Welcome to the introduction of DataFlex 20.0

DataFlex is a complete software platform for rapidly developing and deploying Windows, web, mobile and cloud business applications - fast! It is one of many products Data Access Worldwide has delivered to the software development community since 1976. DataFlex is for application development and is available as a Commercial Edition, which is used by businesses, institutions and government agencies to build and deploy applications, and as a Personal Edition, which is a free, fully functional edition that can be used for non-commercial, private use.
At this point, you should have installed the DataFlex Studio. If not, before you read through this introduction, make sure to download a copy from www.dataaccess.com and install it.
Once you have DataFlex Studio installed, you will be ready to follow through this document to learn the steps involved in building a basic database application with DataFlex that includes data entry and reporting. Once you get the basics down, get further into the underlying programming language and framework to take advantage of the whole range of features that DataFlex has to offer.

First, let's start by covering some concepts important to fully understand the DataFlex framework.

Object oriented

DataFlex is an object oriented programming (OOP) language. This means that all common components belong to a pre-defined class. The classes hold the definition of how such components look, function, and what they exactly do. Components are defined as objects of certain class in your application and that guarantees consistency and reusability of code throughout your project.
A major advantage of using OOP is that a lot of technical details are defined in the classes and you can concentrate on the real functionality of the application you want to develop customizing components as you need.

Workspace

Before you start a project, you need to create a new environment where all the elements of your project – such as programs, objects, and rules – will be stored. That environment is called a *workspace*. A workspace is a set of folders in which the database, source-code and all files necessary to your project are stored.
A workspace can hold one or more projects, which are the eventual programs or 'executables'. In the Studio, Workspace Dashboard gives you an overview of the workspace and more details are displayed in Workspace Explorer and Code Explorer. You can also see a workspace's directory structure in the Configure Workspace Properties dialog under the Tools menu in the Studio.

Database

DataFlex can work with any popular database management system (DBMS). DataFlex comes with the necessary database Connectivity Kits that allow you to utilize those DBMS in your project, but for the purposes of this introduction we will limit ourselves to the embedded DataFlex database, our native 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 time, you may easily convert native tables to any other supported database backend using the available DataFlex conversion tools that will not only perform the conversion of table structures but also their existing data.

Data Dictionary

Data dictionaries are the business rules in your workspace that keep your data accurate and consistent. For example, before saving records, the entered data needs to be validated – states should be uppercase and of a certain value, and customers may not order more than the value specified in his credit limit.
Those validations are referred to as business rules and the files where they are stored are called data dictionaries. All applications using data dictionaries will follow those same rules and if any rules change, only the data dictionaries

need to be adjusted and the new rules will be applied throughout all applications.
Having data dictionaries also means that if you were to migrate the application to another database platform, the same business rules will automatically be 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

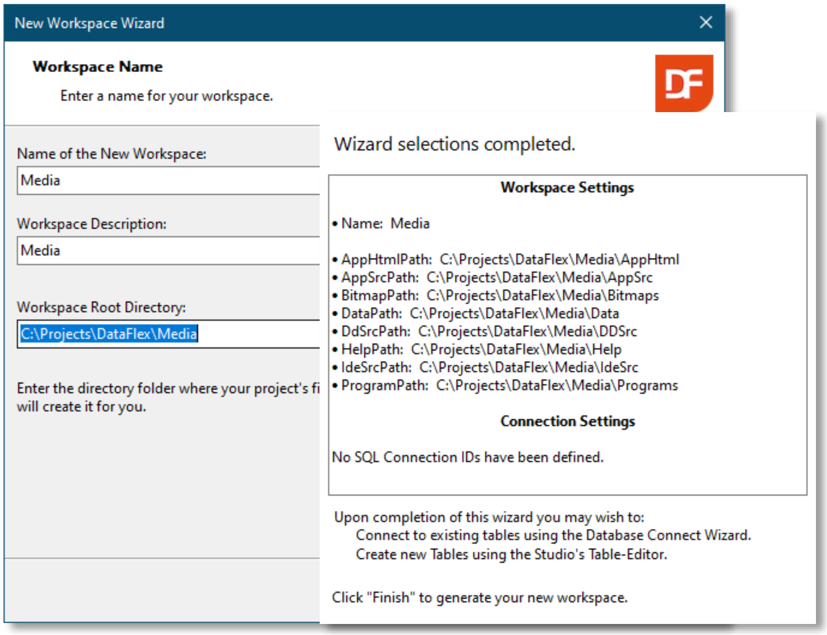
We will build a small database application that we will call Media. We will create a table Media in which we store all our CD's, DVD's, Books etc. Next, we will make a table named People to store the names of friends and relatives. These two tables will be linked to each other in such a way that you can keep track of the location of each item in your media library.
This example 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 each row 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 begin 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\DataFlex\Media.
After clicking the Next button skip the Database Connection 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 selected folders for the Media workspace and returns to the Studio so that you may take the next steps.



New Workspace Wizard

Workspace Name

Enter a name for your workspace.

Name of the New Workspace: Media

Workspace Description: Media

Workspace Root Directory: C:\Projects\DataFlex\Media

Enter the directory folder where your project's fi
will create it for you.

Wizard selections completed.

Workspace Settings

- Name: Media
- AppHtmlPath: C:\Projects\DataFlex\Media\AppHtml
- AppSrcPath: C:\Projects\DataFlex\Media\AppSrc
- BitmapPath: C:\Projects\DataFlex\Media\Bitmaps
- DataPath: C:\Projects\DataFlex\Media\Data
- DdSrcPath: C:\Projects\DataFlex\Media\DDSrc
- HelpPath: C:\Projects\DataFlex\Media\Help
- IdeSrcPath: C:\Projects\DataFlex\Media\IdeSrc
- ProgramPath: C:\Projects\DataFlex\Media\Programs

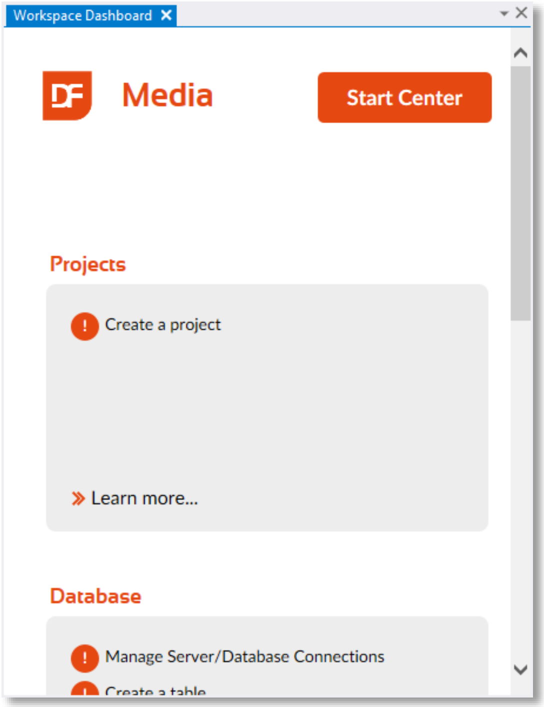
Connection Settings

No SQL Connection IDs have been defined.

Upon completion of this wizard you may wish to:
Connect to existing tables using the Database Connect Wizard.
Create new Tables using the Studio's Table-Editor.

Click "Finish" to generate your new workspace.

Note: Workspace information can be altered later on via the DataFlex Studio and – if folder renaming is desired – the Windows Explorer.



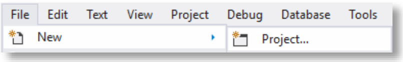
The Dashboard

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 what you may want to pay attention to.
The dashboard as shown here says that the next step is either the table or project creation. We will first create the project.

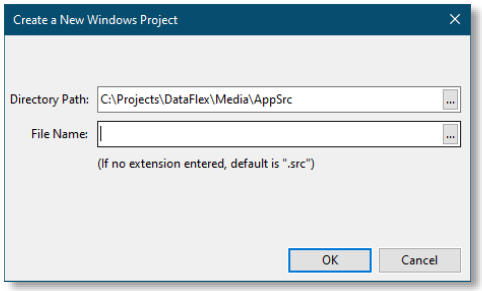
Tip: *Keep the dashboard open and notice that it will automatically update during the whole process of application development.*

Tip: *At any time in the project development, you can add TODO markers that are picked up by the dashboard. This means you can use the dashboard as a project management tool.*

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. For example via the File pull-down menu, option New, and Project but you can also click the 'Create a Project' option in the Workspace Dashboard.



Directory Path: C:\Projects\DataFlex\Media\AppSrc

File Name:

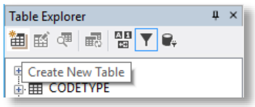
(If no extension entered, default is ".src")

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 and 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 data.

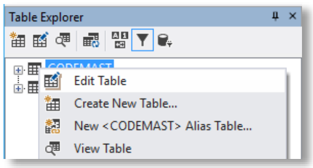
Creating the People Table

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 start the table creation. For this document, do this via the Table Explorer. Make sure you have the Table Explorer window open. If Table Explorer is not open – 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 or edit a table for modification, view the contents of a table, refresh table definition, sorting and/or filtering the list of tables and as last one opening the SQL Connection Manager.



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.

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

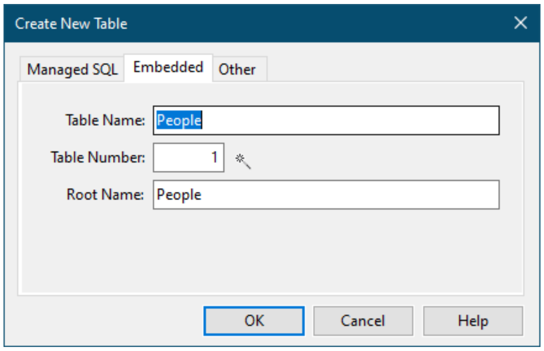


Table Name: People

Table Number: 1

Root Name: People

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 table number value can be changed but the suggestion is good to go.

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

The Table Editor panel contains areas with Columns, Indexes and Relationships information.

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 to match the following screenshot.

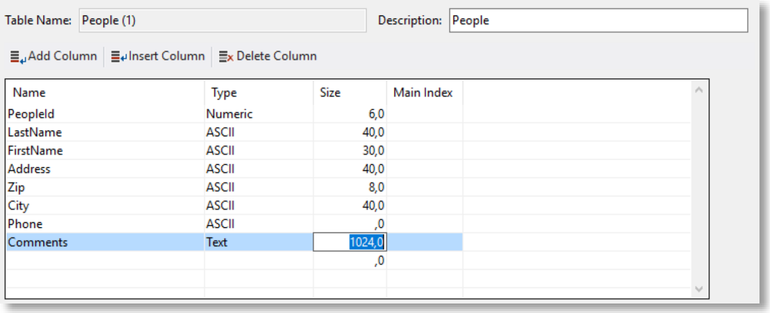
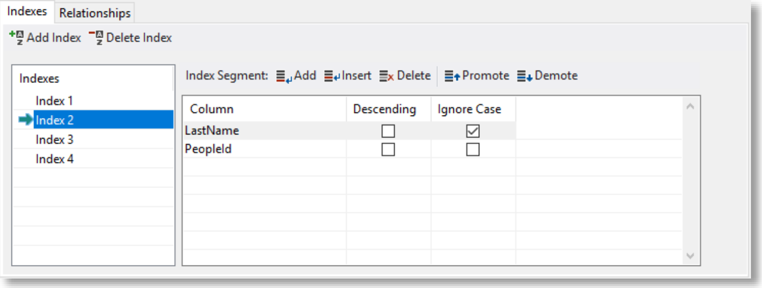


Table Name: People (1) Description: People

Name	Type	Size	Main Index
PeopleId	Numeric	6,0	
LastName	ASCII	40,0	
FirstName	ASCII	30,0	
Address	ASCII	40,0	
Zip	ASCII	8,0	
City	ASCII	40,0	
Phone	ASCII	,0	
Comments	Text	1024,0	
		,0	

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. To identify a specific row in the People table create the column PeopleId as a key-field.

In order to look up people, we will create a couple of indexes. Let us suppose we want to allow to search by LastName, FirstName and Zip. We need to create an index for each one of those columns and each index need to be unique by itself.



Column	Descending	Ignore Case
LastName	□	☑
PeopleId	□	□

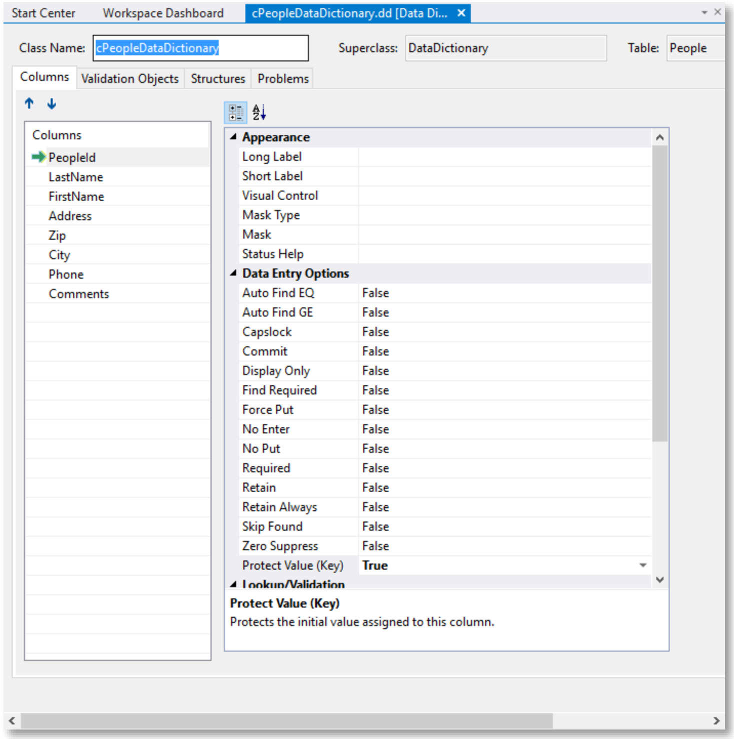
The picture shows that the second index consists of two segments: LastName and PeopleID, the latter making the index unique. Also notice that the checkbox *Ignore Case* is checked. This means that when a user searches on "Johnson" the order in which it is found is not dependent on whether it is typed 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. Change the order of index segments with the "promote" and "demote" tool-bar buttons if they are in not in the desired order.

Save the table structure for the table People by pressing the Ctrl+S key-combination or click the Save tool-bar button.

The Business Rules

When a column is marked as a key-field (‘Protect value (Key)’ attribute) the value cannot be changed after the record has been created. PeopleID is used to link the records between the later to be created Media table and the People table and for this reason we do not want to allow this value to be changed. 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 the key field will be set.



Class Name: cPeopleDataDictionary
Superclass: DataDictionary
Table: People

Columns
- PeopleId
- LastName
- FirstName
- Address
- Zip
- City
- Phone
- Comments

Appearance	
Long Label	
Short Label	
Visual Control	
Mask Type	
Mask	
Status Help	
Data Entry Options	
Auto Find EQ	False
Auto Find GE	False
Capslock	False
Commit	False
Display Only	False
Find Required	False
Force Put	False
No Enter	False
No Put	False
Required	False
Retain	False
Retain Always	False
Skip Found	False
Zero Suppress	False
Protect Value (Key)	**True**
Lookup/Validation	

Protect Value (Key)
Protects the initial value assigned to this column.

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

- Make sure that the column LastName is always entered – 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 – select Zip from the columns list and change the Capslock attribute to True; then repeat the same for City

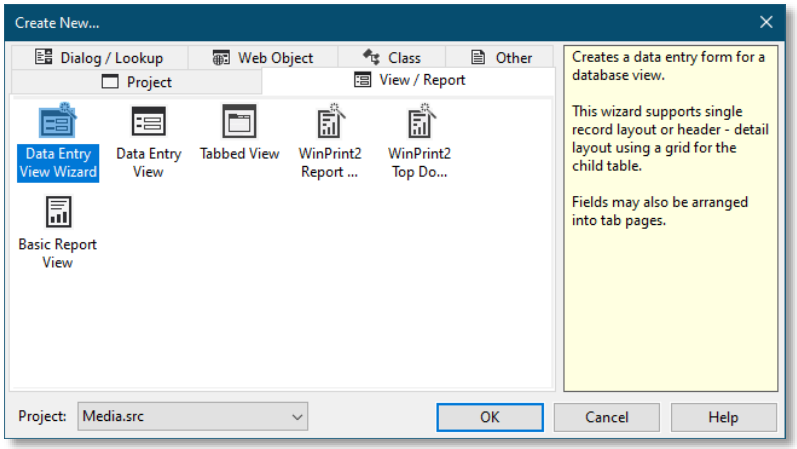
We need to save the table and data dictionary to disk. Either save each one independently, or use the Save all option. If you select the Save all option you also save changed source code which might be a good idea anyway. You may still undo changes after saving them as long as you do not close the file. Closing a file would affect the undo stack and you may not be able to 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 Create New panel just like the one pictured on the right will be displayed.

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

Enter the following information while processing the wizard:



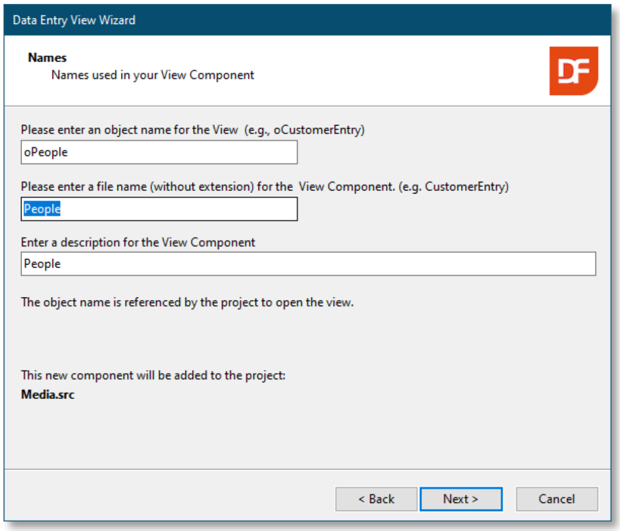
Creates a data entry form for a database view.

This wizard supports single record layout or header - detail layout using a grid for the child table.

Fields may also be arranged into tab pages.

Project: Media.src

- oPeople for the object name
- People for the file name
- People for the description
- Create a simple data entry screen. Choose Simple Form Entry View.
- Choose the People table (the only table available at this time).



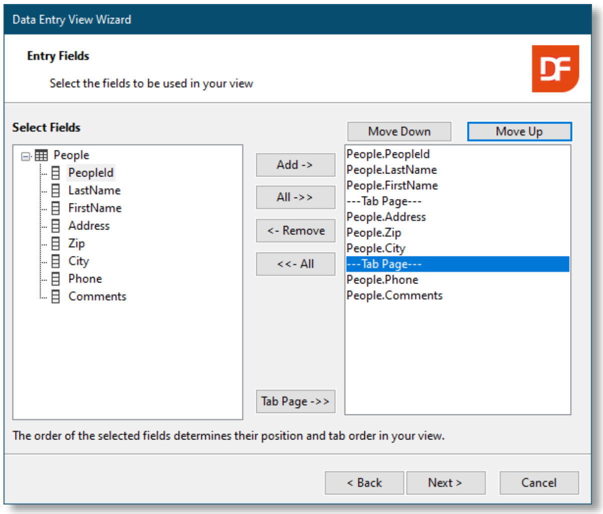
Please enter an object name for the View (e.g., oCustomerEntry)
oPeople

Please enter a file name (without extension) for the View Component. (e.g. CustomerEntry)
People

Enter a description for the View Component
People

The object name is referenced by the project to open the view.

This new component will be added to the project:
Media.src



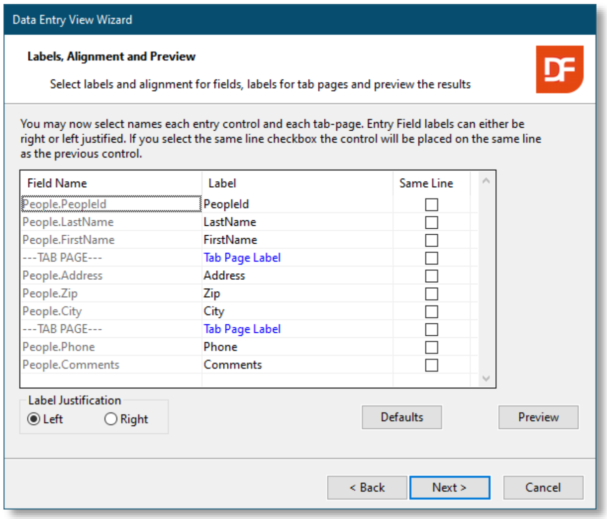
People
PeopleId
LastName
FirstName
Address
Zip
City
Phone
Comments

People.PeopleId
People.LastName
People.FirstName
---Tab Page---
People.Address
People.Zip
People.City
---Tab Page---
People.Phone
People.Comments

The order of the selected fields determines their position and tab order in your view.

As shown, place all the columns on the View.

Add two Tab-pages as indicated so the View will be better organized. In the end, the Entry Fields wizard page should look like as shown to the left.


Data Entry View Wizard
Labels, Alignment and Preview
Select labels and alignment for fields, labels for tab pages and preview the results
Defaults
Preview
< Back
Next >
Cancel

You may now select names each entry control and each tab-page. Entry Field labels can either be right or left justified. If you select the same line checkbox the control will be placed on the same line as the previous control.

Field Name	Label	Same Line
People.PeopleId	PeopleId	☐
People.LastName	LastName	☐
People.FirstName	FirstName	☐
---TAB PAGE---	Tab Page Label	☐
People.Address	Address	☐
People.Zip	Zip	☐
People.City	City	☐
---TAB PAGE---	Tab Page Label	☐
People.Phone	Phone	☐
People.Comments	Comments	☐

Label Justification
● Left ○ 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".

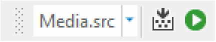
You can click the button labeled "Preview" to see what your screen would look like.

You can now click Next and Finish to be brought 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 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 (.EXE) that will be what is run; your application. The testing can be started in different ways:

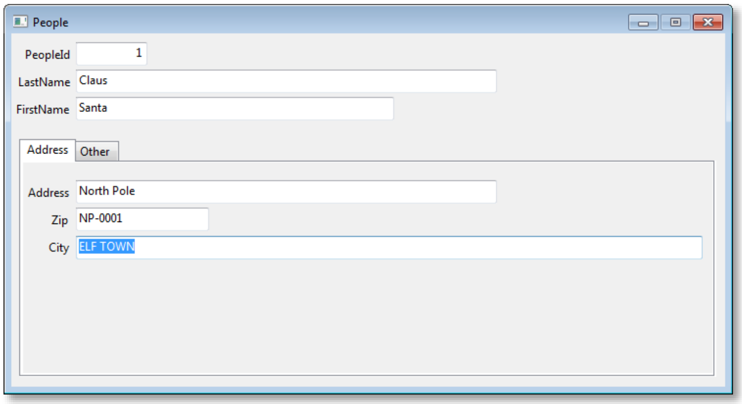
1. Press F8 to only compile.



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 green 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 data. Notice the following:

- Begin with the first person by using PeopleID '1'. The next is '2' and so on (more about this later)
- Save data by pressing the F2 key, or clicking the save icon on the toolbar
- Clearing the screen can be done by F5 (the row will not be deleted)
- Once you have entered multiple rows, 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 because 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!



PeopleId: 1
LastName: Claus
FirstName: Santa

Address: North Pole
Zip: NP-0001
City: ELF TOWN

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.

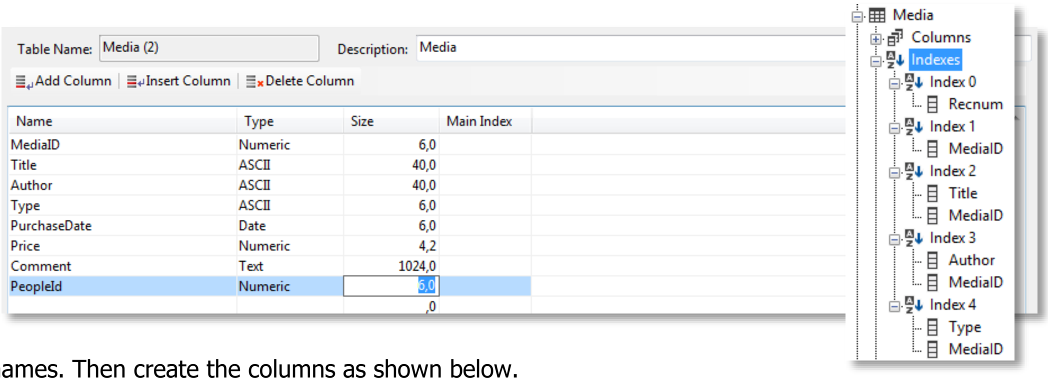


Table Name: Media (2)
Description: Media

Name	Type	Size	Main Index
MediaID	Numeric	6,0	
Title	ASCII	40,0	
Author	ASCII	40,0	
Type	ASCII	6,0	
PurchaseDate	Date	6,0	
Price	Numeric	4,2	
Comment	Text	1024,0	
PeopleId	Numeric	6,0	
		,0	

- Media
  - Columns
  - Indexes
    - Index 0
      - Recnum
    - Index 1
      - MediaID
    - Index 2
      - Title
      - MediaID
    - Index 3
      - Author
      - MediaID
    - Index 4
      - Type
      - MediaID

Note:

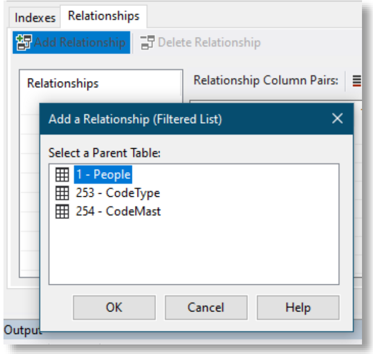
- In the column Type we want to record the type of the media – CDs, DVDs, BOOKs etc.
- PurchaseDate is a Date type column
- Price is numeric with the 4.2 format. This indicates that the price can have 4 digits before and two after the decimal point.
- The column PeopleID will be used in the relationship 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 People table.

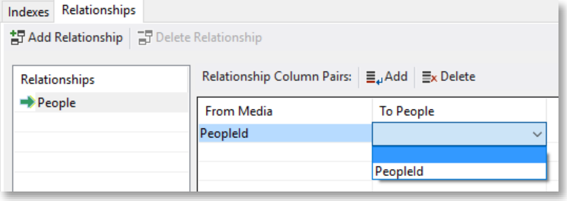
MediaID will be the key field. Create an index for it as well for Title, Author and Type. To make those indexes unique, add MediaID as last segment for each index and select the Case Insensitive option.

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.



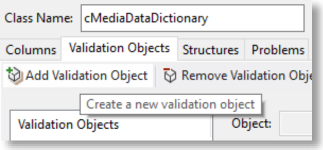


From Media	To People
PeopleId	PeopleId

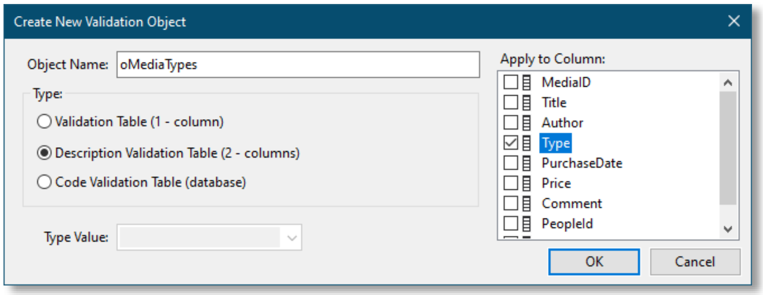
Business Rules for Media

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 us 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 is 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.


Class Name: cMediaDataDictionary

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.


Create New Validation Object

Object Name: oMediaTypes

Type:

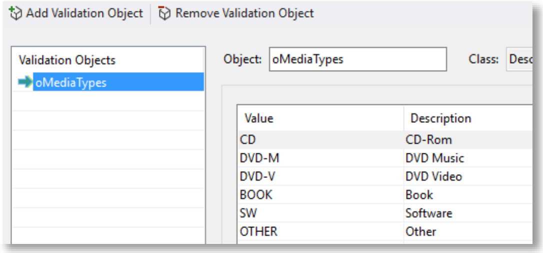
○ Validation Table (1 - column)
● Description Validation Table (2 - columns)
○ Code Validation Table (database)

Type Value:

Apply to Column:

☐ MediaID
☐ Title
☐ Author
☑ Type
☐ PurchaseDate
☐ Price
☐ Comment
☐ PeopleId

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 be unique.

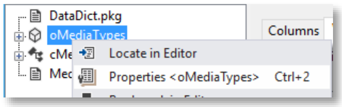


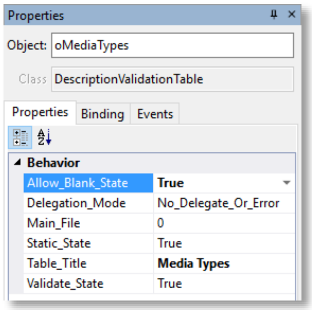
Object: oMediaTypes Class: Desc

Value	Description
CD	CD-Rom
DVD-M	DVD Music
DVD-V	DVD Video
BOOK	Book
SW	Software
OTHER	Other

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.





Object: oMediaTypes

Class DescriptionValidationTable

Behavior

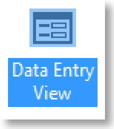
Allow_Blank_State	True
Delegation_Mode	No_Delegate_Or_Error
Main_File	0
Static_State	True
Table_Title	Media Types
Validate_State	True

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 row. Configure this setting as you please.

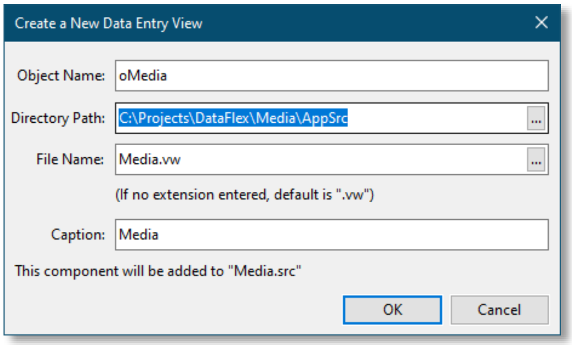
The user can pick one of the options from a drop-down list or a look-up list. If using the look-up list – which is the default – the Table_Title property setting becomes important to give more explanation to the user about what the purpose of the dialog is.

Tip: *As a side note, 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 we create 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 in a less automated way and will have more control over what happens. From the menu under File, choose New, View / Report, but now click on the second icon: Data Entry View.


Create a New Data Entry View

Object Name: oMedia

Directory Path: C:\Projects\DataFlex\Media\AppSrc

File Name: Media.vw

(If no extension entered, default is ".vw")

Caption: Media

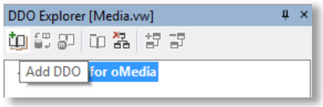
This component will be added to "Media.src"

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.

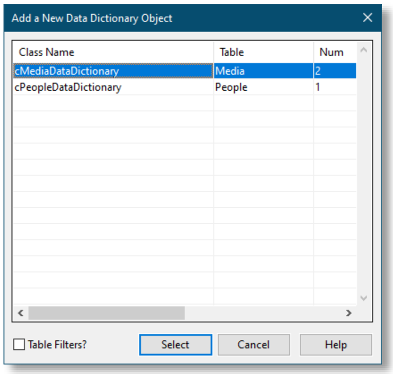
Tip: The screen indicates that the component will be added to the project Media – it will also be placed in the menu of the application automatically.

A new concept: Data Awareness

Before we continue, let us 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 is right, that is 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.


DDO Explorer [Media.vw]

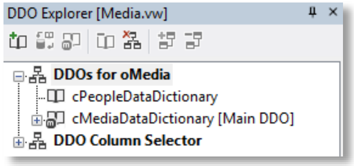
You are now looking at an empty dbView in the Studio. The first thing we need to do is to decide which tables we want to maintain in this View. In the menu, under View, open DDO Explorer. In DDO Explorer no tables are selected yet. Adding a DDO is done via clicking the "Add DDO" button. You can do the same from the floating menu.


Add a New Data Dictionary Object

Class Name	Table	Num
cMediaDataDictionary	Media	2
cPeopleDataDictionary	People	1

☐ Table Filters?

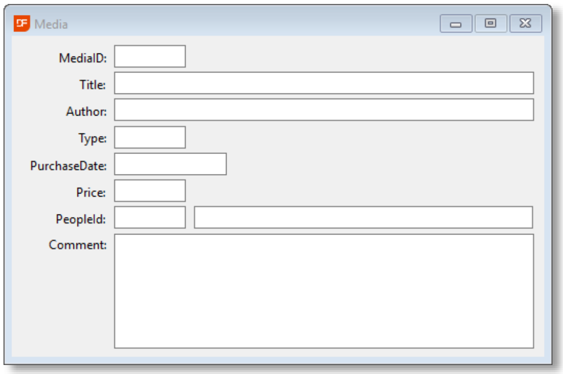
In the dialog that opens you have to select the right data dictionary for this new view. Since we want to edit (create, modify, delete) the table Media, select the cMediaDataDictionary class (highlighted in the picture).

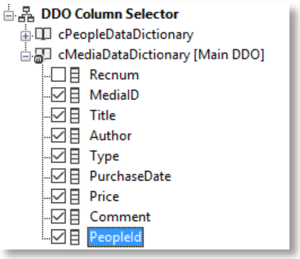

DDO Explorer [Media.vw]

DDOs for oMedia
cPeopleDataDictionary
cMediaDataDictionary [Main DDO]
DDO Column Selector

The 'Table Filters?' checkbox applies the filters defined for the Table Explorer to this list. This helps when the workspace' database contains a large number of tables.

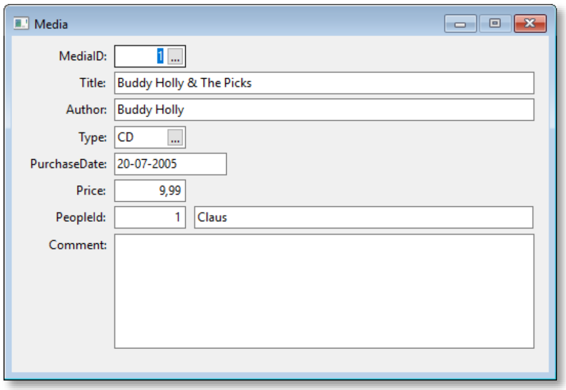
The Studio automatically makes the DDO for the Media table the main DDO and because Media has a relationship to a parent table (People), the DDO for that table will also be automatically added. When the choices are not correct, you can change these via the floating menu on each of the DDOs and apply the change. In our example this is correct, so we keep the default settings.





□	Recnum
☑	MediaID
☑	Title
☑	Author
☑	Type
☑	PurchaseDate
☑	Price
☑	Comment
☑	PeopleId

From the DDO Column Selector select all columns of Media (select all but the Recnum checkbox) 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 using the Object Properties. Change the layout in such a way that it looks similar to what is pictured on this page. For example, all labels are right aligned at a distance of 2 from the control.



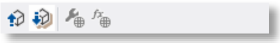
MediaID:	1
Title:	Buddy Holly & The Picks
Author:	Buddy Holly
Type:	CD
PurchaseDate:	20-07-2005
Price:	9,99
PeopleId:	1 Claus
Comment:	

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

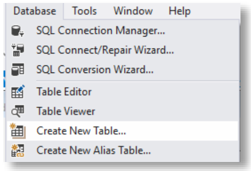
Enter a couple of media rows. Let the Media ID start at '1' and choose the Person related to it. Once you have created a few media records, use F7 and F8 to browse through the data.

The Order in which Data is Entered

When you changed the positions from the controls for 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 controls has not changed accordingly. We can change the position the controls are displayed in the View, but that does not automatically change the order in which you navigate through the controls. 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 entry 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 row. 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.

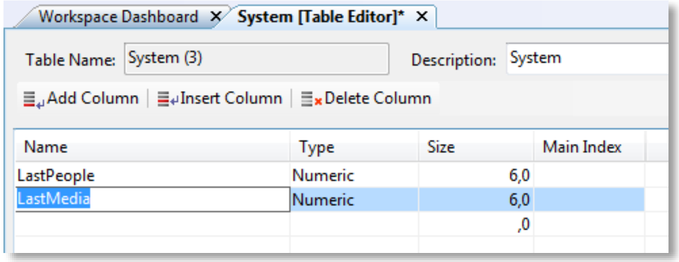
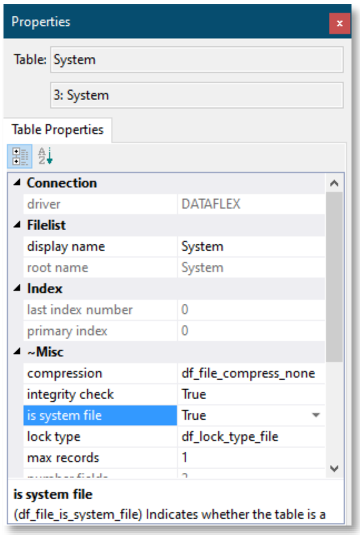


Table Name: System (3)
Description: System

Name	Type	Size	Main Index
LastPeople	Numeric	6,0	
LastMedia	Numeric	6,0	
		,0	



Properties

Table: System

3: System

Table Properties

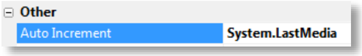
Connection	
driver	DATAFLEX
Filelist	
display name	System
root name	System
Index	
last index number	0
primary index	0
~Misc	
compression	df_file_compress_none
integrity check	True
is system file	True
lock type	df_lock_type_file
max records	1

is system file

(df_file_is_system_file) Indicates whether the table is a

To make sure it actually becomes a system-file, select True for the "is system file" table attribute. Save the table structure before continuing.

We want the Data Dictionaries to automatically increment the ID each time a new row 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.

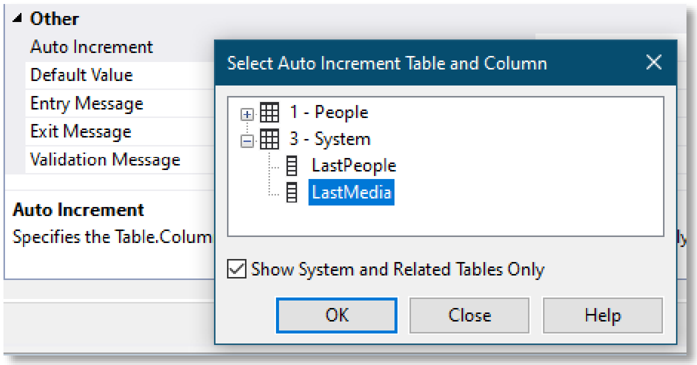


Other

Auto Increment: System.LastMedia

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 rows, with ID's 1, 2, 3 etc. The first time we will use our automated increment function it will try save a row with ID of '1' and that will not work because that value already exists. ID's should always be unique – it is a key field! Our new application cannot save any new rows. We could delete our existing rows, but there is another way to solve this.



Auto Increment

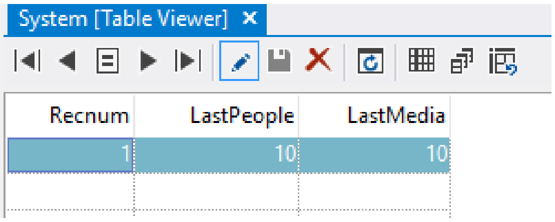
Specifies the Table.Colum

Select Auto Increment Table and Column

☑ Show System and Related Tables Only

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 us assume that the last ID you have entered was 10. Then enter the value 10 for both the LastPeople and LastMedia. Next time a row is created, the IDs will be automatically set to 11.



System [Table Viewer]

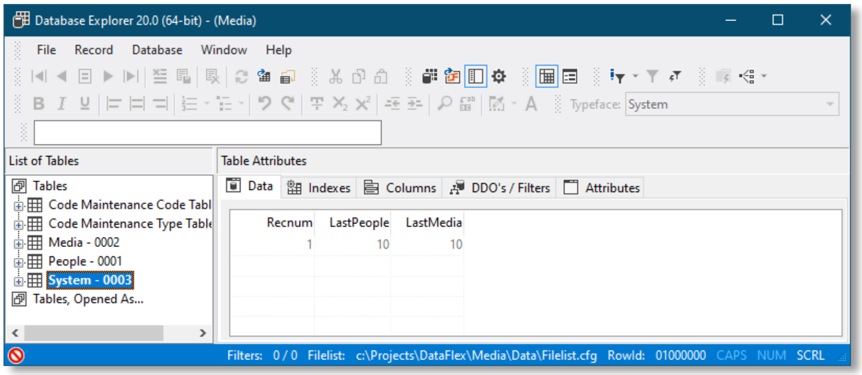
Recnum	LastPeople	LastMedia
1	10	10

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 data to be viewed but not edited. Therefore, click on the little icon at the very bottom-left to change it from a red colored icon (read-only mode) to a gray one (full-access mode).



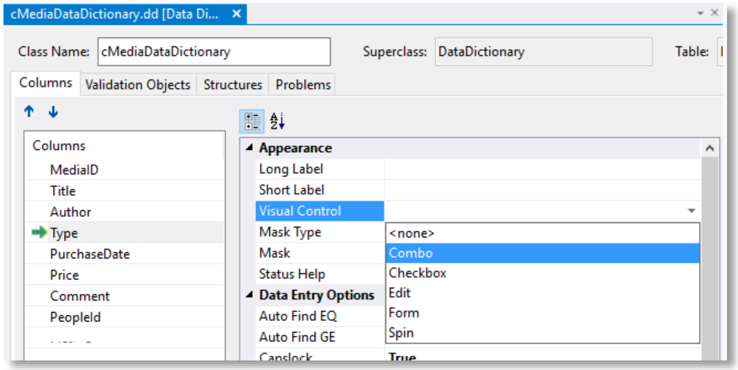
Database Explorer 20.0 (64-bit) - (Media)

Recnum	LastPeople	LastMedia
1	10	10

Filters: 0 / 0 Filelist: c:\Projects\DataFlex\Media\Data\Filelist.cfg RowId: 01000000 CAPS NUM SCRL

Data Dictionary changes

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


cMediaDataDictionary.dd [Data Di...

Class Name: cMediaDataDictionary
Superclass: DataDictionary
Table:

Columns
- MediaID
- Title
- Author
- Type
- PurchaseDate
- Price
- Comment
- PeopleId

Appearance
- Long Label
- Short Label
- Visual Control
- Mask Type
- Mask
- Status Help

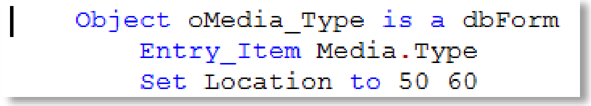
Data Entry Options
- Auto Find EQ
- Auto Find GE
- Capslock: True

- <none>
- Combo
- Checkbox
- Edit
- Form
- Spin

Then, highlight the column MediaID and in the Data Entry Options group, set "Auto Find EQ" to True.

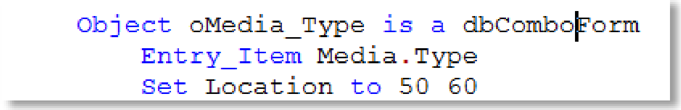
Change the Type Control

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, we need change the control used for the Type column by editing the view. In Code Explorer find the object oMedia_Type. Right-click on the object name in Code Explorer and select "Locate in editor" to have the Studio take you to the source code of the object. The code should look like the following:

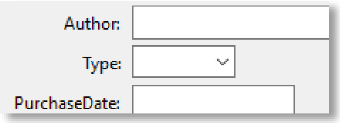


```
Object oMedia_Type is a dbForm
    Entry_Item Media.Type
    Set Location to 50 60
```

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



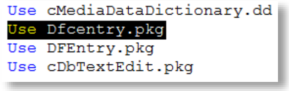
```
Object oMedia_Type is a dbComboForm
    Entry_Item Media.Type
    Set Location to 50 60
```


Author:
Type:
PurchaseDate:

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. Notice that it is now no longer necessary enter to ID's (PeopleID or MediaID) when creating a new row.

We have made the following improvements:

- The ID's for People and Media are automatically serialized/incremented
- Filling in an existing ID at Media and pressing the Tab key immediately after 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 People ID as well
- Entering data for Media Types is easier now: it is always uppercase, and may be selected through 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 will always display this column as a combo box by default



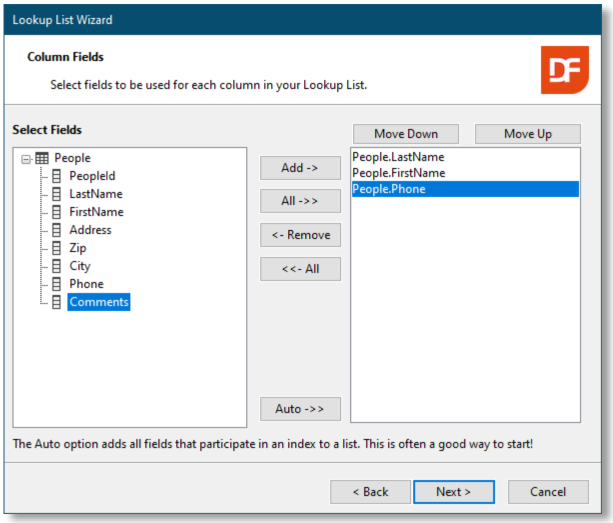
```
Use cMediaDataDictionary.dd
Use Dfcentry.pkg
Use DFEntry.pkg
Use cDbTextEdit.pkg
```

Note: *By changing the type of control manually we may need to add a USE statement to the component file in order to have all the definitions needed in its own source code. That will keep the component file self-contained and autonomous. In our case, the project will compile and run, but there are situations where that is not going to happen. So, to keep our component file autonomous, add "Use Dfcentry.pkg" to the list of USE statements at the top of the component.*

Selection Lists

We can make more improvements: If there are only a few rows in the tables, it is not a problem to find the right data using F7 and F8, but we must offer a better way for finding the right data for when the tables grow. 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 us first make a selection list for People, in which we place LastName, FirstName and Phone (number) on the list.


Lookup List Wizard

Column Fields

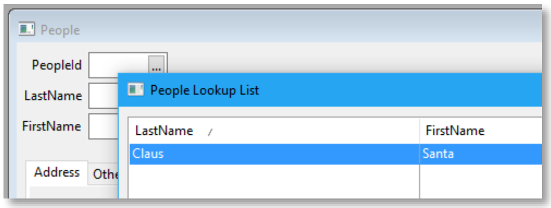
Select fields to be used for each column in your Lookup List.

Select Fields: People, PeopleId, LastName, FirstName, Address, Zip, City, Phone, Comments

People.LastName
People.FirstName
People.Phone

The Auto option adds all fields that participate in an index to a list. This is often a good way to start!

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).


People

PeopleId, LastName, FirstName, Address

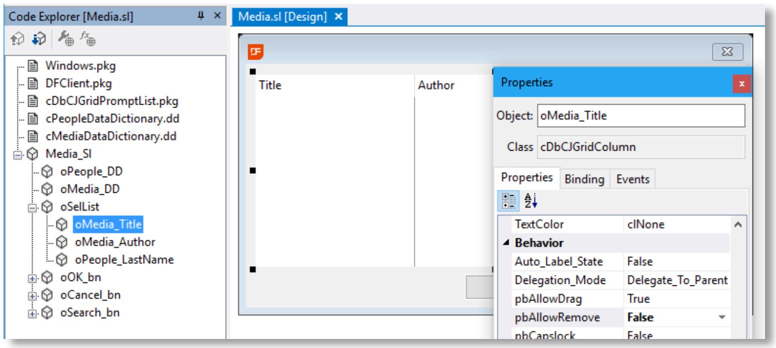
People Lookup List

LastName	FirstName
Claus	Santa

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 search dialog pops up. Pressing Enter will take you to the closest row. Because we use indexes, finding rows in a set of just a few rows will be as fast as when searching a row 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 to use drag & drop instead of using the wizard. In that case, here are some tips: Choose File, New, Dialog / Lookup: Table Lookup. Enter "oMediaLookup" as the object name.


Code Explorer [Media.sl]

Windows.pkg
DFClient.pkg
cDbCJGridPromptList.pkg
cPeopleDataDictionary.dd
cMediaDataDictionary.dd
Media_Sl
oPeople_DD
oMedia_DD
oSelList
oMedia_Title
oMedia_Author
oPeople_LastName
oOK_bn
oCancel_bn
oSearch_bn

Media.sl [Design]

Title | Author

Properties

Object: oMedia_Title

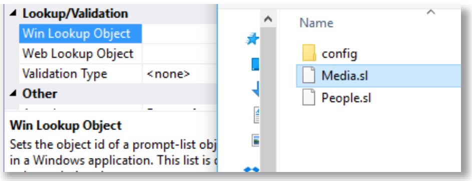
Class: cDbCJGridColumn

TextColor	clNone
Behavior	
Auto_Label_State	False
Delegation_Mode	Delegate_To_Parent
pbAllowDrag	True
pbAllowRemove	False
pbCapslock	False

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 look-up 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.


Lookup/Validation

Win Lookup Object	
Web Lookup Object	
Validation Type	<none>
Other	

Win Lookup Object

Sets the object id of a prompt-list obj in a Windows application. This list is c

Name

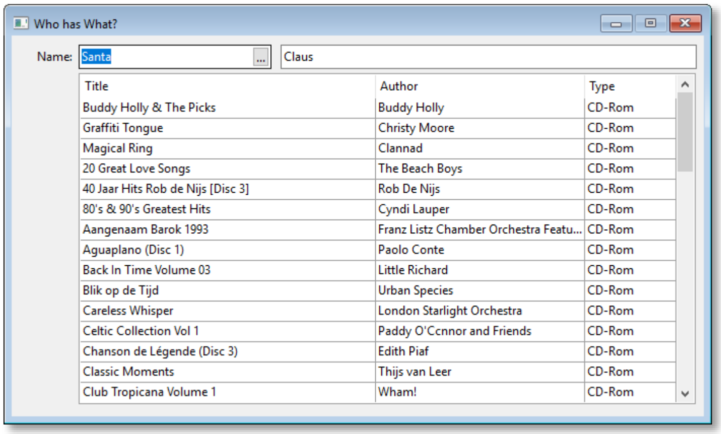
config
Media.sl
People.sl

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.Sl file for the column property Lookup Object.

Show All Media Borrowed

Let us 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, i.e. a filter.



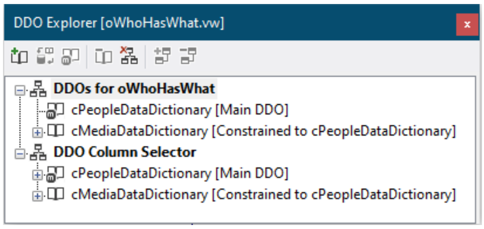
Name: Santa | Claus

Title	Author	Type
Buddy Holly & The Picks	Buddy Holly	CD-Rom
Graffiti Tongue	Christy Moore	CD-Rom
Magical Ring	Clannad	CD-Rom
20 Great Love Songs	The Beach Boys	CD-Rom
40 Jaar Hits Rob de Nijs [Disc 3]	Rob De Nijs	CD-Rom
80's & 90's Greatest Hits	Cyndi Lauper	CD-Rom
Aangenaam Barok 1993	Franz Listz Chamber Orchestra Featu...	CD-Rom
Aguaplano (Disc 1)	Paolo Conte	CD-Rom
Back In Time Volume 03	Little Richard	CD-Rom
Blik op de Tijd	Urban Species	CD-Rom
Careless Whisper	London Starlight Orchestra	CD-Rom
Celtic Collection Vol 1	Paddy O'Ccnnor and Friends	CD-Rom
Chanson de Légende (Disc 3)	Edith Piaf	CD-Rom
Classic Moments	Thijs van Leer	CD-Rom
Club Tropicana Volume 1	Wham!	CD-Rom

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
```



DDO Explorer [oWhoHasWhat.vw]

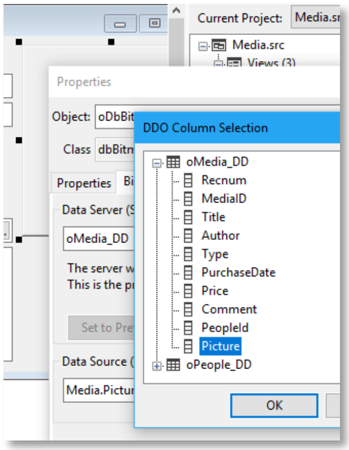
DDOs for oWhoHasWhat
cPeopleDataDictionary [Main DDO]
cMediaDataDictionary [Constrained to cPeopleDataDictionary]
DDO Column Selector
cPeopleDataDictionary [Main DDO]
cMediaDataDictionary [Constrained to cPeopleDataDictionary]

- 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 us say that we want to store a picture with each Media row. First, we need to change the table:

- Open the Media table and add a column named Picture
- Specify the type ASCII, and a length of 255. The actual picture will not be stored here. Each Media row 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)



Current Project: Media.sr
Media.src
Views (3)

Properties

Object: oDbBit

Class dbBitm

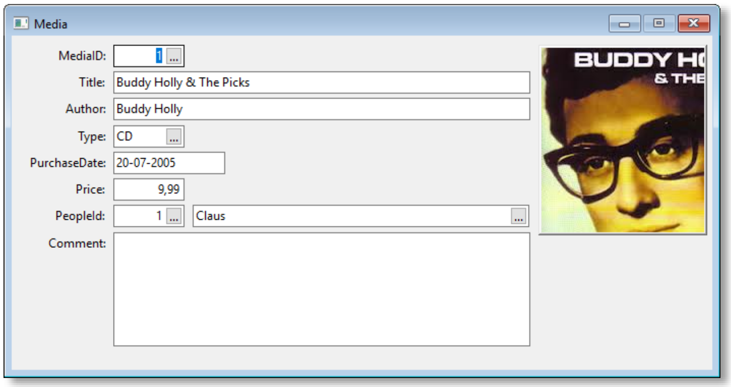
Data Server (S
oMedia_DD
The server w
This is the pr

Data Source (
Media.Pictur

DDO Column Selection

oMedia_DD
Recnum
MediaID
Title
Author
Type
PurchaseDate
Price
Comment
PeopleId
Picture
oPeople_DD

In the application, find a Media row 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 row and make sure you save the data!



MediaID: 1
Title: Buddy Holly & The Picks
Author: Buddy Holly
Type: CD
PurchaseDate: 20-07-2005
Price: 9,99
PeopleId: 1 Claus
Comment:

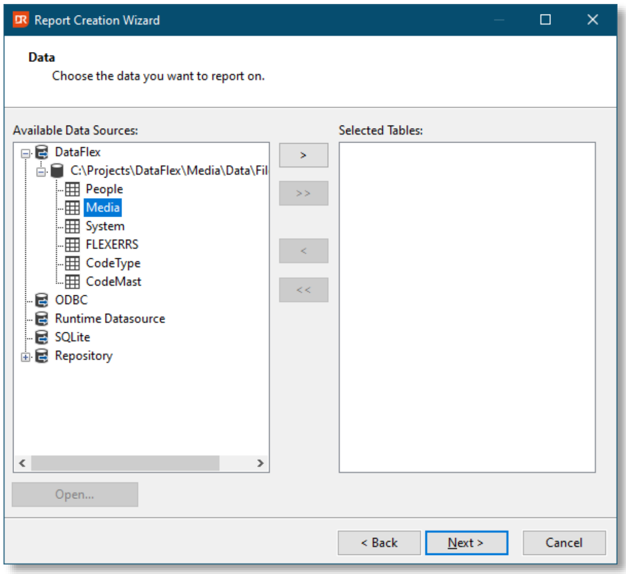
Note: *Out of the box DataFlex "only" supports bitmaps (all images are drawn by Windows as bitmap) but via the free add-on Graphics Library you can add TIFF, JPG, GIF and many other graphic formats. The connection is as easy as described above (we only need to connect to the library and use a different 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 incorporates a report in your Windows application. The end-user will be able to start the report from the menu and still be able to adjust the sort order, output device, and even selection criteria. It is 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.

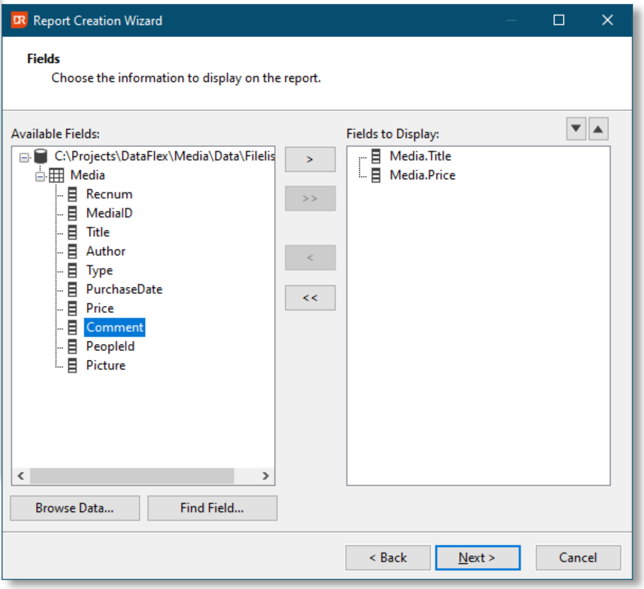


Data
Choose the data you want to report on.

Available Data Sources:
DataFlex
C:\Projects\DataFlex\Media\Data\Fil
People
Media
System
FLEXERRS
CodeType
CodeMast
ODBC
Runtime Datasource
SQLite
Repository

Selected Tables:

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



Fields
Choose the information to display on the report.

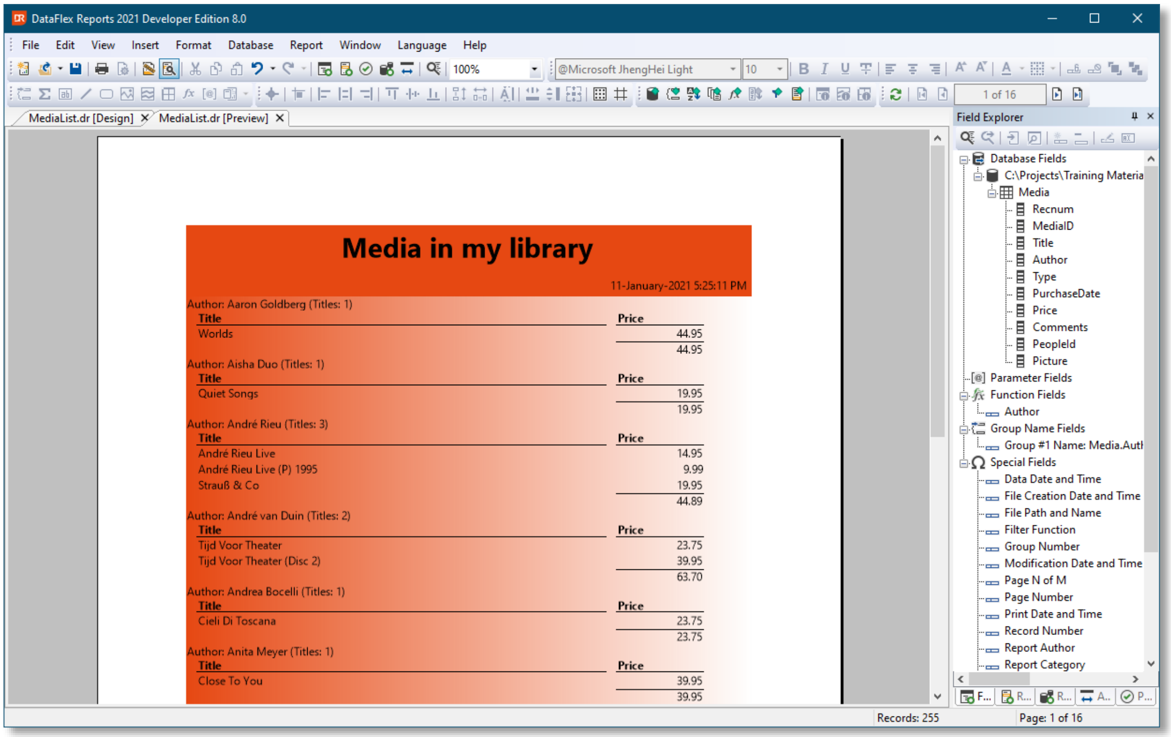
Available Fields:
C:\Projects\DataFlex\Media\Data\Filelis
Media
Recnum
MediaID
Title
Author
Type
PurchaseDate
Price
Comment
PeopleId
Picture

Fields to Display:
Media.Title
Media.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 five 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.



Media in my library

11-January-2021 5:25:11 PM

Author: Aaron Goldberg (Titles: 1)

Title	Price
Worlds	44.95
	44.95

Author: Aisha Duo (Titles: 1)

Title	Price
Quiet Songs	19.95
	19.95

Author: André Rieu (Titles: 3)

Title	Price
André Rieu Live	14.95
André Rieu Live (P) 1995	9.99
Strauß & Co	19.95
	44.89

Author: André van Duin (Titles: 2)

Title	Price
Tijd Voor Theater	23.75
Tijd Voor Theater (Disc 2)	39.95
	63.70

Author: Andrea Bocelli (Titles: 1)

Title	Price
Cieli Di Toscana	23.75
	23.75

Author: Anita Meyer (Titles: 1)

Title	Price
Close To You	39.95
	39.95

- Database Fields
  - C:\Projects\Training Materia
    - Media
      - Recnum
      - MediaID
      - Title
      - Author
      - Type
      - PurchaseDate
      - Price
      - Comments
      - PeopleId
      - Picture
- Parameter Fields
- Function Fields
  - Author
- Group Name Fields
  - Group #1 Name: Media.Autl
- Special Fields
  - Data Date and Time
  - File Creation Date and Time
  - File Path and Name
  - Filter Function
  - Group Number
  - Modification Date and Time
  - Page N of M
  - Page Number
  - Print Date and Time
  - Record Number
  - Report Author
  - Report Category

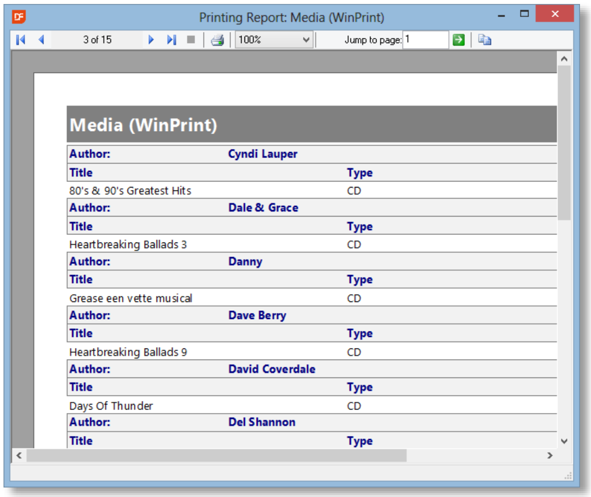
Records: 255

Page: 1 of 16

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.

WinPrint2 Report Wizard

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.



Media (WinPrint)

Author:	**Cyndi Lauper**	
Title		**Type**
80's & 90's Greatest Hits		CD
Author:	**Dale & Grace**	
Title		**Type**
Heartbreaking Ballads 3		CD
Author:	**Danny**	
Title		**Type**
Grease een vette musical		CD
Author:	**Dave Berry**	
Title		**Type**
Heartbreaking Ballads 9		CD
Author:	**David Coverdale**	
Title		**Type**
Days Of Thunder		CD
Author:	**Del Shannon**	
Title		**Type**

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 revenue.

Get Started!

This concludes this Quick Introduction Guide for DataFlex.

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

- https://www.dataaccess.com/
- https://learning.dataaccess.com

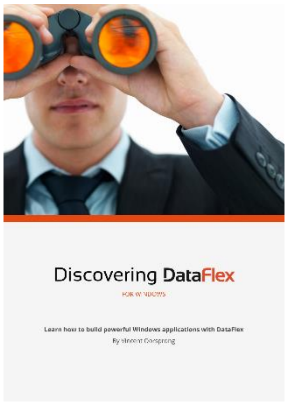
Other related sites:

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

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

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 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 book is available in PDF format.



Discovering DataFlex

FOR WINDOWS

Learn how to build powerful Windows applications with DataFlex

By Vincent Oorsprong