

## はじめに

4D をお試しくださいありがとうございます。

このドキュメントはサンプルデータベースアプリケーション「アドレス帳」を使って、4D の基本的な操作を説明します (所要時間：1 時間程度)。

また、4D v17 で導入された「ORDA」の概念を簡単に紹介するため、同アプリケーション内に「アドレス帳 (ORDA)」という二つ目のユーザーインターフェースも用意しました。ORDA については、ドキュメントの最後で説明しています。

テキストの要所に、オンラインマニュアルへのリンクが埋め込んでありますので、詳細を確認したい場合にご利用ください。

ダウンロードしたサンプルの zip ファイルは、あらかじめ解凍しておいてください。

### 1. 4D の準備

まずは 4D をダウンロード、インストールおよび、お試し版の登録を済ませておきます。

### 2. 4D でサンプルアプリケーションを起動

4D を起動し、初期画面として表示される Welcome ウィザードで「アプリケーションを開始 > ローカルアプリケーションを開く」をクリックします。

(Welcome ウィザードを閉じてしまった場合は、上部メニューから「ヘルプ > Welcome ウィザード」で開けます)

解凍したサンプルアプリケーション (Windows は.4db、Mac は.4dbase) を開きます。

(今回開いたサンプルアプリケーションは次回以降、Welcome ウィザードの「最近使用したアプリケーション」に表示されます)

### 3. サンプルアプリケーションの説明

サンプルアプリケーションを起動すると、「アドレス帳」ユーザーインターフェースが表示されます。サンプルアプリケーションは名前のおり、簡単な住所録です。アドレス帳には 2 件のデータのみ格納されています。1 件のデータのことを「レコード」といいます。

#### ◇ データの閲覧

左のリストボックスで項目を選択すると、その項目がハイライト表示され、右側に当該レコードの詳細が表示されます。「メールする」 ボタンをクリックすると既定のメールソフトで新規メール作成画面が開きます。

レコードを選択している状態だと、左下の「編集」「削除」ボタンが有効になります。

リストボックスの空白行が選択されていると、これらのボタンは無効化されます。

#### ◇ データの検索

上部の検索欄に姓、名、または住所の一部を入力して検索できます。簡略化のため、複数文字列による絞り込み検索には対応していません。

#### ◇ データの編集

リスト項目のダブルクリック、または「編集」ボタンで編集画面に移行します。

#### ◇ データの削除

「削除」ボタンをクリックすると確認ダイアログが表示され、OK ボタンで選択中のレコードが削除されます。

#### ◇ データの作成

「新規作成」ボタンをクリックすると、データの新規登録画面に移行します。新規レコードを入力して OK ボタンをクリックすると、続けて新規登録できます。元の画面に戻るにはキャンセルボタンをクリックします。なお、OK およびキャンセルボタンにはそれぞれ Enter および Esc キーがショートカットとして割り当てられています。

サンプルアプリケーションの動作は以上です。

では、どういう仕組みになっているのか見ていきましょう。

## 4. デザインモード

まず、「アドレス帳」ユーザーインターフェースを閉じます。

(閉じた画面は上部メニューから「ファイル>アドレス帳」で再度開けます)

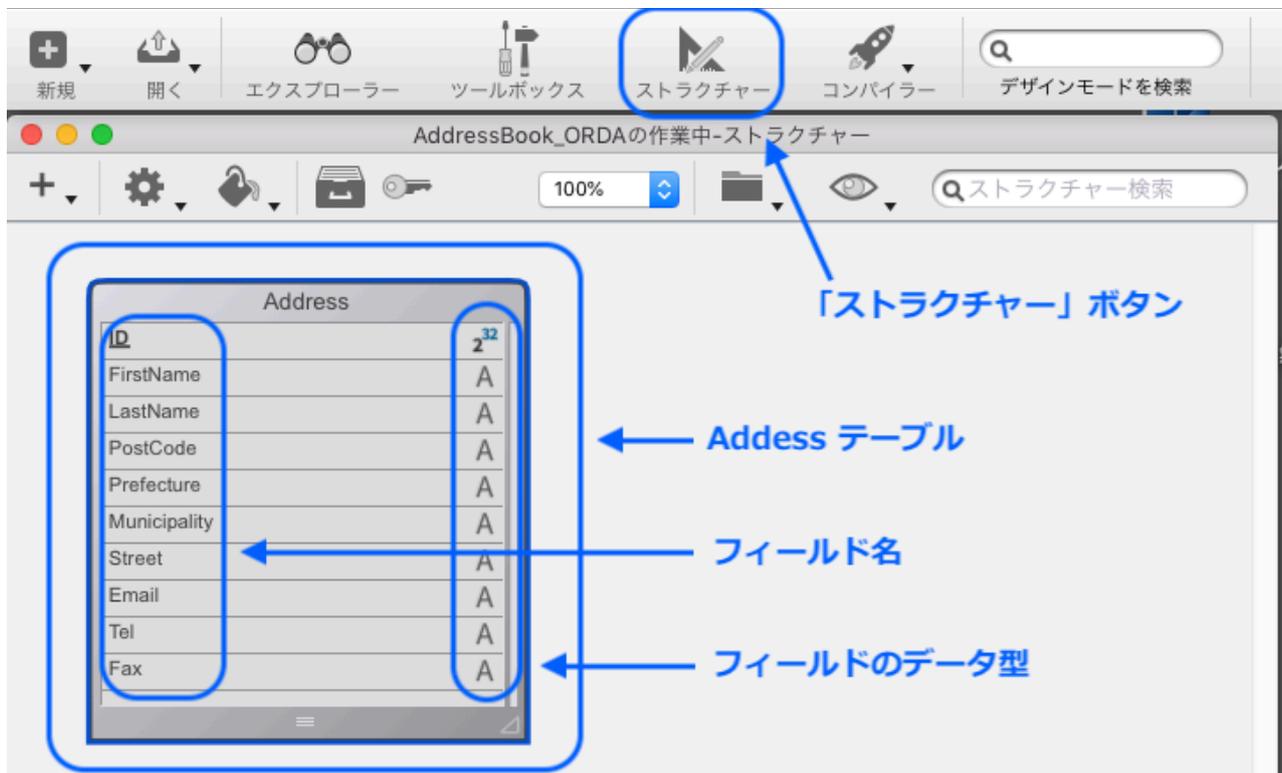
上部メニューから「モード>デザインモードに戻る」を選択します。

デザインモードは、データベースやユーザーインターフェースの設計をするためのモードです。いままでの画面はアプリケーションモードといって、デザインモードで作成したユーザーインターフェースを実際を使ってデータベースを操作するためのモードです。

(なお、アプリケーションモードに戻るには、上部メニューから「実行>アプリケーションモード」を選択します)

## 5. ストラクチャー画面

4D ツールバーの「ストラクチャー」ボタンをクリックします。



ストラクチャーウィンドウが表示されます。ここでは、データベースアプリケーションの骨子となるデータ構造を定義します。アドレス帳にはテーブルがひとつだけ作成されており、住所データはすべてこの[Address]テーブルに格納されています。

テーブルにはデータごとに一つのフィールドが定義されています。例えば、郵便番号を格納するために、PostCode フィールドが作成されています。

(テーブル名とフィールド名には命名規則が存在します)

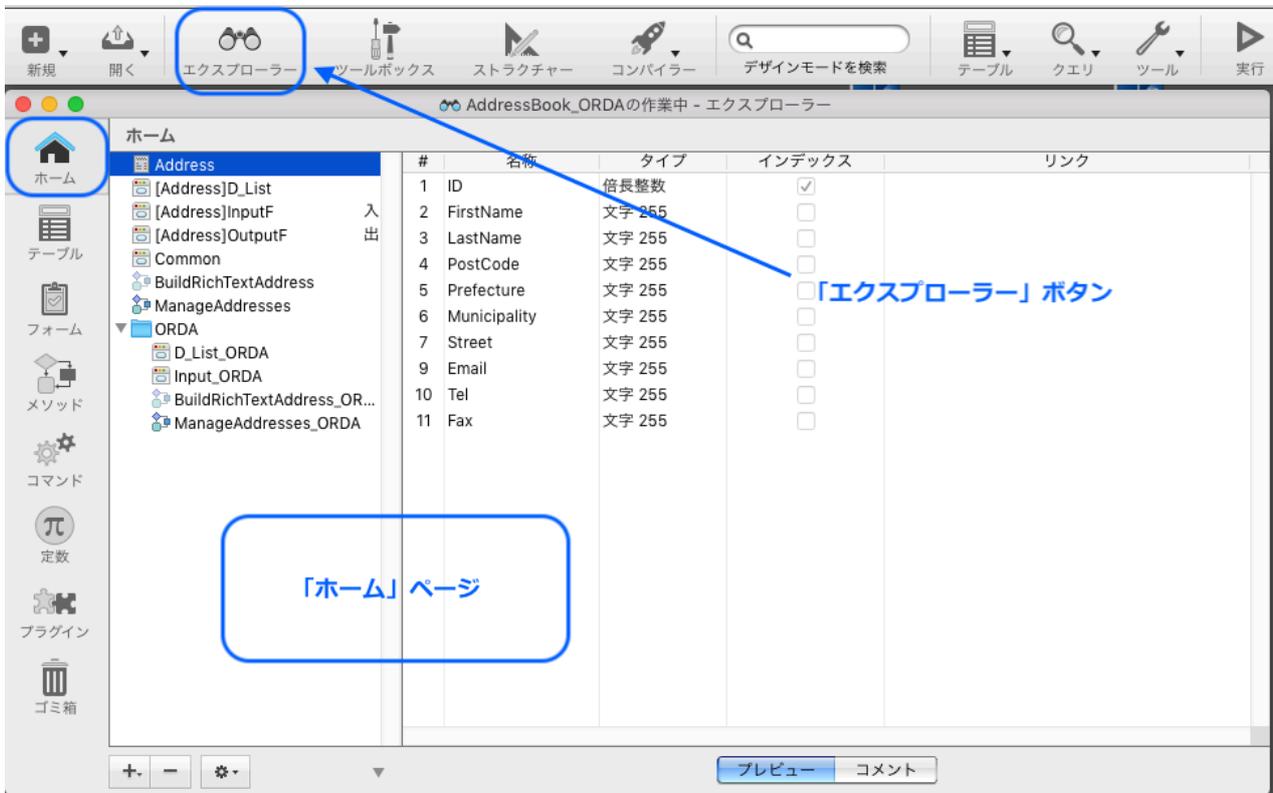
フィールドにはそれぞれデータ型が定義されています。ここでは、ID フィールド以外はすべて文字列型です。

(ID フィールドは、住所データを格納するほかのフィールドとは役割の異なる特別な主キーと呼ばれるフィールドで、一件一件のレコードを区別して管理するために使用されます)

(フィールドをダブルクリックするとインスペクターパレットが開き、各フィールドの詳細設定 (フィールドプロパティ) を確認できます)

## 6. エクスプローラー画面

4D ツールバーの「エクスプローラー」ボタンをクリックします。

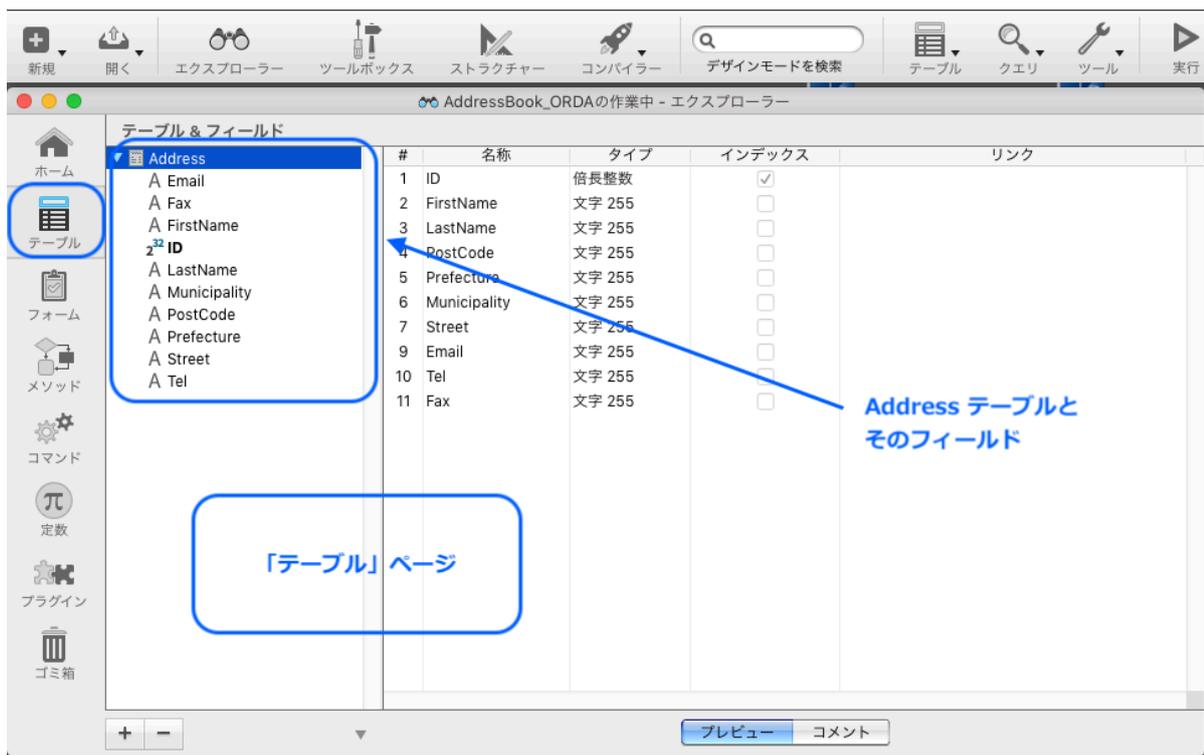


エクスプローラーウィンドウが表示されます。エクスプローラーは、アプリケーションの構成要素の確認を容易にしてくれる管理画面で、おそらく一番よく使うことになるウィンドウでしょう。

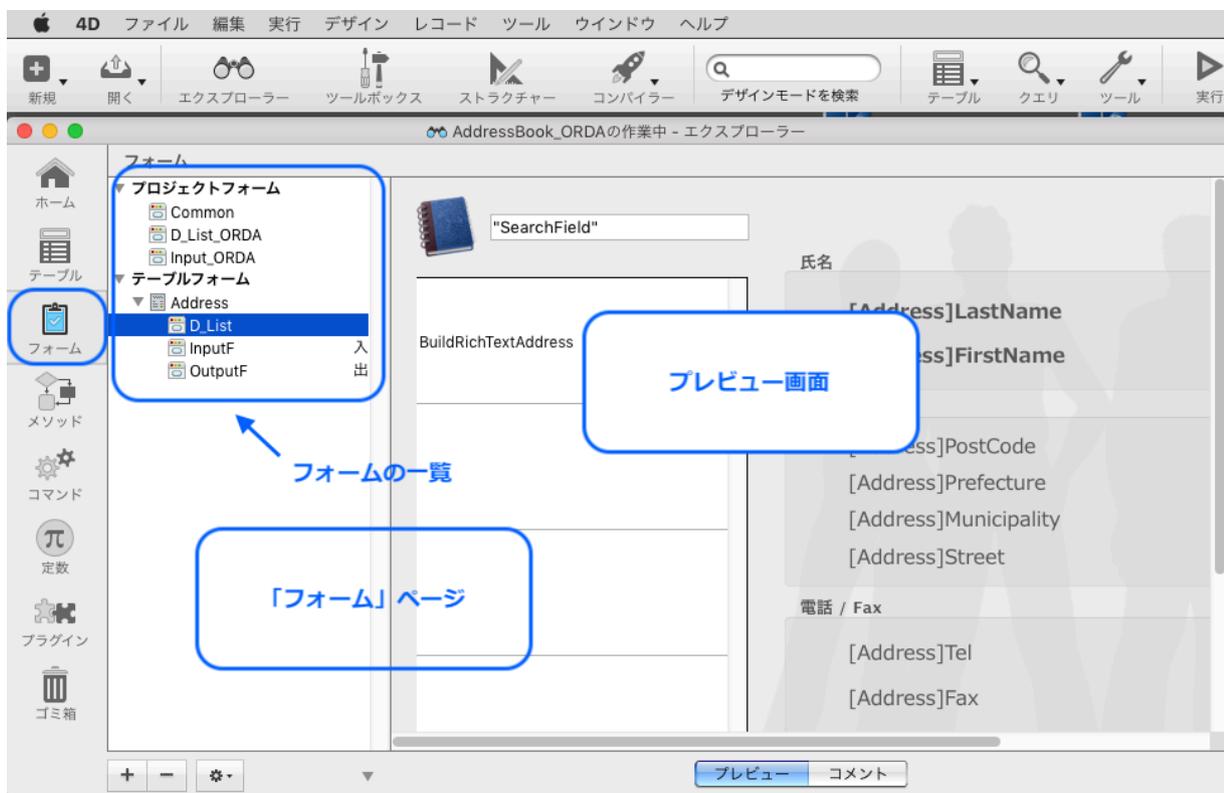
「ホーム」を選択すると、アプリケーションの主な構成要素（テーブル、フォーム、メソッド）が一覧表示されます。各要素を選択すると、それぞれの概要が右にプレビュー表示されます。

「ホーム」ページでは、これらの構成要素を管理しやすいようにフォルダー分けすることができます。サンプルの例では、最後に紹介する「ORDA」関連の要素を別フォルダーに分けました。

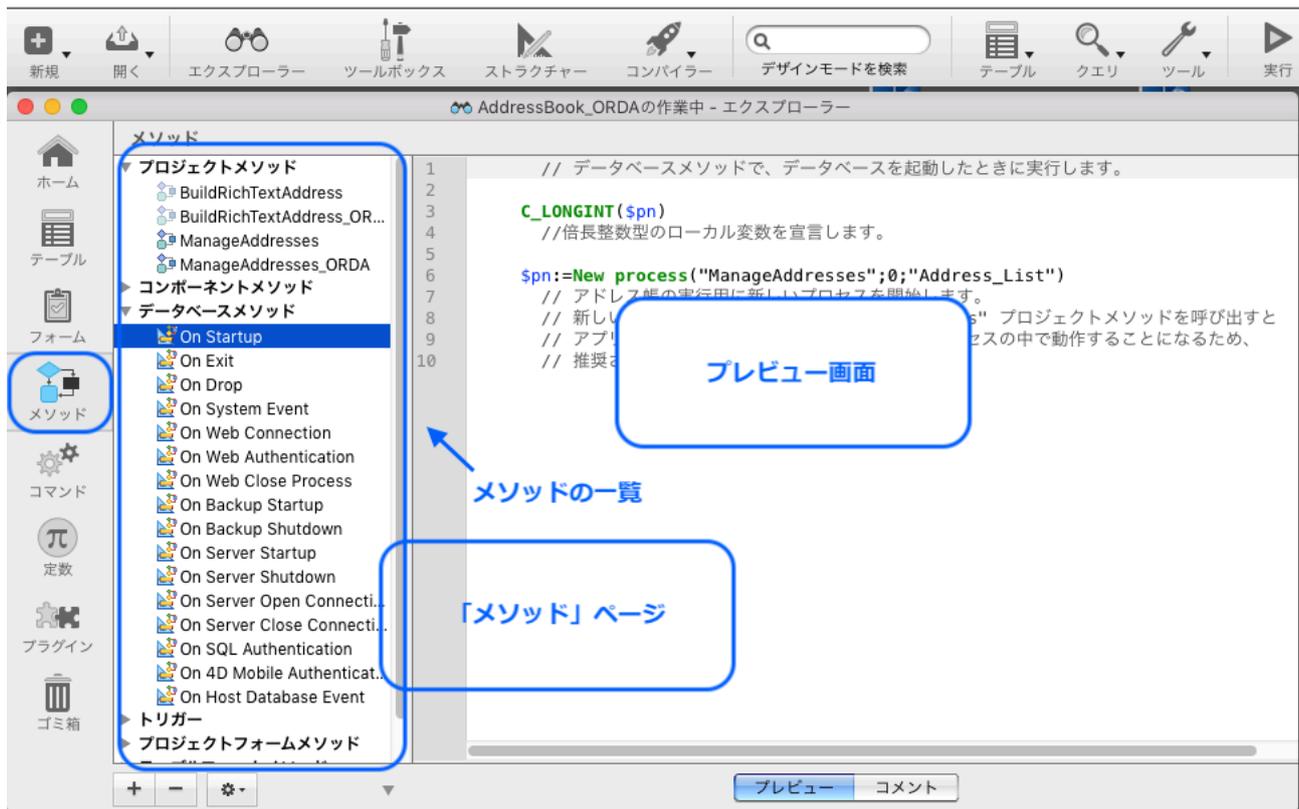
「ホーム」の下にはアプリケーションの各構成要素の専用ページが続きます。



「テーブル」を選択すると、アプリケーション内で定義されているテーブルが一覧表示されます。アドレス帳では、前章で説明した[Address]テーブルのみが定義されています。[Address]テーブルの項目をダブルクリックするとストラクチャー画面が開きます。



「フォーム」を選択すると、アプリケーション内で定義されているフォームが一覧表示されます。このデータベースにはプロジェクトフォームとテーブルフォームが定義されています。任意のフォームをダブルクリックすると、フォームエディターが開きます。



「メソッド」を選択すると、アプリケーション内で定義されているメソッド（プログラムのこと）が一覧表示されます。アドレス帳ではプロジェクトメソッドのほか、データベースメソッド、フォームメソッド、さらにオブジェクトメソッドも使用しています。サンプルでは、コードの合間に説明を入れています。任意のメソッドをダブルクリックすると、メソッドエディターが開きます。

※ オブジェクトメソッドはこのページには表示されません。オブジェクトメソッドを見るにはフォームエディターにて当該オブジェクトを選択し、右クリックよりメソッドを開きます。

エクスプローラーにはそのほかにも、次のページがあります：

- ◇ 「コマンド」：4D ランゲージ (4D で使用するプログラミング言語) で提供されているコマンドが、テーマごとに分類されたリストとして一覧できます。
- ◇ 「定数」：4D で提供されている定数が、テーマごとの分類で一覧できます。
- ◇ 「プラグイン」：4D に最初からついてくるプラグインが表示されています。
- ◇ 「ゴミ箱」：削除した要素が格納されます。これらの要素は回復することができます。

※ 削除する要素によっては、ゴミ箱に格納されず、回復できない場合があります！  
(例：オブジェクトメソッド)

## 7. フォーム画面

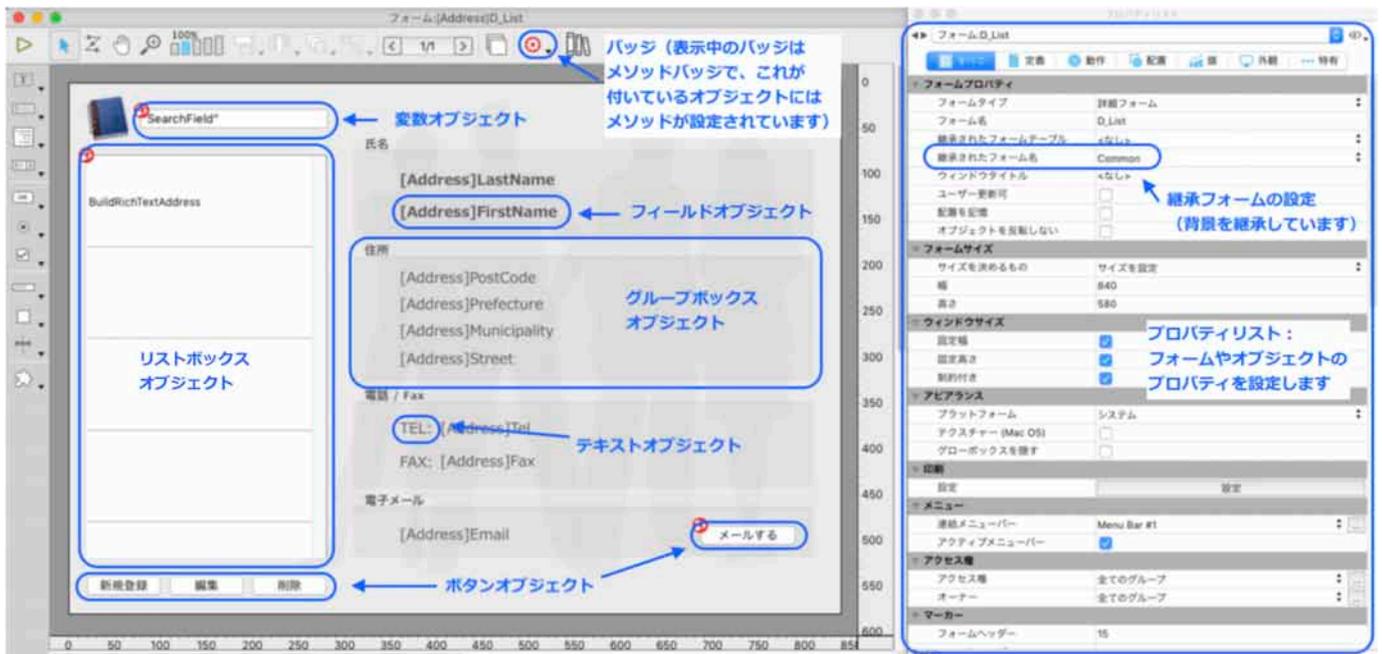
「アドレス帳」では3つあるテーブルフォームのうち2つをインターフェースに使用しています。テーブルフォームは特定のテーブルに属していて、そのテーブルと連動します。ほかにプロジェクトフォームという、テーブルと連動しないフォームが存在します。

まずは、アプリケーション内で使用しないテーブルフォームについて説明します。

**OutputF:** 規定の出力フォームとして[Address]テーブルに設定されているリストフォームです。アプリケーションモードでは使用しませんが、デザインモードにおいて格納データを一覧表示や、検索(「クエリ」といいます)で絞り込むなどして、データ管理に役立ちます。リストフォームは、テーブルのカレントセレクション(「カレント」は「現在の」という意味です)と連動する都合上、テーブルフォームとしてのみ作成することができます。

次に、実際にインターフェースとして使用しているテーブルフォームについて説明します。

**D\_List:** アプリケーションを起動したときに最初に表示される画面です。



"Common"プロジェクトフォームを継承しています(プロパティリストの設定参照)。

フォーム上には住所レコードを一覧表示するリストボックスオブジェクト、選択レコードの詳細を表示する右側のフィールドオブジェクト、それらをまとめるグループボックスオブジェクト、操作に使うボタンオブジェクトなどが配置されています。

各オブジェクトのプロパティは、プロパティリストで確認することができます。データや値を表示するオブジェクトには、変数や式、データソースが設定されています。

リストボックスやフィールドオブジェクトは入力可に設定されていないため、このフォームではデータの閲覧しかできません。

オブジェクトにオブジェクトメソッドが設定されていれば、そのオブジェクトにはメソッドバッジが付きます。バッジは数種類あり、デフォルトで表示されているのがメソッドバッジです。表示するバッジの種類は変更することができます。

例えば、標準アクションバッジに変更すると、リストボックスの下に並ぶ3つのボタンにバッジが付きます。テーブルフォームは特定のテーブルに属しているため、メソッドを設定しなくても、標準アクションを利用してデータの新規作成や編集、削除などを簡易におこなうことができます。

検索エリアには入力可に設定された変数オブジェクトが使われています。ユーザーが入力する検索文字列に応じて、クエリをおこないます。

InputF: 規定の入力フォームとして[Address]テーブルに設定されているフォームです。このフォーム上ではフィールドオブジェクトが入力可に設定されています。OK およびキャンセルボタンには標準アクションが設定されており、クリックすると入力した内容がキャンセル、あるいは保存されます。

"D\_List" および "InputF" フォームは両方とも[Address]テーブルの詳細フォームです。テーブルの詳細フォームは、テーブルのカレントレコードと連動する一方で、カレントセレクションと連動するタイプのオブジェクト(リストボックスなど)も利用することができます。入力や出力に使用する詳細フォームおよびリストフォームは、ウィザードを使って簡単に作成することができます。

Common: このプロジェクトフォームは、"InputF"フォームと"D\_List"フォームに継承されることで、両フォームに共通の背景を提供します。

## 8. メソッド

各メソッドの説明はコメントとしてコードの合間に記述しています。

次のメソッドを確認してみてください：

### ◇ データベースメソッド: On Startup

データベース起動時に自動的に実行されます。アドレス帳のインターフェースを表示させる "ManageAddresses" メソッドを新規プロセスにて実行します。

### ◇ プロジェクトメソッド: ManageAddresses

ユーザーインターフェースとして "D\_List" フォームを表示します。

#### ◇ テーブルフォームメソッド : D\_List

"D\_List" フォームで特定のイベントが発生したときに実行されるコードです。

#### ◇ プロジェクトメソッド : BuildRichTextAddress

"D\_List" フォーム上で "AddressesLB" リストボックスの列のデータソースに、このメソッドが式として設定されています。リストボックス内のレコード毎に式が評価され、戻り値がセルの値となります。(フォームエディターにてリストボックス選択中に列をクリックすると、列要素を選択でき、列のプロパティを見ることができます)

その他にも、以下のフォームオブジェクトメソッドが使用されています(オブジェクトメソッドを確認するにはフォームを開き、メソッドを持つオブジェクトを右クリックして、コンテキストメニューから「オブジェクトメソッド」を選択します):

◇ 「メールする」ボタン : 規定のメールソフトにて、表示中のメールアドレスに宛てたメールを新規作成します。

◇ "AddressesLB" リストボックス : 同リストボックスの行選択をカレントレコードに反映することで、内容が右側に表示されるようにします。

◇ "SearchField" 変数オブジェクト : ユーザー入力に応じたクエリを実行します。

## 9. 手を加えてみましょう

サンプルアプリケーションの説明は以上です。せっかくですので、最後にこのサンプルに少し手を加えてみましょう。

新規レコードを作成するより、既存レコードを複製して使えたら便利な場合があるかもしれませんので、「複製」ボタンを追加します。

1. まず、"D\_List" フォームの編集画面を開きます。
2. 削除ボタンを選択して、上部メニューから「編集>複製」を選択します。
3. 二つ目の「削除ボタン」を一つ目のボタンの右に並べましょう。
4. 複製したボタンのプロパティリストで次の項目を変更します：
  - オブジェクトテーマ : タイトル : 複製
  - アクションテーマ : 標準アクション : アクションなし
  - イベントテーマ : On Clicked : 有効化
5. 変更後、プロパティリストのアクションテーマのメソッド欄にあるボタンをクリックします。すると、「複製」ボタンのオブジェクトメソッド編集画面が開きます。
6. 次のコードをコピーペーストで追加しましょう：  
(手入力の場合は入力中に候補が提案されますので、Tab キーで採用できます)

```
CREATE SET([Address];"tempSet")
DUPLICATE RECORD([Address])
SAVE RECORD([Address])
ADD TO SET([Address];"tempSet")
PUSH RECORD([Address])
USE SET("tempSet")
CLEAR SET("tempSet")
ORDER BY([Address];[Address]LastName;[Address]FirstName)
POP RECORD([Address])
CREATE EMPTY SET([Address];"$HighLightSet")
ADD TO SET([Address];"$HighLightSet")
```

7. メソッドエディターを閉じると、「複製」ボタンにメソッドバッジが追加されたのが確認できます (表示するバッジの種類にメソッドバッジを選択している場合)。
8. 早速、上部メニューから「実行>アプリケーションモード」に入ります。
9. 上部メニューから、「ファイル>アドレス帳」を選択します。
10. いくつかレコードを複製するなどして、実際の動作を確認してみてください。

上記メソッドの説明：

リストボックスに表示されているレコード群は SearchField オブジェクトのメソッドによるクエリ結果のカレントセレクションです。これを後で再利用するために、セレクションを tempSet というレコードのセットに保存します。次にカレントレコードを複製し、これを保存します。この時点で、カレントセレクションは複製レコード 1 件に変更されています。先ほど作成したセットにカレントレコード (=複製レコード) を追加してから、カレントレコードをレコードスタックに一旦よけておきます。tempSet セットを呼び出してカレントセレクションにし、名前を並べ替えます。この時点で、カレントレコードはセット作成時のものに戻ります。レコードスタックによけておいた複製レコードを戻してカレントレコードにします。最後に、リストボックスの選択行をカレントレコードと同期します。

4D の基本的な操作の説明は以上です。最後に、ORDA の説明を少々追加いたします。

## 10. 「ORDA」とは?

ORDA とは、4D v17 より導入された新しいデータ操作の概念で「オブジェクト・リレーショナル・データ・アクセス」(Object Relational Data Access) の略称です。

ORDA では「データストア」と呼ばれる抽象化レイヤーを介することでデータがオブジェクトに変換され、これを操作することができます。具体的には、テーブルはデータクラス、フィールドやリレーションはデータクラスの属性として、レコードはエンティティ、セレ

クシオンはエンティティセレクションの形で提供されます。

ORDAによって、カレントレコード・カレントセレクションの概念にとらわれず、オブジェクト記法を用いたデータ操作が可能になりました。これは、リレーションを多用している場合にとくに本領を発揮しますが、このサンプルデータベースでは初歩的な例としてリレーションを用いないストラクチャーを使用しています。

二つのユーザーインターフェース「アドレス帳」と「アドレス帳 (ORDA)」は、違う方法を採用しながら、ほぼ同じアプリケーションを動作させています。ぜひ、二つを比較してみてください。

※ 動作に多少の違いがありますが、このサンプルでは標準アクションと同一の動作をメソッドで再現するよりも、サンプルの簡素化を優先しました。

## 11. 「アドレス帳 (ORDA)」でも複製ボタンを作ってみましょう

1. まず、"D\_List\_ORDA" フォームの編集画面を開きます。
2. 削除ボタンを選択して、上部メニューから「編集>複製」を選択します。
3. 二つ目の「削除ボタン」を一つ目のボタンの右に並べましょう。
4. 複製したボタンのプロパティリストで次の項目を変更します：
  - オブジェクトテーマ：タイトル：複製
5. 「複製」ボタンを右クリックして、オブジェクトメソッドの編集画面を開きます。
6. すでにあるコードは全部削除し、次のコードをコピーペーストして差し替えましょう：  
(手入力の場合は入力中に候補が提案されますので、Tab キーで採用できます)

```
C_OBJECT($new_entity;$temp_object)
$temp_object:=Form.selected.toObject()
OB REMOVE($temp_object;"ID")
$new_entity:=ds.Address.new      ()
$new_entity.fromObject($temp_object).save()
Form.addresses.add($new_entity)
Form.addresses:=Form.addresses
```

7. メソッドエディターを閉じます。
8. 早速、上部メニューから「実行>アプリケーションモード」に入ります。
9. 上部メニューから、「ファイル>アドレス帳 (ORDA)」を選択します。
10. いくつかレコードを複製するなどして、実際の動作を確認してみてください。

メソッドの説明：

まず、オブジェクト型のローカル変数を宣言します。リストボックスにて選択中のエンティティをオブジェクト化したものを、\$temp\_objectに代入します。このオブジェクトから、あらかじめ ID プロパティを削除しておきます。

次に、\$new\_entity に Address\_ORDA データクラスの新規生成エンティティを代入します。さきほど\$temp\_object に代入したオブジェクトのプロパティと値を、今度は\$new\_entity の属性および属性値に受け渡し、保存します（本来は、保存できなかった場合を考えて処理するのが好ましいですが、ここでは簡略化のため省きます）。保存すると同時に、自動的にあたらしい ID 番号が\$new\_entity に振られます。※まえて ID を削除しておかないと、既存の複製元エンティティへの新しい参照が作成されてしまいます。

リストボックスに表示しているエンティティセレクションに、この新規エンティティを追加します。リストボックスのデータソース Form.address を強制更新します。

このスターターアプリはv17用に作成されています。

v18R5以降で動作させるには一つ編集が必要になります：

ManageAddresses\_ORDAメソッドの24行目

```
$form:=New object("addresses"; ds.Address.all().copy().orderBy("ID asc"))
```

v18R5以降、.all() が返すエンティティセレクションは共有可能なものに変更されました。共有可能なエンティティセレクションは追加不可なため、そのままでは新規エンティティを追加できません。そのため、.copy() を挟んで追加可能なエンティティセレクションに変更する必要があります。