



Anwendertage 2019

Elvis Workshop





- Wlan: AWT2019  
Passwort: TPAWT2019
- Netzlaufwerk: ...
- Elvis Version 3.3.289  
→ Installiert .net Framework 4.6.1



1. KNX Secure mit Elvis
2. IoT-Protokoll MQTT mit Anwendungsbeispielen
3. Verschachtelung von Parts
4. CustomItem an Part binden
5. Projektierungsbeispiele: Generic Web
6. Einführung in Xaml am Beispiel CharateristicCurveDisplay
7. Package-Beispiel: Server-Ergänzung Custom Item



## Vorgehensweise:

- ETS
- Elvis



- Export .knxproj
- Export Schlüsselbund/ Keyring-Datei (.knxkeys)



AWTKNX

Details Sicherheit Projektlogbuch Projektdateien

Export

Schlüsselbund exportieren

Gerätezertifikate

+ Hinzufügen | ✗ Löschen

Seriennummer ▲	Fabrikschlüssel (FDSK)	Gerät
000Δ-096F0812	00000000000000000000000000000000Δ096F0812	112 1 Funk-Handsender Δ-ka



ETS-Schlüsseldatei (knxkeys) enthält:

- Gruppenadress-Schlüssel
- Letzte bekannte Geräte-Sequenznummern
- Evtl. Zugangsdaten (Passwort und DAC) einer sicheren Tunneling-Schnittstelle



- Knxproj importieren
  - Passwortabfrage
- Importieren der Keyring-Datei
  - Serverprojekt rechtsklick ->hinzufügen
  - >vorhandenes Element

- Serverkonfiguration Verbindungen  
Passwort für Keyring-Datei angeben



# KNX Secure Elvis



Kommunikation | Speicher | Verbindungen | Sicherheit | Redundanz | Betrieb

Prozessanschlüsse\Darksky  
Prozessanschlüsse\Hue  
Prozessanschlüsse\KNX  
Prozessanschlüsse\Shelly  
Prozessanschlüsse\ShellyBl  
Prozessanschlüsse\Sonoff  
Prozessanschlüsse\System

Aktualisieren

Export Wiederherstellen

**General**

Abfrageintervall beim Hochlauf (ms)	200
Leseabfragen beim Hochfahren puff...	<input type="checkbox"/> False
Lesen-Abstand (ms)	250
Warten auf Lesebestätigung (ms)	500
Warten auf Schreibbestätigung (ms)	0

**KNX**

Falcon-Verbindungszeichenfolge	
GroupAddressFormat	ThreeLevel
IP Port	3671
IP-Adresse	
Keyring-Dateiname	AWTKNX.knxkeys
LocalAddress	
Lokale IP-Adresse	
Min. Sendeabstand (ms)	0
Multicast-Adresse	224.0.23.12
NAT Mode	<input checked="" type="checkbox"/> True
QueryDevicesInterval	0
Verbindungstyp	USB

Keyring password:

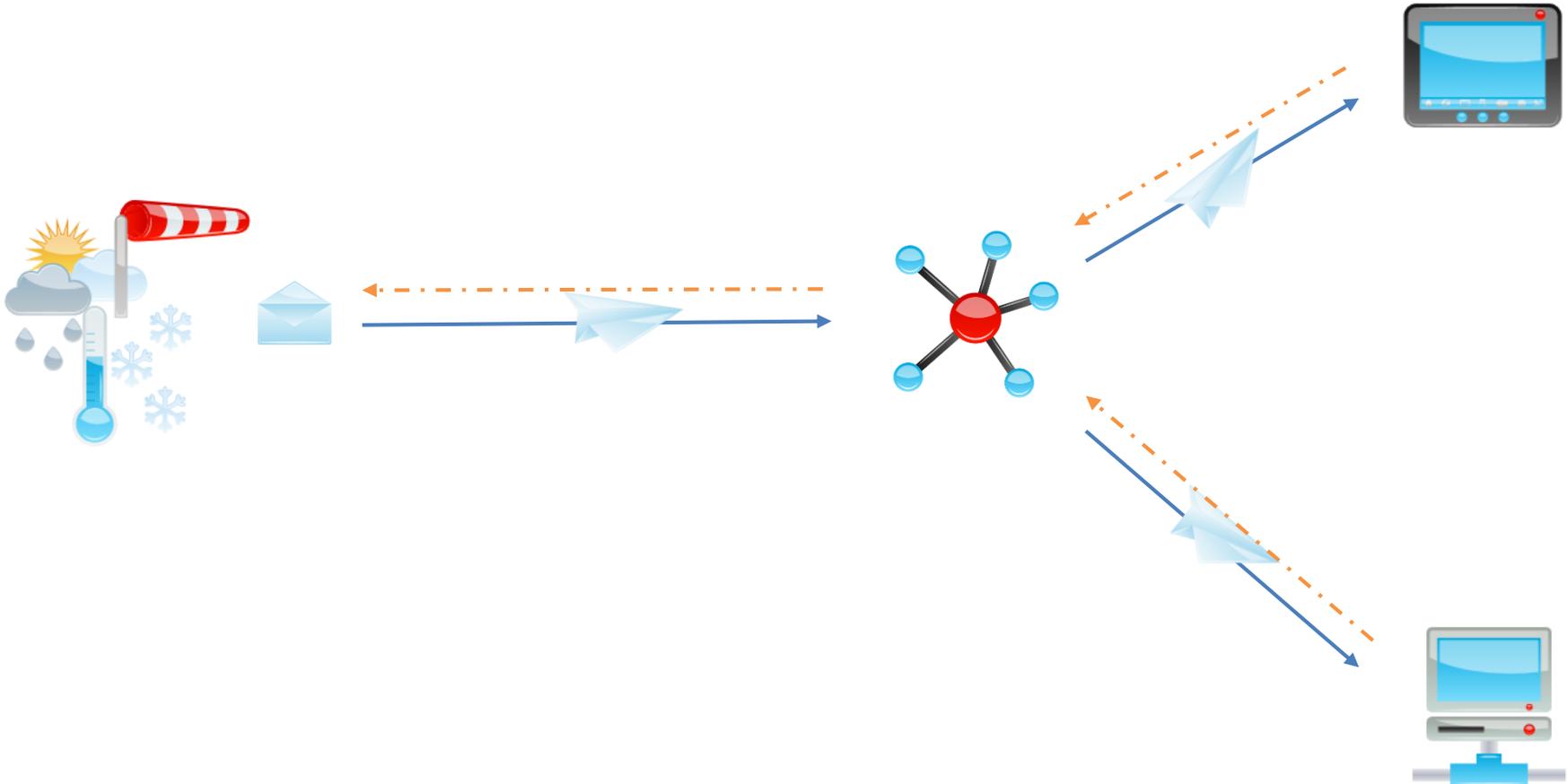


# Demo



- **MQTT** (Message Queuing Telemetry Transport) ist ein offenes Nachrichtenprotokoll für Machine-to-Machine-Kommunikation (M2M), das die Übertragung von Telemetriedaten in Form von Nachrichten zwischen Geräten ermöglicht, trotz hoher Verzögerungen oder beschränkten Netzwerken ermöglicht.
- 1999 Entwickelt von Andy Stanford-Clar von IBM und Arlen Nipper von Cirrus Link Solutions
- Verwendung von MQTT für den Messenger von Facebook
- Seit 2010 als freie Lizenz verfügbar

# MQTT Broker (Server)





- Publish/Subscribe
- Topics and subscriptions
- Quality of service levels
- Retained message
- Clean sessions
- Will message

# Topics

---



- Nachrichten werden auf Topics veröffentlicht
- Aufbau Topic  
z.B.: EG/Wohnzimmer/Sensor/Temperatur
- Ein Client abonniert bestimmte Topics oder benutzt eine „Wildcard“

# Quality of Service QoS

---

- QoS = 0: Nachrichten werden genau einmal versendet ohne Bestätigung
- QoS = 1: Nachricht wird mindestens einmal gesendet.
- QoS = 2: Nachricht wird exakt einmal gesendet.  
→ Zweistufige Empfangsbestätigung





- Shelly IoT-Geräte
  - Anbindung von Liegenschaften
  - Einbindung von 433 MHz Funksensoren
- Broker und Client im Elvis Viewer direct

# IoT Devices & Hue Bridge

---



- Shelly1: 192.168.104
- ShellyBulb: 192.168.105
- Sonoff: 192.168.2.100
- Hue-Bridge: 192.168.2.101  
→ Einbinden über Supportanwendung  
(Datenpunkt: Light3)

# Demo



Part in Part in Part in...

- Viele Steuerelemente auf einer Bedienseite
  - weniger Bindungen
  - Parts werden erst beim öffnen geladen
  - bessere Performance

# Verschachtelung von Parts



*Auriga*  
HOTEL & SPA

Hotelgebäude    Appartementhaus    Villa de Lux    ZENTRAL FUNKTIONEN

Notlichtanlagen  
Aussenbel.  
Aussenheizungen  
Dachrinnenheizung  
Wassertechnik  
Speisereste  
Alarm

DALI Steuergeräte ok

EVG    LED  
Treiber

1  
2  
3  
4  
5  
6  
7  
8





# Demo



- Erstellen eines CustomItems
- Bindung der Ein- und Ausgänge an einen Part
- CustomItem an PrototypeBinding



- Darksky  
→ Json

```
https://api.darksky.net/forecast/8 x +
https://api.darksky.net/forecast/8afcc8c4bce74f4ba295
{"latitude":49.5,"longitude":11.1,"timezone":"Europe/Berlin","curr
day","precipIntensity":0.0002,"precipProbability":0.09,"precipType
ility":6.31,"ozone":348.56},"hourly":{"summary":"Den ganzen Tag la
day","precipIntensity":0.0002,"precipProbability":0.1,"precipType'
ty":6.21,"ozone":349},{ "time":1553601600,"summary":"Überwiegend be
day","precipIntensity":0,"precipProbability":0,"temperature":44.14
{"time":1553605200,"summary":"Überwiegend bewölkt","icon":"partly-
day","precipIntensity":0,"precipProbability":0,"temperature":44.71
{"time":1553608800,"summary":"Überwiegend bewölkt","icon":"partly-
day","precipIntensity":0,"precipProbability":0,"temperature":45.56
{"time":1553612400,"summary":"Stark
bewölkt","icon":"cloudy","precipIntensity":0,"precipProbability":0
0,"ozone":338.7},{ "time":1553616000,"summary":"Stark
```

```
1
2 "latitude": 49.5,
3 "longitude": 11.1,
4 "timezone": "Europe/Berlin",
5 "currently": {
6   "time": 1553598369,
7   "summary": "Überwiegend bewölkt",
8   "icon": "partly-cloudy-day",
9   "precipIntensity": 0.0002,
10  "precipProbability": 0.09,
11  "precipType": "rain",
12  "temperature": 43.28,
13  "apparentTemperature": 37.24,
14  "dewPoint": 31.73,
15  "humidity": 0.63,
16  "pressure": 1029.47,
17  "windSpeed": 11.25,
18  "windGust": 12.15,
19  "windBearing": 287,
20  "cloudCover": 0.87,
21  "uvIndex": 3,
22  "visibility": 6.31,
23  "ozone": 348.56
24 },
25 "hourly": {
26   "summary": "Den ganzen Tag lang überwiegend bewölkt.",
27   "icon": "partly-cloudy-night",
28   "data": [
29     {
30       "time": 1553598000,
31       "summary": "Überwiegend bewölkt",
32       "icon": "partly-cloudy-day",
```

→ z.B. in Notepad kopieren

→ Erweiterungen Json Format



- **ReadRequestURL**='https://api.darksky.net/forecast/{P}/49.5,11.1?lang=de&units=ca&exclude=minutely,hourly,alerts,flags'
- **ContentType**='Json'
- **ReadRequestMethod**="GET"
- **ReadResponseExtractor**="\$.{A}" \$. Root object/element
- **ReadResponseValueExtractor**="{V}"
- **CacheLevel**='Default'





# Demo



- Properties
  - explizit
  - implizit
  
- ResourceDictionary
  - lokal (Page)
  - Projekt
  - Global (ClientLib)



- Values oder Objekte in Resources speichern
- Expliziter Style
- Impliziter Style
  - lokal/ global
  - Page, App.xaml, ControlResources.xaml

# XAML Properties



Zwei Wege Properties für Elemente anzugeben

```
<Label ....TextColor="Blue".. />
```

```
<Label ... >
```

```
    <Label.TextColor>
```

```
        Blue
```

```
    </Label.TextColor>
```

```
</Label>
```

# XAML Properties



```
<Label Content="Label" Canvas.Left="585" Canvas.Top="591"  
Width="135" Height="211">
```

```
<Label.Background>
```

```
<LinearGradientBrush EndPoint="1,1" StartPoint="0,1">
```

```
<GradientStop Color="Green" Offset="1"/>
```

```
<GradientStop Color="Blue" Offset="0.011"/>
```

```
</LinearGradientBrush>
```

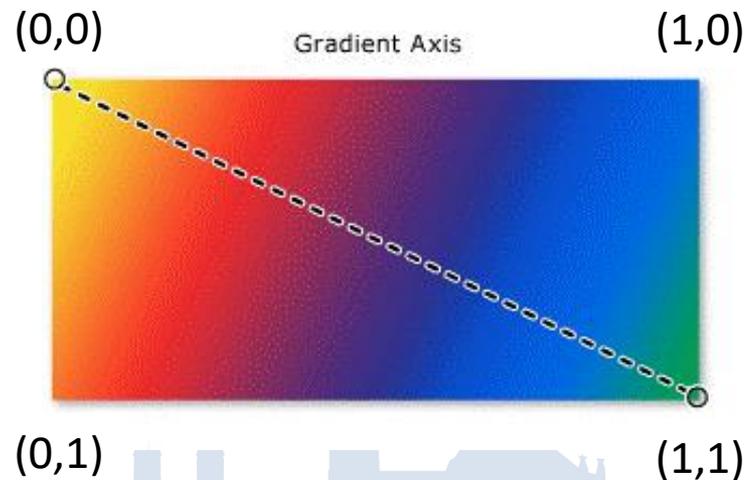
```
</Label.Background>
```

```
</Label>
```

# XAML LinearGradientBrush



- Default linear gradient diagonal  
→ in Elvis Horizontal
  - StartPoint = (0.5, 0)
  - EndPoint = (0.5, 1)





- Lokales und globales Festlegen von Properties

Beispiel:

```
<SolidColorBrush x:Key="Color" Color="Violet"/>
```

```
<Style TargetType="Label" x:Key="LabelStyle">
```

```
  <Setter Property="FontSize" Value="32"/>
```

```
  <Setter Property="Foreground" Value="Red"/>
```

```
</Style>
```

# Setter Property Background



```
<Setter Property="Background">
```

```
  <Setter.Value>
```

```
    <LinearGradientBrush EndPoint="0,1"  
      StartPoint="0,0">
```

```
      <GradientStop Color="Green" Offset="1"/>
```

```
      <GradientStop Color="Blue" Offset="0.011"/>
```

```
    </LinearGradientBrush>
```

```
  </Setter.Value>
```

```
</Setter>
```

**→ Expliziter Style**

```
<Label ... Style="{StaticResource LabelStyle}" />
```

# Impliziter Style

---



```
<Style TargetType="Label">
```

...

```
</Style>
```

→ Gültig für alle Labels



Verwendung nur im Teilprojekt

→ Page

→ App.xaml

Global ClientLib

→ ControlResource.xaml



- Microsoft Expression Design 4  
(Kostenlos erhältlich)

→ XAML Designer

→ Xaml kann direkt in Elvis kopiert werden



# Demo



- AxisItem - Definition der X- und Y-Achsen
- TextItem - Definition einer Textanmerkung.
- PointItem - Definition eines Punktes .
- LineItem - Definition einer Linie.
- SeriesItem - Definition eines Polygons.
- SensorPoint - Definition eines Punktes, der den Wert eines Sensors als Farbe repräsentiert.
- ArrowItem - Definiert einen Pfeil.
- ArrowGroup - Definiert eine Gruppe von Pfeilen.



- Lineltem → ShowIntersectionsWith
- CurveCollection  
→ CurvePoints als Pfad für arrows
- Positionierung nach angelegter Reihenfolge
- ElvisPage.Resources

---

# Demo