APPEON

The New Horizon for PowerBuilder

Armeen Mazda, Appeon
November 16, 2020

©2020 Appeon Limited and its subsidiaries. All rights reserved.

ELEVATE 2020

PowerBuilder
Defined

Native Windows Apps

PowerScript for UI

```
event open;If Not
IsNull(Message.PowerObjectParm) And
TypeOf(Message.PowerObjectParm) =
TypeOf(istr_parm) Then
    istr_parm = Message.PowerObjectParm
End If

dw_order.SetTransObject(sqlca)
dw_orderitem.SetTransObject(sqlca)

is_CustNo = istr_parm.FCustNo
is_OrderNo = istr_parm.FOrderNo
is_OrgCustNo = is_CustNo

dw_order.Retrieve(is_OrderNo)
dw_orderitem.Retrieve(is_CustNo,is_OrderNo)
end event
```

DataWindow-Centric

Customers' Evolving Needs

Maintain

PowerBuilder
- -Client/Server
- -PowerScript UI
- -PowerScript Logic
- -DataWindows

API-Enable

SnapDevelop
- -Hybrid
- -PowerScript UI
- -C# Logic
- -DataWindows

Cloudify

PowerServer
- -Desktop Cloud App
- -PowerScript UI
- -C# Logic
- -DataWindows

Develop New

PowerClient
- -Desktop Cloud App
- -PowerScript UI
- -C# Logic
- -DataWindows

DataWindows in the Cloud

Demo

PowerServer 2021,
The Missing Piece

	PowerServer 2020	
PB Features	50-65%	~100%
Conversion Effort	Many Months	Days
Design	Blackbox	Open
Security	Rigid	Flexible
Architecture	Traditional Web	Cloud-Native
.NET Framework	.NET (Obsolete)	.NET Core
Server OS	Windows Only	Cross-OS
Client	Browser Plugin	Desktop Cloud App

Runtime Architecture

The Magic Process

Step 1 of 2:
Client Deployment

- PowerScript exe & Byte Code Files
- Application Resource Files
- PowerBuilder VM & Runtime Files

Within the PB IDE, PowerServer:

- Removes ESQLs from PBLs
- Encrypts SQL/PBSelect in DW
- Compiles the application
- Extracts objects from PBDs and encrypt (granular files)
- Signs the application
- Deploys all client files (DLLs, PBVM, etc.) to Web server

The Magic
Process

Step 2 of 2:
Server Generation

Within the PB IDE, PowerServer:

- Converts ESQLs to C# consts
- Converts DataWindows to C# models
- Generates C# project for ESQL & DataWindows
- Generates .NET Core Web API project (app API skeleton)

Truly Automatic
Conversion

Unsupported PowerScript Features

- ReselectRow function of DataWindow (may support in future)
- Database-related code (DW retrieve, SQL, etc.) contained in the RetrieveRow event only of DataWindow

Of course, must compile the C# project and do configuration… such as reconfigure database connection, time out settings, etc.

Open Design,
Control Security

New open design puts you in control of security, but requires knowledge of security and some work to go into production

1. In PowerScript, modify login code to call PowerServer authentication API (~several hours)
2. In C#, enable authentication in startup file (~15 minutes)
3. In C#, replace the default authentication code with your own, such as OAuth 2.0 (~several hours to days to weeks)

Readily Share
Business Logic

1. Partition Business Logic
2. Convert Automatically
3. Polish Generated Code
4. Scaffold Automatically
5. Call REST API from PB
6. Include in C# Project

Benefits
for Users

Internet Access

Automatic Updates

No User Retraining

Benefits
for IT Dept.

Cloud-Native Architecture

Reuse Existing

Negligible Cost

Thank You