**LIGHTSWITCH DEMO**

SETUP

* ZoomIt
* Visual Studio as Admin
* Warm up VS & SQL
* Metro theme installed

New project -> Lightswitch C#

* talking point here about install, standalone etc…

Name: VGR

Create new table

* SQL Express database is used.
* Can be used or upgraded in production.

Create new table called EmploymentStage

* Talk about ID.
* Add Stage:String:Required

Add screen

List & Details, named: StagesList. Data: Employement Stages

SET AS START PROJECT

F5!

Talk about output Window during build

Talk about the UI

* Very sparse
* Ribbon

Add some values to stages: Approached, Reviewed& Rejected

* Note the pending state - things are done in batches.
* Save

More UI talk

* Talk about filtering
* Talk about searching
* Talk about export to excel

Close. Let’s look at the DB!

* Server explorer:
* Attach DB, switch to SQL Server db file. VirginGalacticRecruitment\VirginGalacticRecruitment\Bin\Data
* Open it up, ooohh at the role stuff. We will get back to that later

Attach to external data source

* Talk about different options here.
* Remember Azure
* Remember different field values that are supported
* Create connection to local space data service. Import the Astronauts, Missions, AstronautMission & Sex tables. Name: Space
* Note the "fixing" of my bad data structure!

FIX the Astronaut table

* Remove display by default on ID
* Fix up display name on Sex1
* Remove display by default for species
* Add species1 column (string, computed). Fix display name & edit method.

*if (!this.Species.HasValue)  
            {  
                result = "Unknown species - CALL MiB";  
                return;  
            }  
   
            switch (this.Species.Value)  
            {  
                case 0:  
                    {  
                        result = "Human";  
                        return;  
                    }  
                case 1:  
                    {  
                        result = "Canine";  
                        return;  
                    }  
                case 2:  
                    {  
                        result = "Chimp";  
                        return;  
                    }  
            }*

Add new screen: AstronautList

* List & detail
* Name: AstronautList
* Data: Astronauts
* Include missions

RUN!

* Check out the master detail in astronaut, note species is correct, note sex is correct!

Hit the design screen option

* + talk about the slight less cool experience on fields, promise more to come
  + Real time is fun
  + Change the Astronaut missions detail to Missions
  + Save – drop to VS
  + Change astronaut details to column layout
  + Change to read only controls
  + Change font style to heading 1
  + Set verticle alignment to stretch and height to 2 lines
  + Add new group > Row layout
  + Drag sex & species there
  + Change sex to summary field
  + Remove add, edit, delete buttons
  + Save
  + Run, Ohh & Ahh
  + Close

Add new table Recruitment

* Remove display by default from ID
* Add relationship to astonauts (CROSS DB RELATIONSHIPS!!!)
  + 1 astro to many recruites
  + Remove display by default from GUID add
* Added InitiatedDate: DateTime NOT REQUIRED
* Add InitiatedBy:String NOT REQUIRED
* Add relationship to Stage
  + 1 stage to many recruites
* Add contact phone field: phone
* Add contact email: email
* Talk about custom types!
* Add HaveCV:Bool NOT REQUIRED – **CHOICE!!**
* Add PaymentWanted:Money – currentcy is ZAR
* Save
* Write Code > Inserting & Updating

*partial void Recruitments\_Inserting(Recruitment entity)  
        {  
            entity.InitiatedDate = DateTime.Now;  
            entity.InitatedBy = this.Application.User.FullName;        }*

* Add new query: SortedRecruitments
  + Sort by date
* Add new grid screen, called: AllRecruitmentsList data is recruitments
  + Click use read-only controls option at top
  + Remove add, edit, delete buttons
* Add new add data screen for recruitment
  + Remove initiaed columsn and stage colum
  + Reorder form
  + Write code saving

*this.RecruitmentProperty.EmploymentStage = this.DataWorkspace.ApplicationData.EmploymentStages.Where(\_ => \_.Stage == "Approached").Single();*

* Run
  + Show the crap autogenerated screen

Now go to astronaut screen and add new screen button:

*this.Application.ShowCreateNewRecruitment()*

* Go to create new recruitments screen and write code
  + Change saved to go to :

*var recruitmentScreen = this.Application.ActiveScreens.Where(\_ => \_.Screen is LightSwitchApplication.AllRecruitmentsGrid).Select(\_ => \_.Screen).SingleOrDefault();  
   
            if (recruitmentScreen != null)  
            {      
                recruitmentScreen.Details.Dispatcher.BeginInvoke(() =>  
                {  
                    recruitmentScreen.Refresh();  
                });  
            }  
            else  
            {  
                Application.ShowAllRecruitmentsGrid();  
            }*

Tour app properties, first general, then extensions, enable metro theme and run again, then type, then access control, then nav

Enable Windows Auth. Add SeeRecruitements role

Go to table and add to the CanRead:

*result = this.Application.User.HasPermission("SeeRecruitments");*

Run show it is still visible, but broken.

Show we can toggle permissions in debug.

Run and show it is working. Show admin permissions briefly.

Walk through a local publish.

Bow