Welcome to the introduction of DataFlex 18.2

Data Access Worldwide is the creator of DataFlex. Since 1976, Data Access has been delivering tools for building database applications. DataFlex is also available in a Personal Edition, which is a free, fully functional edition that can be used for non-commercial, private use.

At this point, it is assumed that you have installed the DataFlex Studio. If not, before you read through this introduction, make sure to download a copy from www.download.com or www.dataaccess.com and install it.

The goal of this introduction is to show you the steps involved in building database applications with DataFlex. We will do so by using an example scenario. In this introduction we will focus on creating a Windows application and reporting. The functionality will be limited to what can be obtained by drag & drop features in DataFlex. But don't let it stop you from getting further into the underlying programming language and framework!

Let us start by introducing some concepts to you.

Object oriented

DataFlex is an object oriented programming language. This means that for all common components, a class is already defined. This class is the definition of how such a component looks and functions and what it exactly does once it is instantiated as an object in a written program. The advantage is that a lot of technical details are taken care of for you and you can concentrate on the real functionality of the application you want to develop.

Workspace

Before you start a project, a new environment must be made on your PC, and this is a "workspace". A workspace is a set of folders in which the database, source-code and such are stored. In a workspace one or more projects can be created. These are the eventual programs, or 'executables'. The workspace and project information is viewed from within the DataFlex Studio through the Workspace Explorer. You can also see a workspace's directory structure in the Configure Workspace Properties dialog.

Database

DataFlex can work with any popular database management system. For instance, the combination of DataFlex Personal and Microsoft SQL Express is very powerful. DataFlex comes with the necessary database drivers, but for our introduction we will limit ourselves to the embedded DataFlex database. Databases are maintained in the DataFlex Studio. The Studio allows for the creation of tables, the definition of business rules and custom coding.

At a later date, you can easily convert the tables in the native database to any other supported database backend. You may use the available DataFlex tools to automatically perform the conversion of existing data as well as table structures.

Data Dictionary

When entering data, you want to have the most accurate and consistent information stored in your database. Before saving records, the entered data needs to be validated. Examples of such validations are that a State should be uppercase, or a customer may not order more than his credit-limit's amount. Those validations are referred to as 'Business Rules'. It makes sense to store these rules in one central place and this is what we call 'Data Dictionaries'. The advantage is that the rules will be applied to all applications that use data dictionaries and when a change in rules has to be programmed; the change only has to be made in one place. It also means that if you were to migrate the application to another database platform, the same business rules are automatically applied no matter what database backend is used. Data dictionaries are created and maintained via the Data Dictionary Modeler in the Studio.

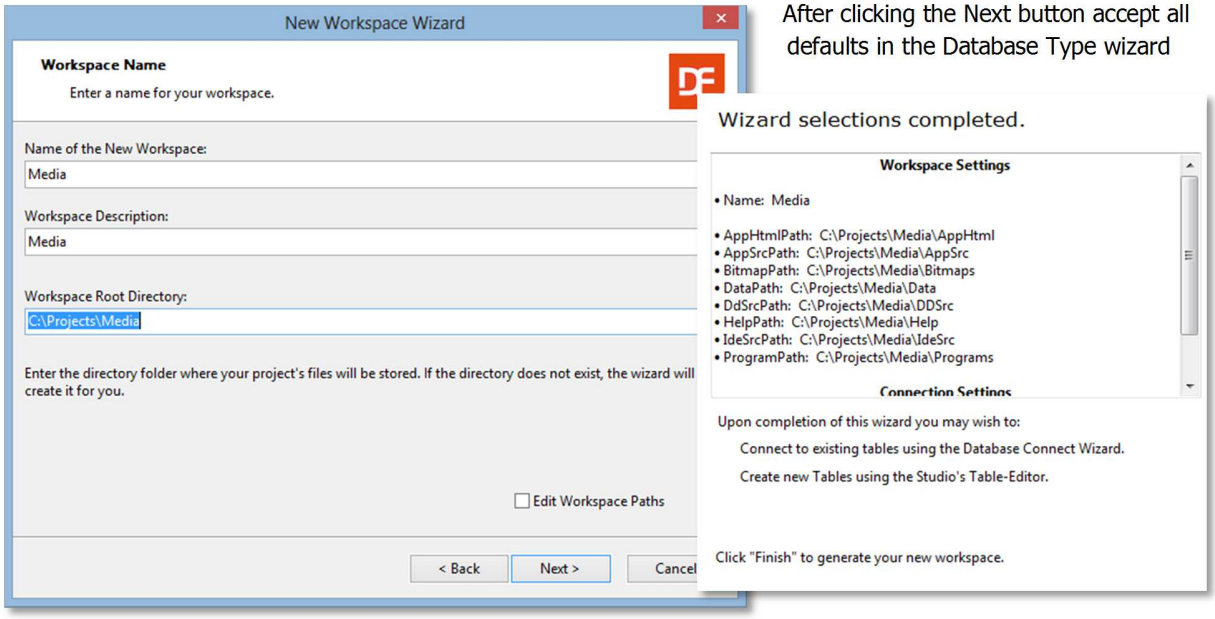
Our Example Scenario: Media

We will build a small database application that we will call Media. We will create a table Media in which we store all our CD's, DVD's, Books etc. Next, we will make a table named People to store the names of friends and relatives. These two tables will be linked to each other in such a way that you can keep track of each media's location: Do you still have it yourself, or have you lent it to a friend? In short, we will take you through the following steps:

- Create a project named Media.
  All components in a workspace must be created while there is a project active.
- Create a table named People.
  The table needs a unique key and columns to store address, phone-number, date of birth, etc. of a Person.
- Create the Person Data Entry View using the Data Entry Wizard to enter data into the People table.
  A view in DataFlex is a window, a non-modal screen dialog to enter data.
- Compile the program and test our first results.
- Create a table named 'Media'. This will have a unique key and a few columns to store Author (/Artist/Writer), what type and maybe the price and purchase date. In the table we also store the PeopleID, to be able to relate to the Person table.
- Manually, by using drag & drop, create a view to enter data into Media.
- After that, we will build in some more advanced features:
  - Make sure that the Media- and People ID's are automatically generated sequential numbers.
  - Put user-friendly pop-up lists (selection lists) in the application to easily and quickly find records in Media and People.
  - Ensure consistent format of the way the column Media.Type is entered. We will create a combo box for it and make sure the user always enters the types in a consistent manner.
  - Enable the user to store a picture with Media.

Getting Started!

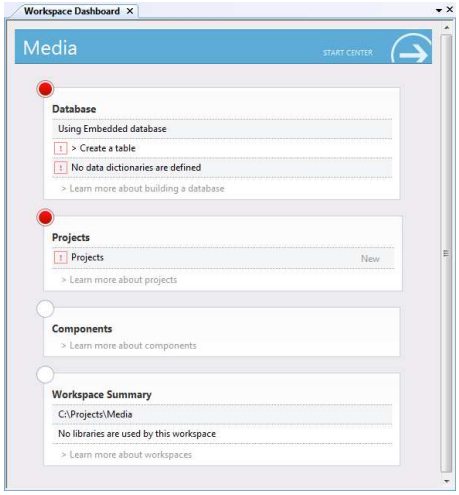
Start the DataFlex Studio. We will start by creating a new workspace. From the File menu, choose New Workspace… Give the workspace a name - in our example, we will name it Media and store it under C:\Projects\Media.



After clicking the Next button accept all defaults in the Database Type wizard page as we use the embedded database.

Prior to closing the wizard you will see a summary of the gathered information and what to do next.

The wizard creates the folders for the Media workspace and returns to the Studio so that you take the next steps.



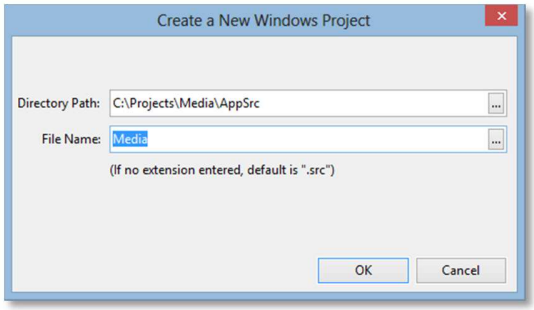
The Dashboard

After the workspace has been created the DataFlex Studio will open the workspace dashboard. The dashboard is feature that guides you through the application development. The dashboard gathers information from the workspace and shows where you might want to pay attention to. Keep the dashboard open and notice that it will automatically update during the whole process of application development.

The dashboard as shown here says that the next step is either the table or project creation. We will first create the project.

A Project

The next step is to create a project. For almost everything we create in the Studio we need an active project. There are several ways to come to the same point where you create a project and for this guide you should create a new project via the File pull-down menu, option New, and Project.



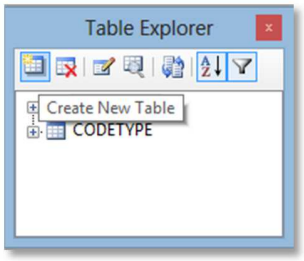
This will open a dialog in which we select "Windows Project", usually the first option in the set of icons. This choice displays a dialog where you then enter the filename and path of the project. The path defaults to the source path of the workspace created.

Click OK now, you will see the file Media.Src is created on disk and Media is made the current project. The project file contains a panel with a menu structure, default toolbars and an area where we can work with our data entry and report views.

For now we will not discuss this further and focus on the creation of a table in which we enter the records.

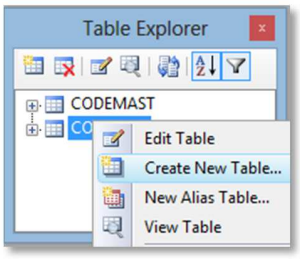
Creating the People Table

The next step in the process is to create one or more tables. Tables can be created in the DataFlex Studio. As usual there are several ways to come to the point to start the table creation. For this document we tell you to do this via the Table Explorer. Make sure you have the Table Explorer window open. If Table Explorer is not opened – by default you will find this window on the left hand side of the screen, grouped together with Code Explorer – you can activate it via the View pull-down menu, Table Explorer option or the button positioned in the views toolbar.



On the left you see the Table Explorer window. The list in the middle shows the tables already present in your workspace.

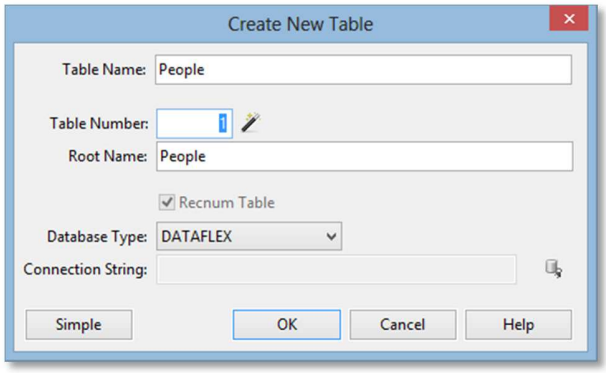
The buttons above the list can be used to create, drop or open a table for modification.



The same functionality can also be found in the floating menu that can be opened with a right mouse click in the Table Explorer window. In the floating menu you will notice a couple of data dictionary options. We will discuss the use of the data dictionaries later.

So click the "Create New Table" button (first button) or choose the "Create New Table" option from the floating menu.

In the dialog you should enter the name of the table – People – in two of the input fields (labeled Table Name and Root Name). The other parts of the dialog – including the button labeled "Simple" are not important at this moment, and are present for more advanced use.

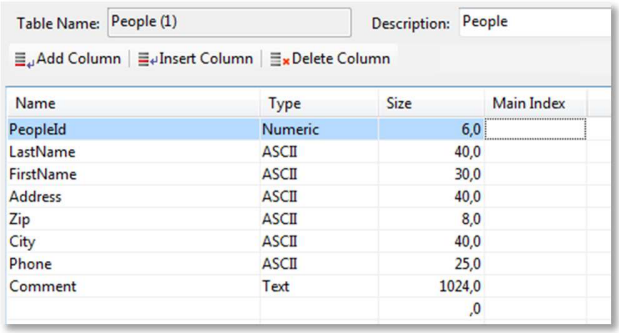


Pressing the OK button will instruct the DataFlex Studio to open a Table Editor for the new table we are creating.

The information on the Table Editor pane is divided in three areas:

- Columns
- Indexes
- Relationships

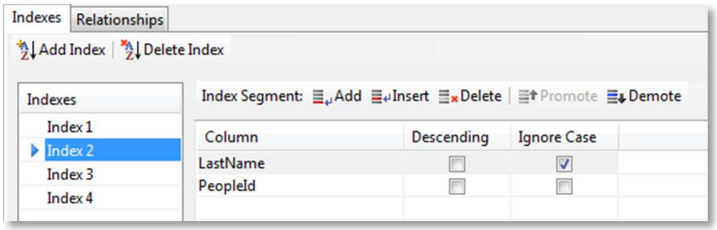
The column information is editable via a grid in which you can specify the names of the columns and their type, length and main index. Create the table so that your screen matches the following screenshot.



If you would like, you can enter more columns; you might want to store the size of their shoes, their hobbies or an e-mail address. Feel free to do so. The quick introduction assumes you have created the above columns of the given type and size. The column PeopleID is a key-field and each People record can be identified by this.

In order to look up records, we will create a couple of indexes. Let's suppose we want to allow to search by LastName, FirstName and Zip. We need to create an index for each one of them and each index need to be unique in itself.

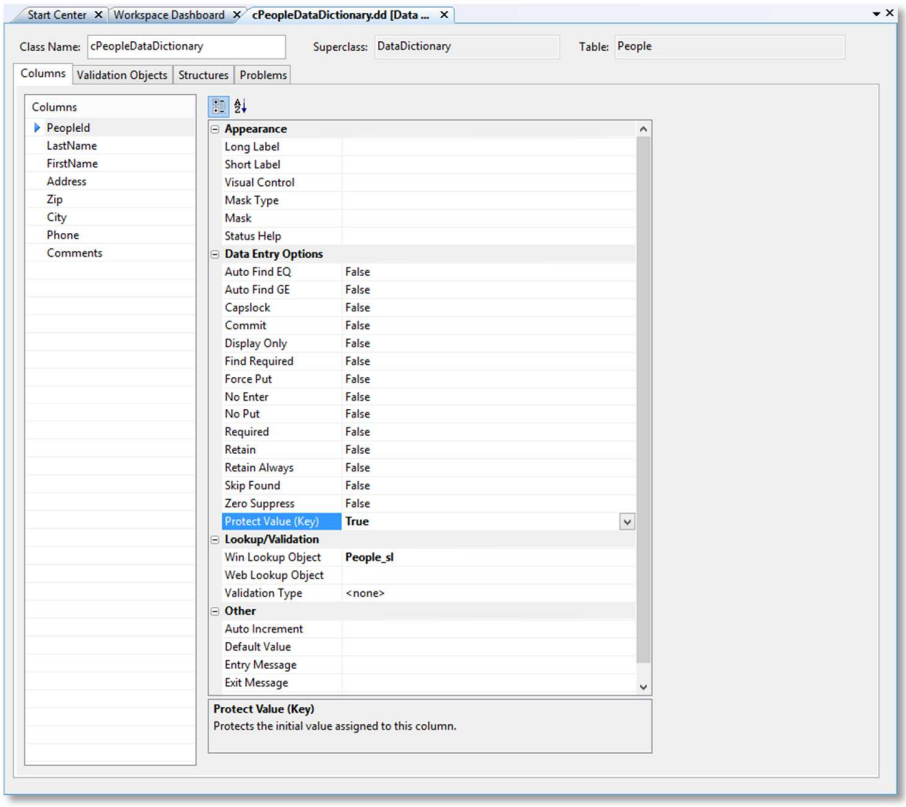
The picture shows that the second index consists of two segments: LastName and PeopleID. The latter makes the index unique. Also notice that the checkbox Ignore Case is checked. This means that when a user searches on "Johnson" the order in which it is found is not dependent on whether it is typed as "JOHNSON", "johnson" or "Johnson".



The tab has two toolbars. One - with the buttons "Add Index" and "Delete Index" – and this works on the list of indexes making it possible for you to add or remove a complete index while the other toolbar works on the grid with index segments.

The Data Dictionary for the People Table

When a column is marked as a key-field the value cannot be changed after the record has been created. PeopleID is used to link the records between Media and People and for this reason we do not want to allow a change to this value. To indicate that the PeopleID column is a key-field, we need to modify a setting in the data dictionary for the table People. While the data dictionary class is automatically created when we created the table, it is not opened for editing yet. To open the data dictionary, right click the table in the table editor and select "Open Data Dictionary" from the menu.



One new tab-page in the code editor part of the DataFlex Studio will open and the focus will be on the DD modeling tab.

Click the PeopleID column in the list of columns and find the option "Protect Value (Key)" in the list of properties as shown to the right. Change the value of this setting from False to True and we've finished setting the key field.

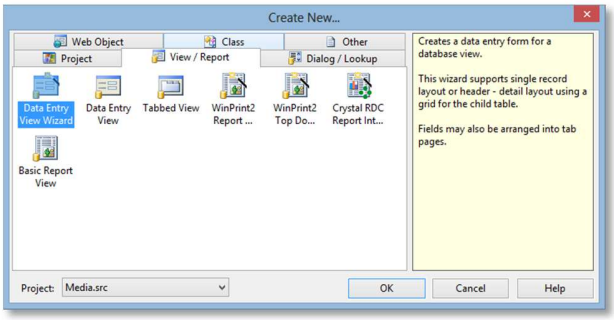
While we are on this screen we can also add a couple more Business Rules:

- Make sure that the column LastName is always entered (Required) – select LastName from the columns list and set its Required attribute to true
- Make sure that the Zip and City are always stored in uppercase (Capslock) – the same for Zip and City, changing the Capslock attribute to True

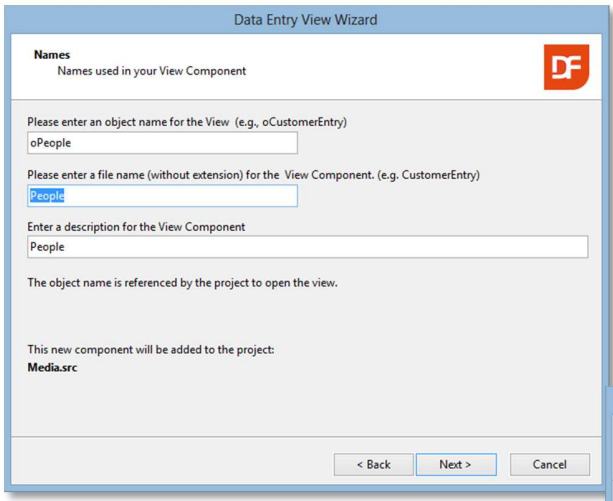
We need to save the table and data dictionary to disk. Either save each one independently, or use the Save all option. If you select the Save all option you also save changed source code which might be a good idea anyway. You may still undo the change after saving as it has no influence on the undo stack. On the other hand, closing a file would affect the stack and you may not undo all the changes.

Data Entry View

We can now create a data entry View and we will do so with the Data Entry Wizard. Again click on File, New and now View / Report and the panel shown to the right appears.

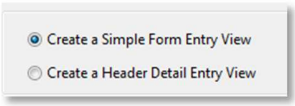


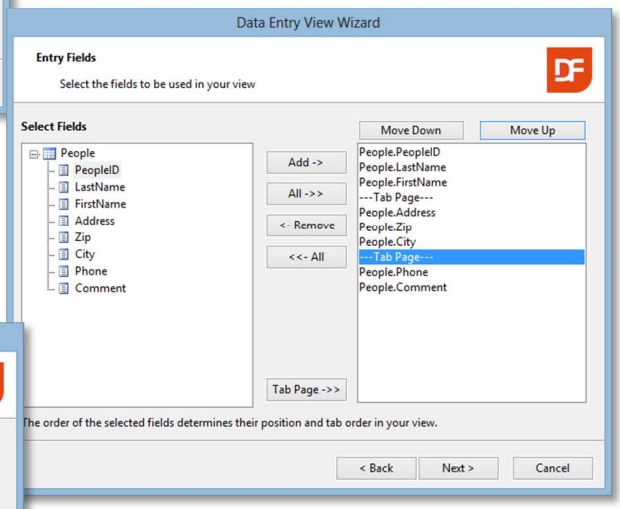
As you can see, there are a number of wizards and templates available. Choose the 'Data Entry Wizard' icon.



Enter the following information while processing the wizard:

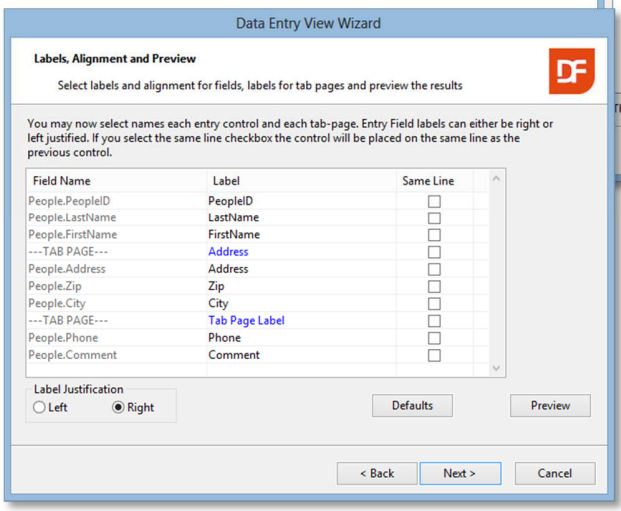
- The object gets the name oPeople, the filename is People and the description People.
- Create a simple data entry screen. Choose Simple Form Entry View.
- Choose the People table (the only table available at this time). As shown, place all the columns on the View.





By adding two Tab-pages, the wizard allows creates a better overview in the View. The Entry Fields wizard page should look like as shown to the right.

On the next wizard page you can indicate whether you want to see the labels aligned left (always placed on the left hand side of the controls) or Right and change the text of each label. In the screenshot you see we selected Right for the label justification option and changed one of the tab-page labels to "Address".



You can click the button labeled "Preview" to see what your screen would look like.

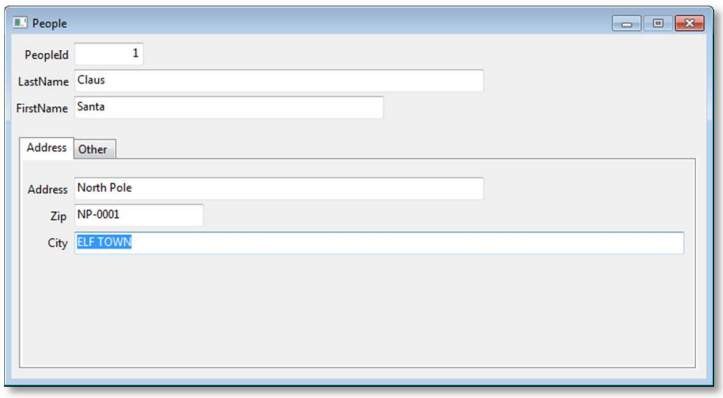
You can now click Next and Finish the wizard which will bring you back to Studio. In the Studio, the View that the wizard has made for us will be loaded automatically. This can now be easily modified simply by dragging components with the mouse, resizing them, etc. If you did not change the labels of the tab-pages, we advise you to give the tab-pages another name now. To do this, under the menu-item View, open Object Properties and click with the Mouse somewhere on the tab-page of which you want to change the label. In the list of the Properties, find "Label" and change it to "Address" and the other into "Other", or whatever you find appropriate.

Tip: *If you cannot find the Label in the properties list, you have clicked with the Mouse on the tab-dialog, or somewhere else. Make sure to click in the tab-page (body) itself and try again. For a visual clue, the corners of the selected object need to show white block markers if the page is selected and black when the container is selected.*

You are now ready to test your first DataFlex application. To do so, the project needs to be compiled; based on the generated source-code, the compiler will make an executable, an .EXE, to run the application. The testing can be started in different ways:

1. Press F8 to only compile.
2. Press F5 to run. When compilation is needed, the compiler will be automatically launched.
3. Choose the Project pull-down and select Compile.
4. Click on the little blue triangle icon in the toolbar.

If you selected option 2 or 4 and the compilation is finished, the application starts. You may bring up the People view and you may now enter some records. Notice the following:

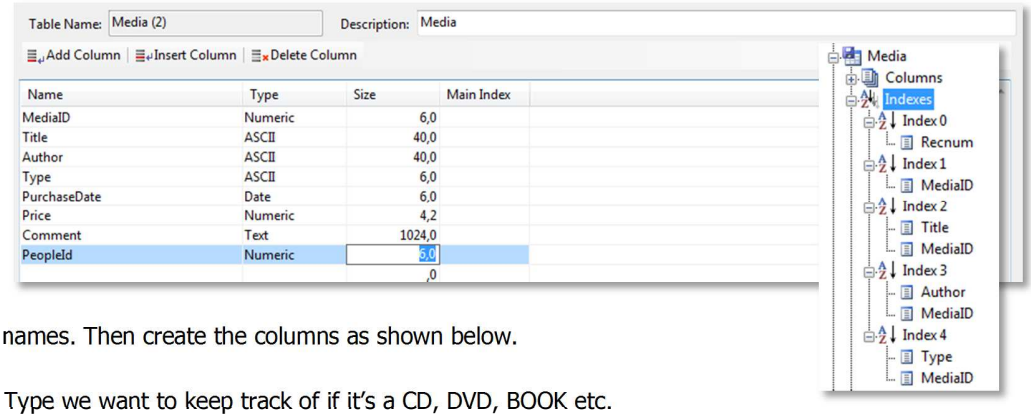


- Begin with the first record by using PeopleID '1'. The next is '2' and so on (more about this later).
- Save records by pressing the F2 key, or clicking the save icon on the toolbar.
- Clearing the screen can be done by F5 (the record will not be deleted).
- Once you have entered multiple records, you can browse through them by using F7 (Previous) and F8 (Next).
- F7 and F8 work only if the cursor is placed in the columns PeopleID, LastName, FirstName or City. And that is logical: These were the columns that we defined indexes on.
- Key in a partial name in LastName and hit F9. This will find (equal) based on the given (partial) string. Try this!

Close the application to return to the Studio.

New Table Media

The creation of the table for Media is the next step in our process. So click again the New Table button in Table Explorer. Enter the value "Media" for the table and the root names. Then create the columns as shown below.

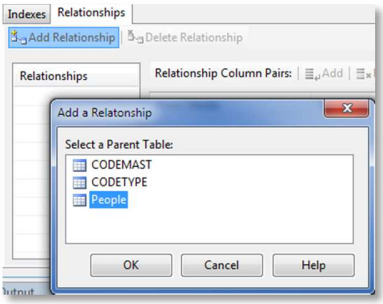


Note:

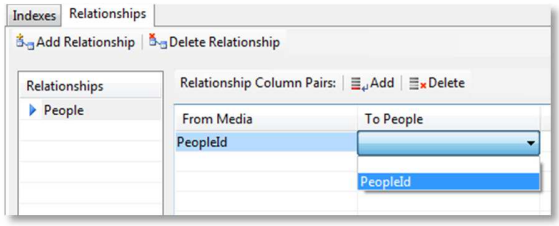
- In the column Type we want to keep track of if it's a CD, DVD, BOOK etc.
- The PurchaseDate is of the type Date.
- The Price is numeric with the 4.2 format. This indicates that the price can have 4 digits before and two after the decimal point.
- The column PeopleID will be used in the relation with the People table. We will, therefore, ensure that it is of the same type (numeric) and same size (6 digits) as the column in the other table.

MediaID will be the key field. Besides that, create indexes on Title, Author and Type. To make them unique, we add MediaID as last segment of each index. We also choose to switch on the Case Insensitive option.

Tip: *Until now we have explained that an index is there to quickly and easily find a record. Indexes have another important function, which is fast sorting in reports. It is not difficult to create more indexes at a later time if you need this for certain reports.*



The Media table will contain records of our media in the possession of a certain person. In technical terms this means there is a relationship between the tables Media and People. Therefore, let us create the relationship between the two tables. Choose the tab-page named Relationships and choose the first toolbar button (Add relationship). A dialog with tables pops up and you should select the table People from this list. Relationships are almost always defined from many to one, so in this case, from Media to People.

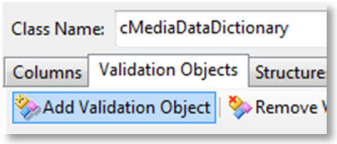


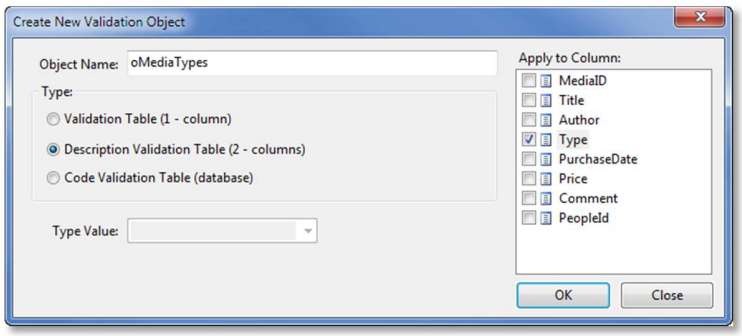
The selection of the parent table opens the option to specify from which child column(s) to which parent column(s) the relationship is made. The column type and length of the related columns must match. The parent column (usually the key-field) must be uniquely indexed. The result should look like the picture on the right.

The Business Rules for Media

Finally, we will add some more Business Rules in the Data Dictionary. For this, the table needs to be saved first. The MediaID column will be a Key Field and the Title column is required. You should be able to do this with the guidelines given with the People data dictionary.

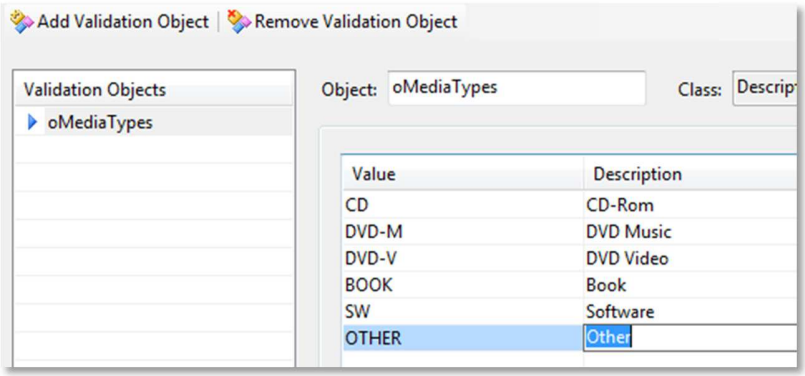
The values for Type should always be entered in uppercase (the Capslock attribute needs to be selected), but let's add something extra. We want the user to use consistent naming when entering Types. If it is a CD-Rom, its Type should be entered as 'CD'. If it's a book, it should always be entered as 'BOOK'. If such details were not entered consistently, think about how difficult it would be to make a selection ('Show me all books') when making a report. Therefore we will make a simple validation table on the column Type. You do this via the Validation Objects tab-page. Click the "Add Validation Object" button and enter the following information in the dialog.



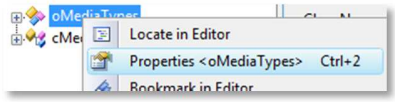


The column selected for the Validation object is Type and the type of the validation object is – as shown – the description validation table. Enter the object name as specified. Object names at this level need to have a unique name.

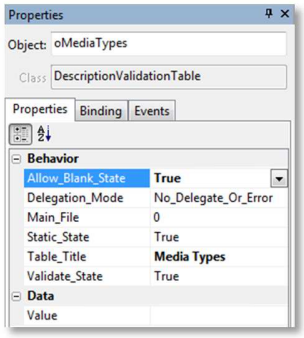
After you clicked the OK button you can start entering values for the table. We suggest you enter the values as shown.



Feel free to add more optional Types.



Via one of the properties of the validation table you can indicate whether the value may be left blank or not. If the value of



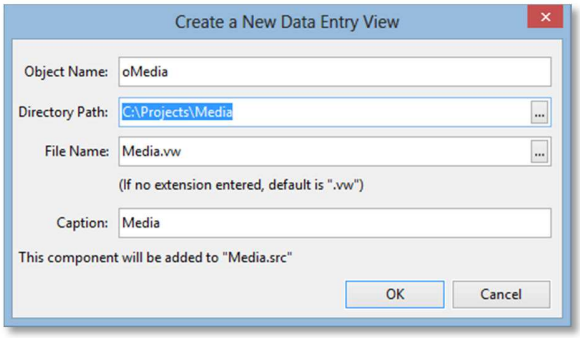
Allow_Blank_State is not set to True the user must select a value from the list when creating or editing a record. Make your own choice.

The user can pick one of the options from a drop-down list or a popup list. If using the popup list – which is the default – the Table_Title property setting becomes important to give more explanation to the user about what the dialog purpose is.

Tip: *It is of no importance at this time, but take a look at the tab-page called Structures where you can see that the People table obviously is added to the structure with Media. This is a hint that validations do not only apply to single tables; related tables are also validated.*

Creating the Media Data Entry View

The second view will be the Media Data Entry View. This time the wizard will not be used to create a data entry View. This means you will learn how to make a data entry view in a more manual way. From the menu under File, choose New, View / Report, but now click on the second icon: Data Entry View.



After that, enter "oMedia" for the object name and the file on disk will be named Media.Vw. The Caption value is the text displayed in the view's caption-bar.

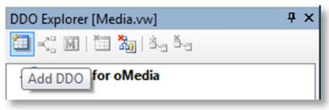
Tip: The screen will already note that the component will be added to the project Media –it will also be placed in the menu of the application automatically.

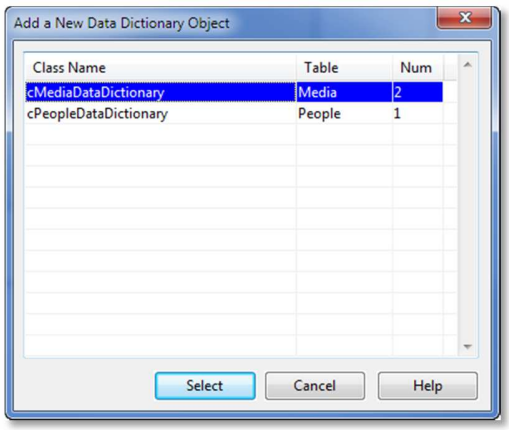
A new concept: Data Awareness

Before we continue, let's explain a new concept: **Data Awareness**. DataFlex is a tool to build database applications. After the first sample we saw that it takes a View (interface) to enter data into the database. The several components in the interface are apparently coupled to the underlying columns in the tables. That's right, that's exactly what happened. But in fact there is an extra layer in between; the Data Dictionaries. DataFlex knows different types of components. An important difference is whether components are 'data aware', or not. If they are, you only have to assign Data Dictionary objects (DDO's) to it in order to have the desired data (tables) at your disposal. Data aware web controls make use of data binding usually via an Entry_Item statement.

To continue. You are now looking at an empty dbView in the Studio. The first thing we need to do is to decide which tables we want to maintain in this View. In the menu, under View, open DDO Explorer. In DDO Explorer no tables

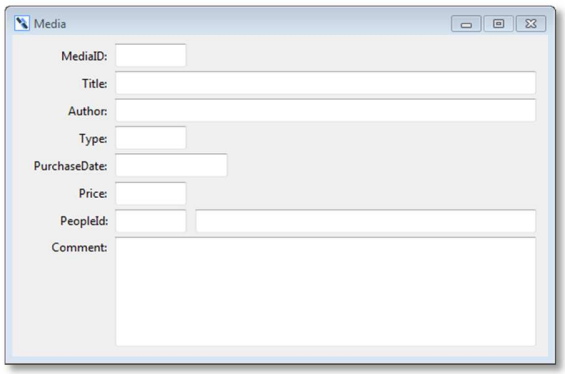
are selected yet. Adding a DDO is done via clicking the "Add DDO" button. You can do the same from the floating menu.



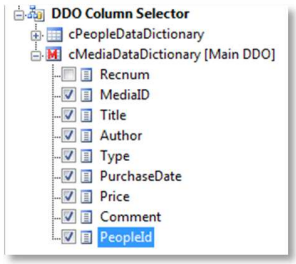


In the dialog that opens you have to select the right data dictionary for this new view. Since we want to edit (create, modify, delete) the table Media, select the cMediaDataDictionary class (highlighted in the picture).

The Studio automatically makes the DDO for the Media table the main DDO and because Media has a relationship to a parent table (People), the DDO for that table will also be automatically added. When the choices are not correct, you can change these via the floating menu on each of the DDOs and apply the change. In our example this is now correct, so no action needed.

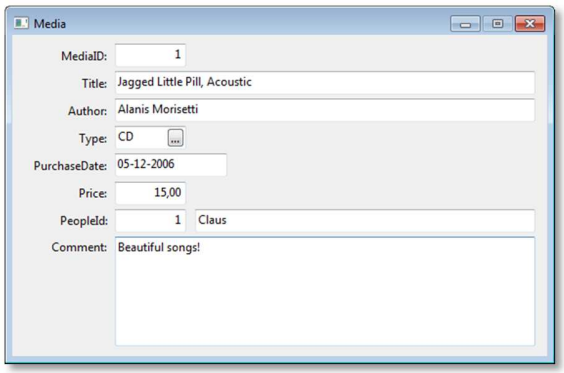


In the DDO Column Selector select all columns of Media (fill in all but the Recnum checkboxes) and drag them onto the View. Similarly, drag & drop the column LastName from People table onto the View. Align the object with the PeopleID column. Remove the label-text from LastName in Object Properties. Change the layout in such a way that it looks similar to what is shown. For example, all labels are right aligned at a distance of 2 from the control.



To see what the application looks like after it's compiled and started you click the run button or press F5.

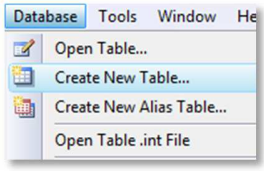
Enter a couple of media records. Let the Media ID start at '1' and choose the Person related to it. Once you have created a few media records, use F7 and F8 to browse through the data.



The Order in which Data is Entered

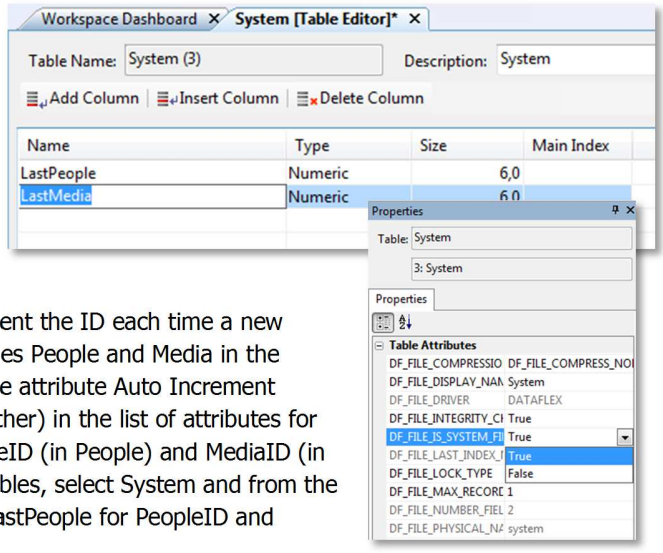
When you changed the positions from the columns PeopleID/LastName and Comment, as in this sample, you will notice while entering data that the order in which you navigate through the columns has not changed accordingly. We can change the position the columns are displayed in the View, but that does not automatically change the order in which you navigate through the columns. We can easily adjust that: From the menu, under View, open Code Explorer. Under the object oMedia, select the object oMedia_Comment and right-click on it. Now move the column two positions down (Choose 'Move Object Down' or Alt+DownArrow). This takes care of that. You can also do this from the toolbar button at the top of the pane.

Automatically Generate Key Fields



For People as well as Media we defined unique, numeric keys. This number stored in those key fields is in fact not relevant to the user. In addition, it would be troublesome for users to remember what the last given number was when they try to create a new record. This can easily be taken care of; it only takes two steps:

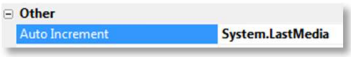
1. Create an extra table. In this table we will always store only one (1) record. This is called a system table. In this table we define two columns only: LastPeople and LastMedia. These columns are numeric, 6 digits.
2. The Data Dictionaries of Media and People will take care of generating these ID's automatically, using the system-file.



Create the System table to match the picture displayed on the right.

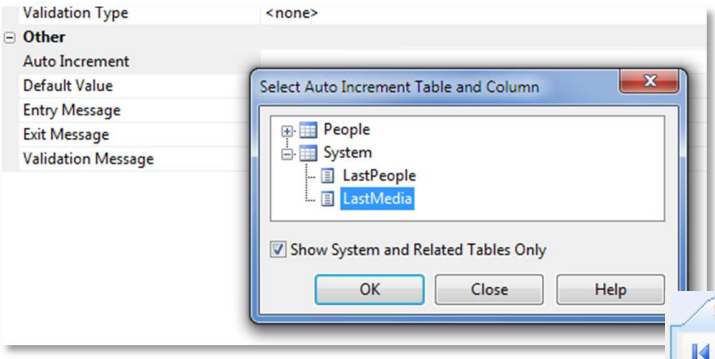
To make sure it actually becomes a system-file, select True for the table attribute DF_FILE_IS_SYSTEM_FILE.

We want the Data Dictionaries to automatically increment the ID each time a new record is saved. Open the data dictionaries for the tables People and Media in the Studio. Look for the attribute Auto Increment (Grouped under Other) in the list of attributes for the columns PeopleID (in People) and MediaID (in Media) and click the prompt button. From the list of tables, select System and from the columns the correct source data column – those are LastPeople for PeopleID and LastMedia for MediaID.

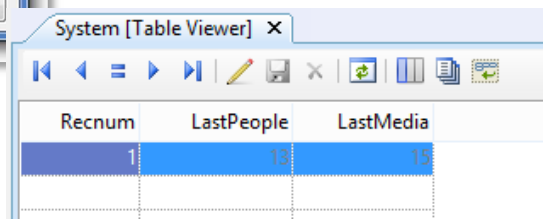


Note: The value can be taken from a system table or a parent table. That is why you see the checkbox labeled "Show System and Related Tables Only".

Ok. This should work, if not it is because you have already created some records, with ID's 1, 2, 3 etc. The first time we will use our automated increment function it will try save a record with ID of '1' and that won't work because that value already exists. ID's should always be unique – it is a key field! Our new application can't save any new records. We could delete our existing records, but there's another way to solve this.



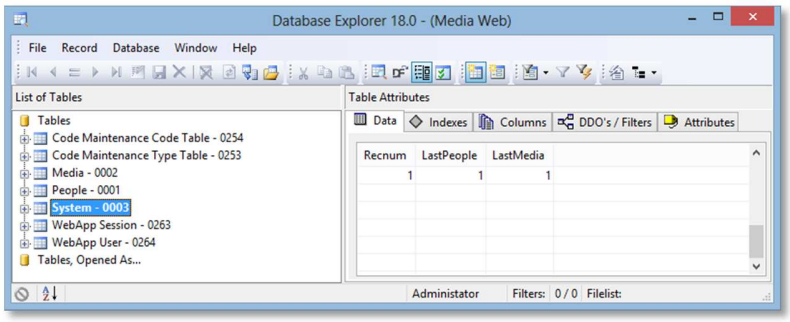
Right click the "System" table in the table explorer and choose "View Table". The contents of the table is opened in a tab-page in the design area of the Studio.



Change the value for LastPeople and LastMedia to the highest used number. Let's assume that the last ID you have entered was 10. Then enter the value 10 for both the LastPeople and LastMedia. Next time a record is created, the IDs will be automatically set to 11

As an alternative you can do the same – and more – via another useful tool from the Studio Tools menu: Database Explorer.

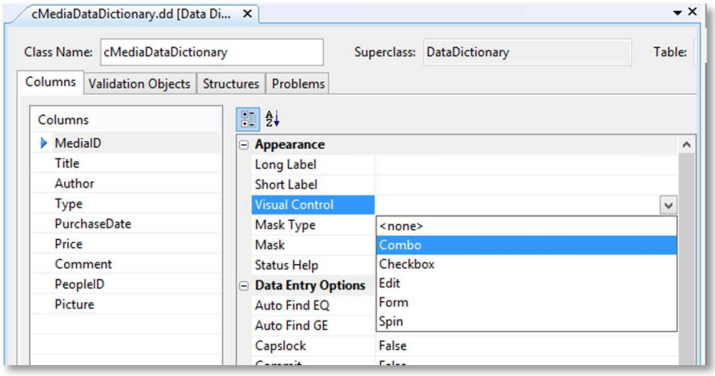
Database Explorer (often called DBExplorer) provides a quick means to directly edit data in tables. It is a typical tool for developers, but be careful if you use it as it bypasses any safety or validations you have built into your applications.



The first thing we have to do is to make it possible to change data. By default, Database Explorer is configured in its most secure mode which is set to only allow us to read data. Therefore, click on the little icon at the very bottom-left. Change it from a red colored icon to a gray colored one.

Data Dictionary changes

Before we compile and test our application, let's make two more improvements. Open the Media data dictionary if it is not already open. In Column Settings for the column Type, go to the group Appearance, find the option "Visual Control" and select the Combo. Every time this column is dragged to a new view during development, we want it to be a Combobox by default. For the same column; in the group Data Entry Options, find the option Capslock and select True.



Set Auto Find EQ to True for the column MediaID.

Change the Type Control

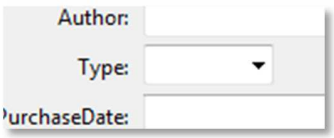
The change we just made (visual control) will only be applied when dragging the column to a new view, but we already have our media view. To change the control used for the Type column we need to program a little bit. In Code Explorer find the object oMedia_Type. Have the Studio take you to the source code of the object via the option "Locate in editor" available when you right-click on the object name in Code Explorer. The code line should look like:

```
Object oMedia_Type is a dbForm
    Entry_Item Media.Type
    Set Location to 50 60
```

It's a dbForm, but we want it to become a dbComboForm object. Change the code to:

```
Object oMedia_Type is a dbComboForm
    Entry_Item Media.Type
    Set Location to 50 60
```

Press the F7 key. This toggles the Studio between design and source-code views and the dbForm has been visually changed into a combo box. Make the combo box a bit wider and press F5 to compile and start the application. Now it is no longer necessary enter to ID's (PeopleID or MediaID) when creating a new record.



We have made the following improvements:

- The ID's for People and Media are automatically serialized/incremented.
- Filling in an existing ID at Media, followed by pressing the Tab will automatically find the Media that goes with that ID, this is what Autofind EQ does. It makes sense to make the same change on ID for People.
- Entering data for Media Types now makes much more sense. Always uppercase, in a combo-box, which is the appropriate visual control for this type of data-entry.
- Setting the Appearance of Media.Type in the data dictionary to Combo-box, this column will always be displayed as a combo box by default.

Note: By changing the type of control manually, the component becomes no longer self-contained, it is non-autonomous. In this situation, the project will compile and run but there are situations where that is not going to happen. To fix it, add "Use Dfcentry.pkg" to the list of USE statements at the top of the component.

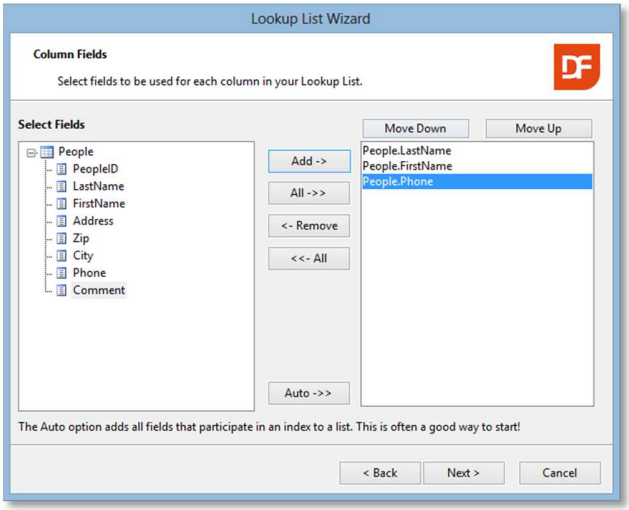
```
Use cMediaDataDictionary.dd
Use Dfcentry.pkg
Use DFEntry.pkg
Use cDbTextEdit.pkg
```

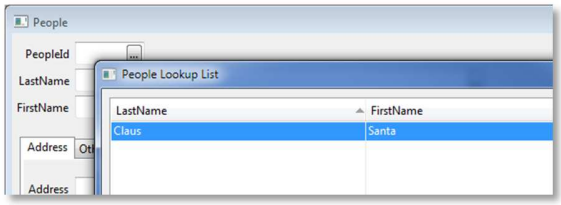
Selection Lists

We can make more improvements: If there are only a few records in the tables, it is not a problem to find the right data using F7 and F8, but when the tables grow, we must offer a better way for finding the right records, so we will add look-up lists, also known as Selection Lists.

From the File menu in Studio, choose New, Dialog / Lookup and start-up the Lookup Wizard. Let's first make a selection list for People, in which we place LastName, FirstName and Phone (number).

We accept all defaults on the other wizard pages and at the end, we compile the program again (F5). You can see that buttons are automatically added to LastName and Firstname (Prompt Buttons).

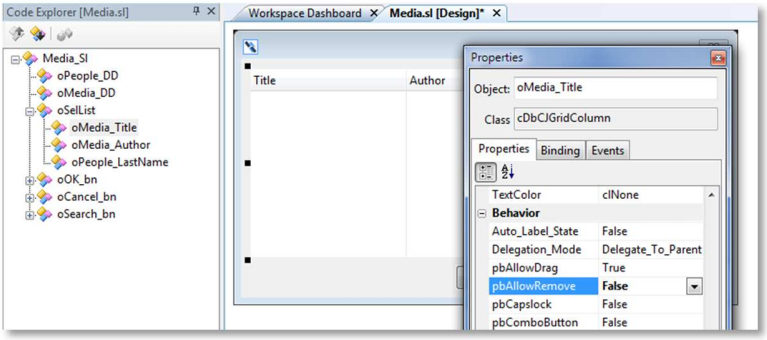




Clicking on the prompt button (or pressing F4) brings the just created selection list to the screen, offering the following standard functionality:

- Depending on the column in which the cursor is, the sort order of the list changes accordingly.
- Simply start typing the desired LastName and a pop-up screen appears. Pressing Enter will take you to the closest record. Because we use indexes, finding records in a set of just a few records will be as fast as when searching a record in a set of a few million!

Now it should be easy to create a selection list for the Media table. Maybe you want to try another way instead of using the wizard; by drag & drop. In that case, here are some tips: Choose File, New,Dialog / Lookup: Table Lookup. Enter "oMediaLookup" as the object name.

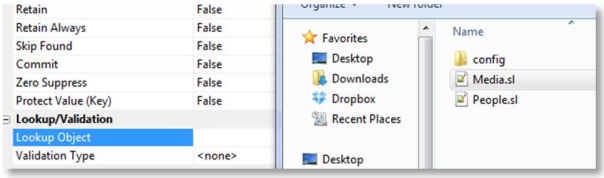


Use the DDO Selector to choose the correct tables (hint: Select the Media Datadictionary) and columns (hint: Title and Author from Media and LastName from People).

To change details in the new Lookup or one of the columns (including their order) use Code Explorer and the Properties panel. There are many options you can change and control. For example, by default users are allowed to remove columns from the list. If you do not want that, you can change the list property called pbAllowColumnRemove or the column property pbAllowRemove.

One of the tasks automatically performed when using the Table Look up wizard is the connection of the lookup list with one or more columns from the table via the data dictionary. Since we did not use the wizard this time, we need to make these connections ourselves.

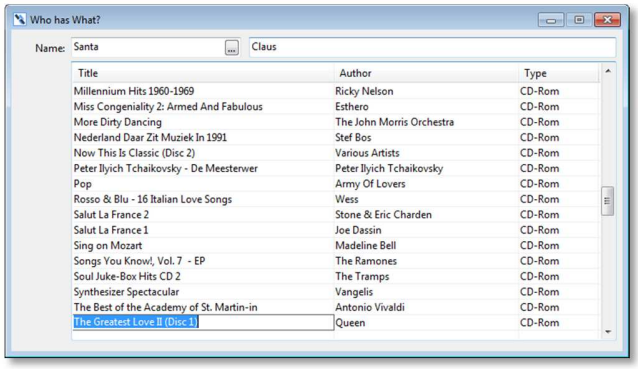
Open the Media data dictionary (if no longer opened), click the column(s) for which you want the oMediaLookup to appear and select the Media.SI file for the column property Lookup Object.



Show All Media Borrowed

Let's do something a bit more advanced. It is not so difficult to create a View where a user can browse through People and display in a grid all the Media that each person has borrowed. The control used here is called a cDbCJGrid, which can be found among the Data Controls in the Class Palette.

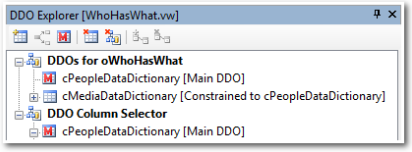
To browse through People and show only the appropriate Media, we need to create a view with a constraint, a filter.



Here is how you can do this:

- Create a new empty View (Select Data Entry View template).
- Use DDO Explorer to select Media and People.
- Drag FirstName and LastName onto the view.
- Use Object Properties to change the labels for FirstName and LastName.
- Align the objects horizontally.
- Remove the Prompt Button from the LastName column (find the Prompt_Button_Mode in the list of Object Properties and set it to PBPromptOff). This does not remove the lookup list completely (you can still press F4 or select lookup from toolbar or floating menu) but it removes the visual hint. To remove the lookup list you would need to add a line of code to the people data dictionary object.

```
Object oPeople_DD is a cPeopleDataDictionary
    Set Field_Prompt_Object Field People.LastName to 0
End_Object
```

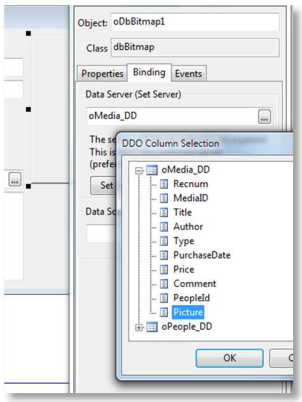


- Check in the DDO Explorer if the constraint is set to People (see image). If this is not the case, right-click on the Media Table in DDO Explorer and choose Add/Change constraint: Add constraint on people (also check Main DD) and the structure should be as shown in the image on the right.

- With drag & drop create an object of the cDbCJGrid class from the class palette.
- From the DDO Column Selector drag the columns Title, Author and Type from the Media table to the grid.

Pictures

The very last enhancement we make to the Windows application is to add pictures for the Media we catalog. Let's say that we want to store a picture with each Media record. First, we need to change the table:



- Open the Media table if not open, and add a column named Picture.
- Specify the type ASCII, and a length of 255. The actual picture will not be stored here. Each record will hold a reference to the picture location (folder) and name.
- Open the Media view again in Studio and make it somewhat bigger.
- In the Class Palette among DataControls find the dbBitmap, drag and drop it onto the Media view.
- In the Properties panel under the Binding tab-page, assign the Data Source (Entry Item) to the Media.Picture column.
- Compile and start the application (F5).

In the application, find a Media record and double-click on the area where the bitmap should be stored. A common File Dialog will pop-up. Select the image you want to store for that record and make sure you save the record!

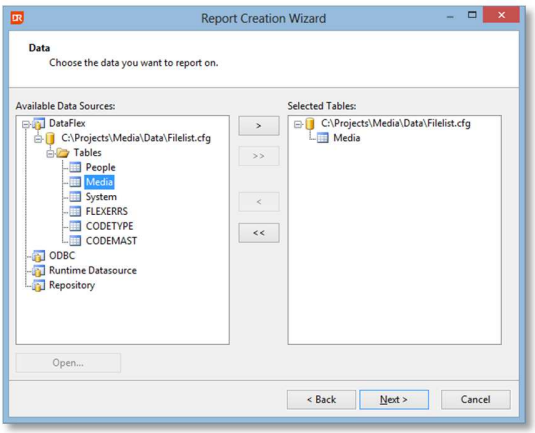
Note: Out of the box DataFlex "only" supports bitmaps (all images are drawn by Windows as bitmap) but via the free add-on Graphics Library you can add TIFF, JPG, GIF and many other graphic formats. The connection is as easy as described above (we only need to connect to the library and use a different class name).

Reports and Lists

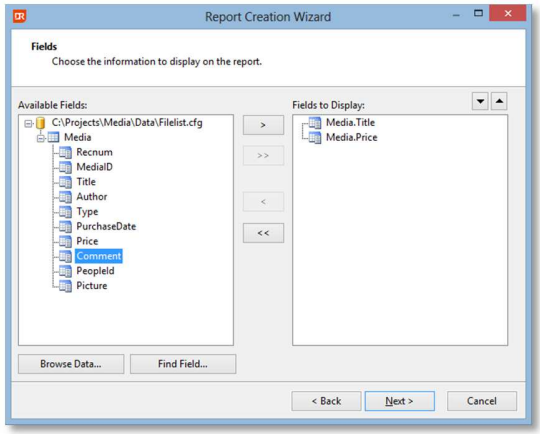
One of the most powerful ways to create reports and integrate them in DataFlex is by using DataFlex Reports and the DataFlex Reports Integration Library. This is a wizard that automatically integrates a report in your Windows application. The end-user will be able to start the report from the menu and still be able to influence the sort order, output device, and even selection criteria. It's simple: First create a report with DataFlex Reports, and then start the wizard – the rest is self-explanatory.

Note: If you do not have a license for DataFlex Reports, please contact the Data Access sales representative in your region to receive an evaluation license, or to purchase a license.

Start DataFlex Reports and select File, New. Then choose Standard Report. You can also press the Ctrl+N key combination. In the wizard select DataFlex as your data source and point to the filelist of your workspace (in the data folder). Select the Media table from the list of tables.



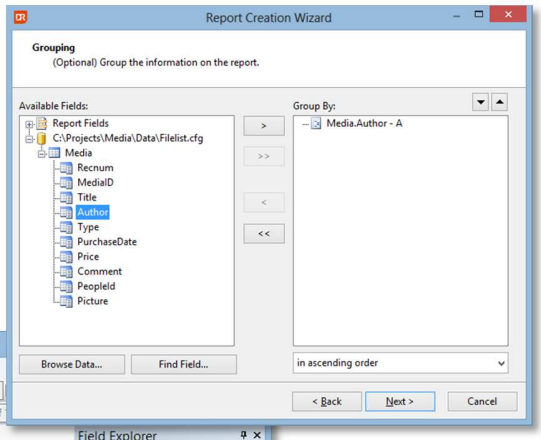
In the Fields wizard page select the columns Title and Price.

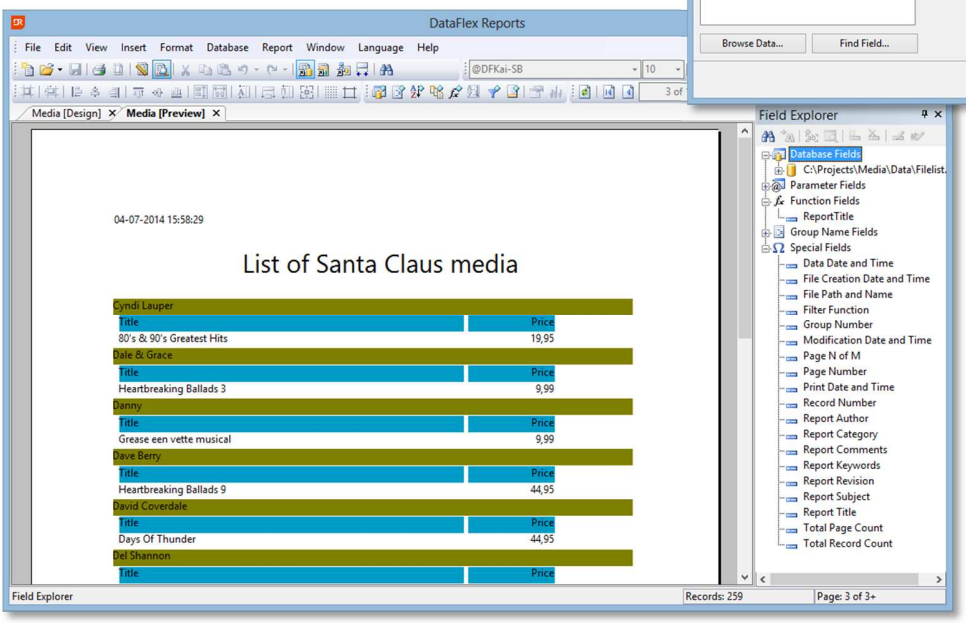


In the Grouping wizard page select the column Author. The page summary, data filters and repository can be skipped for this report.

After finishing preview the report in the designer.

To integrate the above report in your Windows project you will need to connect to the DataFlex Reports Integration library (optionally installed with the developer edition of DataFlex Reports). The library comes with a Wizard to integrate the report into your applications. The result of the Wizard is a Report View component, automatically added to the Reports pull-down of your current project. In the self-explanatory wizard you can select from four different kinds of preview views. The third option gives the most freedom, the standard is the dynamic preview view. This makes it possible to run multiple report and compare result. The big advantage of the report integration is the programmers API that makes it possible to change filters, change data paths at runtime etc.



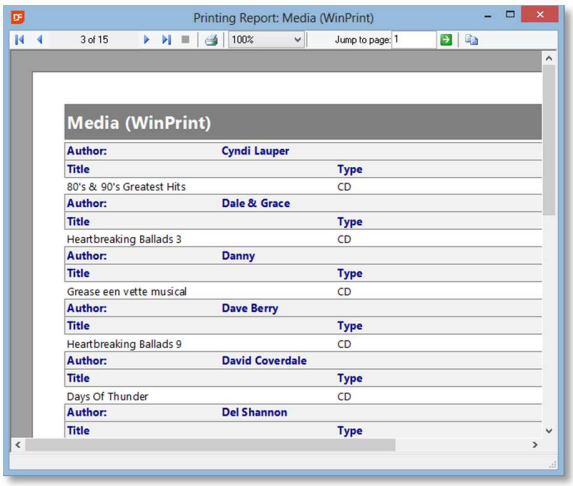


With DataFlex Reports you can fill an array of data in your DataFlex application and pass it to the DataFlex Reports print engine as the data-source to print from. This makes it possible to create reports on data that is not in a database.

An alternative way of producing reports is by using Winprint. A Winprint Report is fully programmable in DataFlex. Fortunately there is a way to create Winprint Reports by using a wizard for a quick start. It takes just a few seconds to make a report that lists all Media, with a break on Authors.

With a report fully coded inside the DataFlex programming language you can process records while being printed. As an example you can flag an invoice as printed (store the print date) while printing.

The downside of such a report, is that for every small change in the layout requested by your users, you need to change program code and redistribute the application. Of course this also increases your revenues.



Get Started!

This concludes this Quick Introduction Guide for DataFlex. If you have not done so yet, take a look at DataFlex Content Management written in and delivered with DataFlex. You can find a Quick Introduction document like this one at www.dataaccess.eu under "Get Started!"

To learn more about DataFlex, visit the following sites to learn more about the product and its creator:

- www.dataaccess.com
- www.electos.com
- www.visualdatapump.com

We Look Forward to Helping You to Get Started With DataFlex!

Other related sites:

- support.dataaccess.com/forums
- www.dynamicai.eu (Business Intelligence tool)

If the documentation, help-files or the forums don't provide you with answers, feel free to ask for assistance via e-mail. Visit the Data Access website for the support options in your region.

Another very good resource is new extensive training guide called "Discovering Visual DataFlex" that contains more than 600 pages. This book has been made available for each revision of DataFlex since the beginning of 2008. Please contact your Data Access sales representative if you would like to purchase a copy of this book. The picture shows the Visual DataFlex 15.1 version of the book. An addendum that lets you discover the new and changed behavior added in version 16.x is available. There will be an update of the book available later.