing forms when the end user's screen resolution is different from yours and you want to make sure all the controls on



This example is what you want to use when the end user's screen is 1024 x 768 and they have their taskbar visible at all times. This will set the size of Data Manager to 1024 x 738 *(768 – 30).* You can select from a pre-determined list or set the Width and Height separately.



2.0 build 4

1. **New Feature:** Added the procedure "*ShowMessageMemoEx(strDescription: string; MsgLines: TStrings)"* for passing the description along with the message lines to the message dialog.

2. **New Feature:** Added the procedure "*SaveUserDefinedSetting(strSettingKey, strSettingValue: String)"* for saving a user defined setting. This is very handy when you want to save a user-specific setting like last folder the user browsed to when opening a photo (or anything you want).

3. **New Feature:** Added the function "*GetUserDefinedSetting(strSettingKey: String): String"* for getting a specified user defined setting.

4. **New Feature:** Added the function "*ConvertLatLongToUTM(constextLat, extLong: Extended; var extNorthing, extEasting: Extended; var strZone: string): Boolean"* for converting Latitude and Longitude to Universal Transverse Mercator (UTM)and returning Northing, Easting, and Zone values.

5. **New Feature:** Added a tool bar icon  to the code editor for showing the "Go to Line Number" dialog. This dialog allows the user to specify a line number to go to in the code editor window.

6. **New Feature:** Added a new control called "TArGoogleMap" for displaying a Google map directly

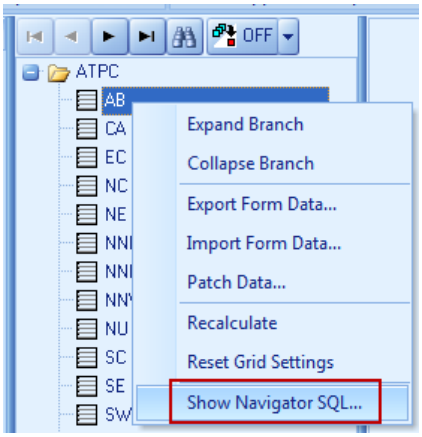
on the form. In the script, you can set either the LocAddress property value or LocLatitude and LocLongitude property values and call the procedure RefreshMap to load the map and have it go to that location.

For example:

```
procedure ArButton1Click(Sender:TObject);
begin
   ArGoogleMap1.LocLatitude := '53.5313';
   ArGoogleMap1.LocLongitude :='-113.41844';
   ArGoogleMap1.RefreshMap;
end;
```

7. **New Feature:** Added a new control called "TArProgressBar" for displaying a progress bar on the form.

8. **New Feature:** Added the "Show Navigator SQL" menu item of the Navigator as shown below. This enables developers to see the query and copy it to other applications like DataViewer Designer.



2.0 build 3

1. **New Feature:** The Navigator Filters feature from the Data Manager version 1 has been fully implemented in this new version.

2. **Enhancement:** The TArDBGrid control now supports copy and paste.

3. **Enhancement:** The TArPanel control now supports the theme color by setting the UseThemeColor property to True.

4. **Enhancement:** Copy the controls from the Data Manager version 1 and paste them into Data Manager version 2 is now working better.

5. **New Feature:** The Form Designer screen now locks the current form for editing for the user who has it opened first and no one else can make changes to the current form until it is unlocked. The form is unlocked when the Form Designer is closed. This prevents two or more people from making changes to the same form at the same time.
6. **Modification:** Changed the validation on the Node Properties dialog to disallow duplicate Node

Name for Form or System Node within the same sidebar group.

7. **Issue:** An error occurs when trying to import configuration.
   Status: This issue is resolved.

2.0 build 2

1. **Upgrade:** The Third-party controls have been upgraded to the latest version.

2. **Enhancement:** The TArScrollBox control now supports themes.

3. **Enhancement:** The Form Designer now supports saving and loading the form with errors in the script. This is so the Designer user does not have to correct all scripting errors before saving the form.

4. **Enhancement:** Added the "Enable Auto Backup for Invalid Script" check box to the Form Designer to automatically backup the script when the Apply or OK button is pressed even if it is invalid.

5. **Enhancement:** Improved dragging multiple controls in the Form Designer.

6. **Issue:** The grid column preview displays the text right justified. It should be left justified.
   Status: This issue is resolved.

2.0 build 1

1. **Issue:** The data for the lookup table is lost when the Lookup Table is used for the navigator on the Main Connection dialog and the Cancel button's DataSource property is linked to the master table and the Cancel button's CancelChildRecords property is checked, then the user presses the New button to cancel the new record.
   Status: This issue is resolved.

2. **New Feature:** Added the event OnNavigatorButtonClick (Sender: TObject; AButtonIndex: Integer; varADone: Boolean) to the TArDBGrid control which enables the user to process navigator button clicks manually. The AButtonIndex parameter identifies the clicked button:

   First = 0;
   Prior Page = 1;
   Prior = 2;
   Next = 3;
   Next Page = 4;
   Last = 5;
   Insert = 6;
   Append = 7;
   Delete = 8;
   Edit = 9;
   Post = 10;
   Cancel = 11;
   Refresh = 12;
   Save Bookmark = 13;
   Go to Bookmark = 14;
   Filter = 15;

   The *ADone* parameter specifies whether the navigator should perform default actions in

response to a button click. If you have implemented all the required actions in the **OnNavigatorButtonClick** event handler and default processing is not required, set the *ADone* parameter to **True**. In this case, the default actions are not invoked. If the parameter value, which is initially **False**, is not changed, the control will perform default actions after the event handler has been executed.

3. **New Feature:** Added the event OnAfterPost (DataSet: TDataSet) to the TArDBGrid control which occurs after an application writes the active record to the database or change log and returns to browse state.

4. **New Feature:** Added the event OnBeforePost (DataSet: TDataSet) to the TArDBGrid control which occurs before an application posts changes for the active record to the database or change log. Use this event to perform validity checks on data changes before committing them. If it encountered a validity problem, it could call Abort to cancel the Post operation.

5. **New Feature:** Added the method RefreshTable(tblData: TMSTable) to refresh a specific table, plus re-locate the current record, refresh the display format and edit format.

6. **New Feature:** Added the Label Editor popup menu for the TArLabel control. This is for showing the Label Editor dialog which allows the user to enter multiple lines for the caption value.

7. **Enhancement:** The grid footer has been changed to display with more readable display formats including commas for the numeric columns.

v2.0 build 0

1. **Major Release:** Data Manager has been completely redesigned from the ground up using the latest development tools.