PowerBuilder User Group Germany - Meeting

Inter-App Communication for Desktop, Web or Mobile Apps

Heino Hellmers / S&F Datentechnik
November 13-14, 2017 – Berlin (Germany)

DISCLAIMER

Author Profile

Heino Hellmers

Name	Heino Hellmers, Dipl.-Informatiker
Place of residence	Leer, Germany
Position	CEO, S&F Datentechnik
E-Mail	hellmers@sf-datentechnik.de

Author Profile

Heino Hellmers

twitter.com/HeinoHellmers

linkedin.com/in/HeinoHellmers/

Key Skills

- PowerBuilder
- SQL Server / Oracle
- Software Development (Desktop, Web, Mobile)
- Appeon Web/Mobile (PowerServer Web/Mobile)
- Visual Studio
- Project Management and Consulting

Recent Projects

- Migrations of Windows Mobile/CE based Apps to Android/iOS using Appeon Mobile and developing new mobile Apps.
- Migration of PB WebForm based Apps to Appeon Web and bringing native PB Windows Desktop App to the Web.
- Maintenance and new Development of different Desktop Apps.
- Moving Desktop Apps to Appeon PowerBuilder from older PB Version.

Agenda

- Company Profile „S&F Datentechnik“
- Initial Situation and Motivation(to use Inter-App Communication)
- Inter-App Communication by using URI Schemes (Basics)
- Calling (custom) URI Schemes
- Creating custom URI Schemes
- Demo (Integration & Application)
- Conclusion

Company Profile

APPEON Consulting Partner

Microsoft Partner
Gold Application Development

ORACLE Gold Partner

S&F DATENTECHNIK
SOFTWARE ▪ CONSULTING

About

S&F Datentechnik is an independent consulting and software development company in Germany with a team of around 45 people (15 PowerBuilder developers).

The company was founded in 1984.

The company began the development and consulting with PowerBuilder in 1996 and has since then implemented various projects based on PowerBuilder.

Experience / Installations / Services

- Software Development for Desktop-, Web- and Mobile- Apps based on Appeon products
- Project Management, Consulting, Customer Service and Sales
- Over 30 years of service provider for Public Administration (product line KOMVOR) and the waste management industry (product line EMOS)
- More than 400 installations in Europe

Initial Situation and Motivation (to use Inter-App Communication)

Inital Situation

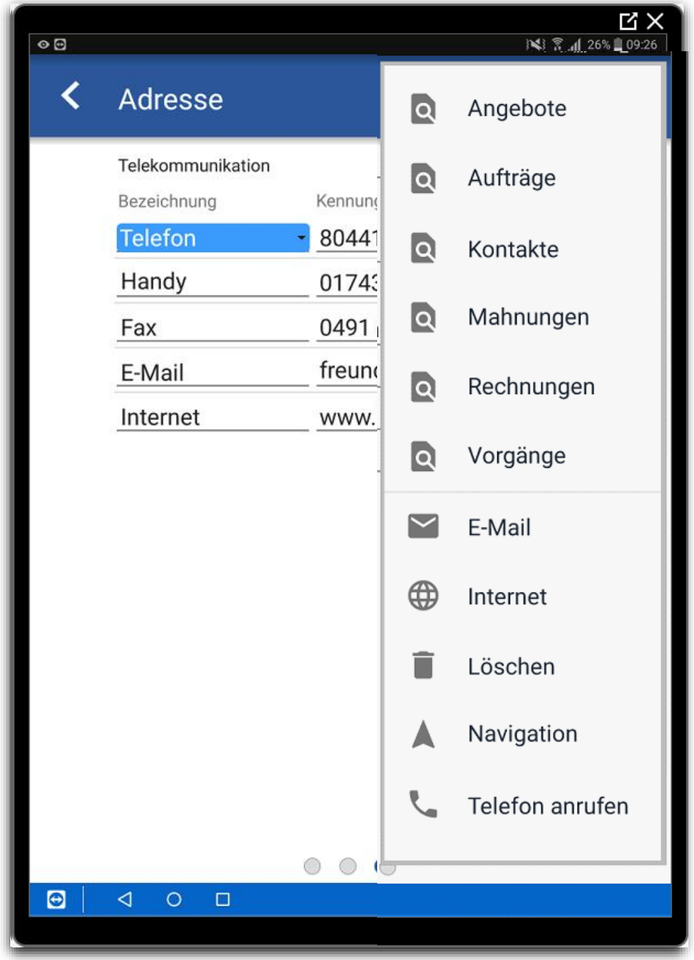
- Various applications with additional information and different functionalities are used in the daily work.
- It is not possible and desired to bundle the functionalities of all applications into one application.
- It is not possible to discuss everything directly.
- documenting the correct records and facts costs a lot of time.

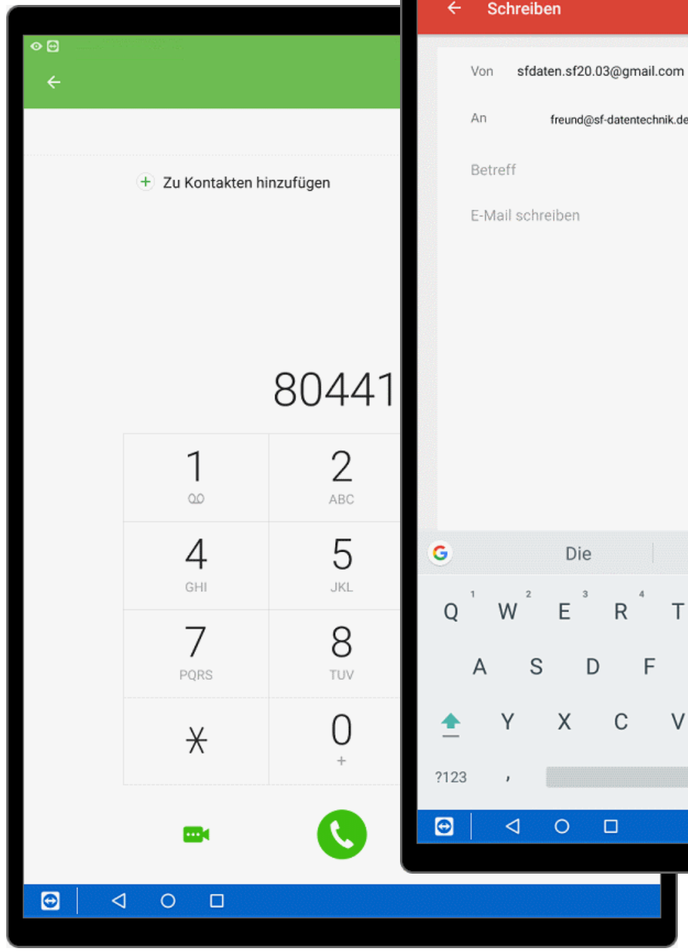
Initial Situation and Motivation (to use Inter-App Communication)

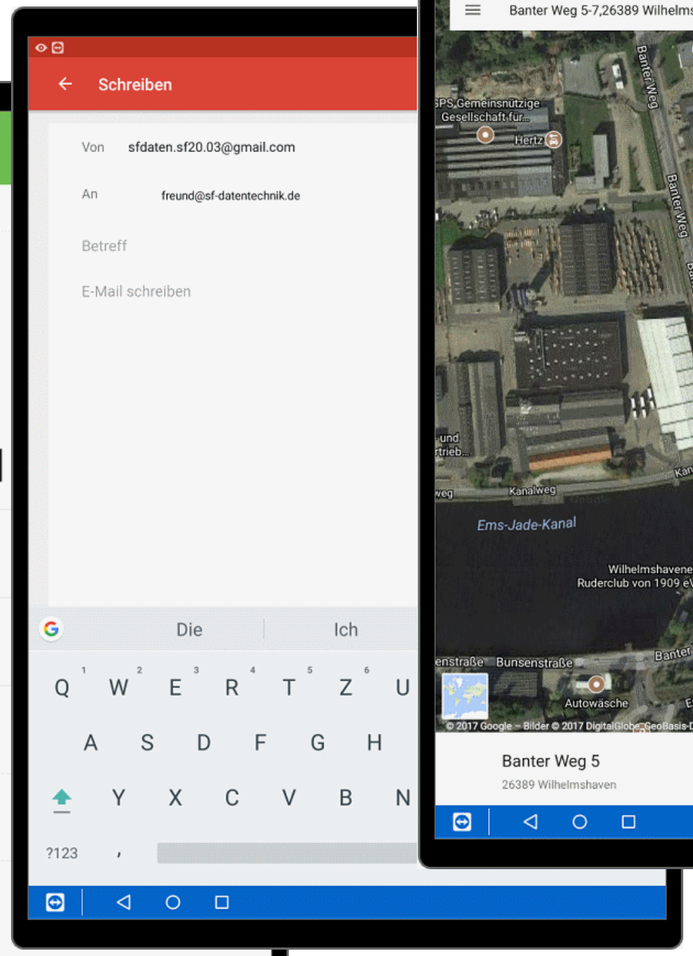
Motivation (Requests)

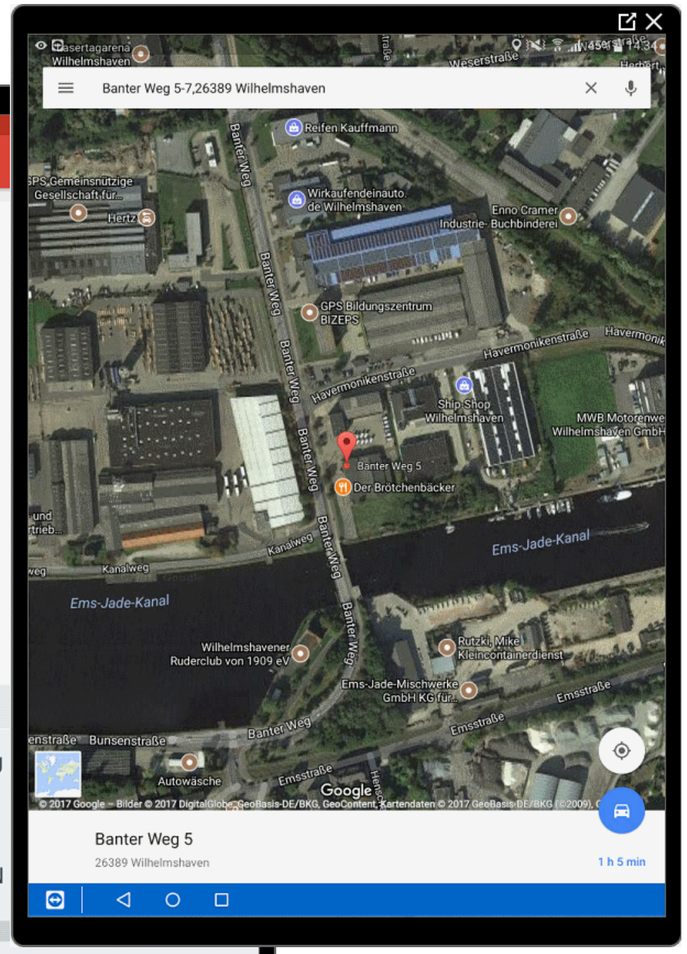
- The wish exists to branch (directly or time delayed) from an application to another application (cross platform) to be able to view additional data and, if necessary, to directly capture or update the data.
- Users want to exchange information quickly and easily with colleagues and they want to use specific references to records by e-mail or other documents.
- They want to extend the possibilities of a single App by using other Apps and their functions and integrate them by using Inter-App Communication.

Appeon Mobile App calling URL Schemes









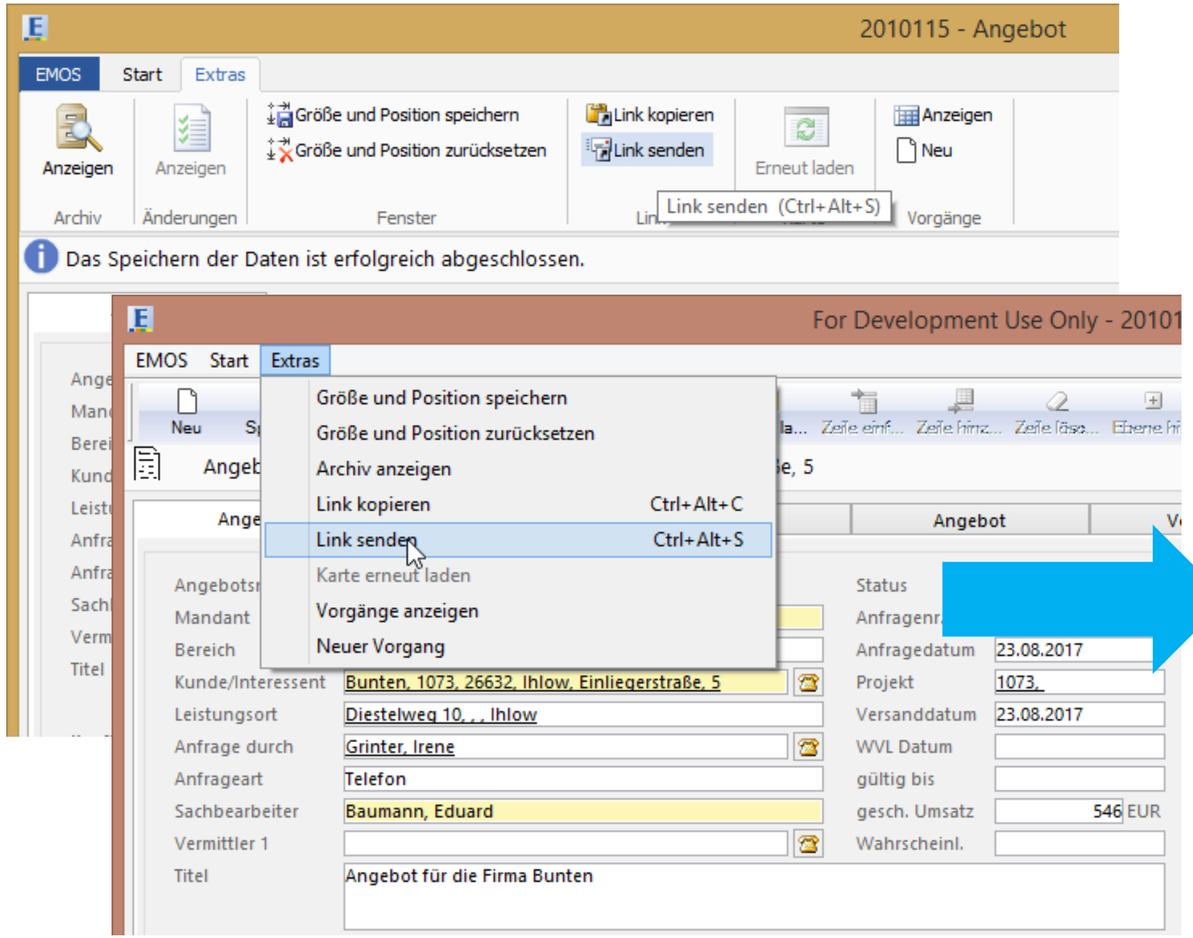
to call

to mail

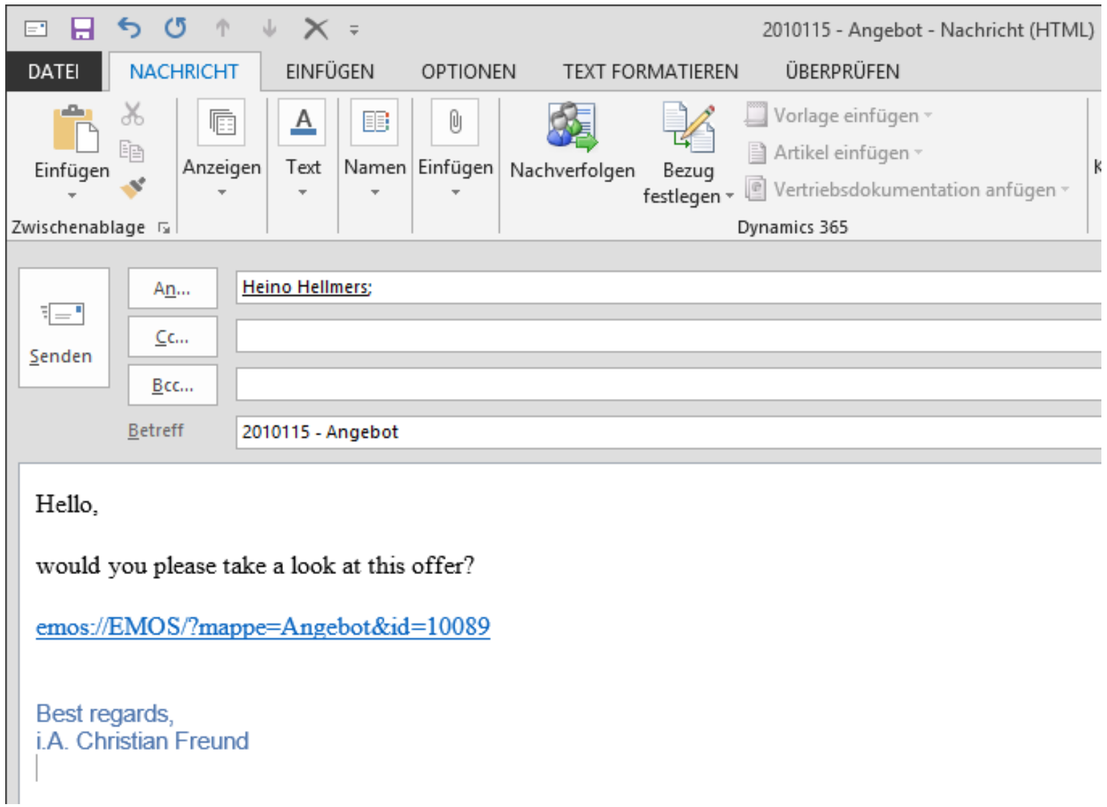
to navigate

PB App sending URL by mail

Desktop App

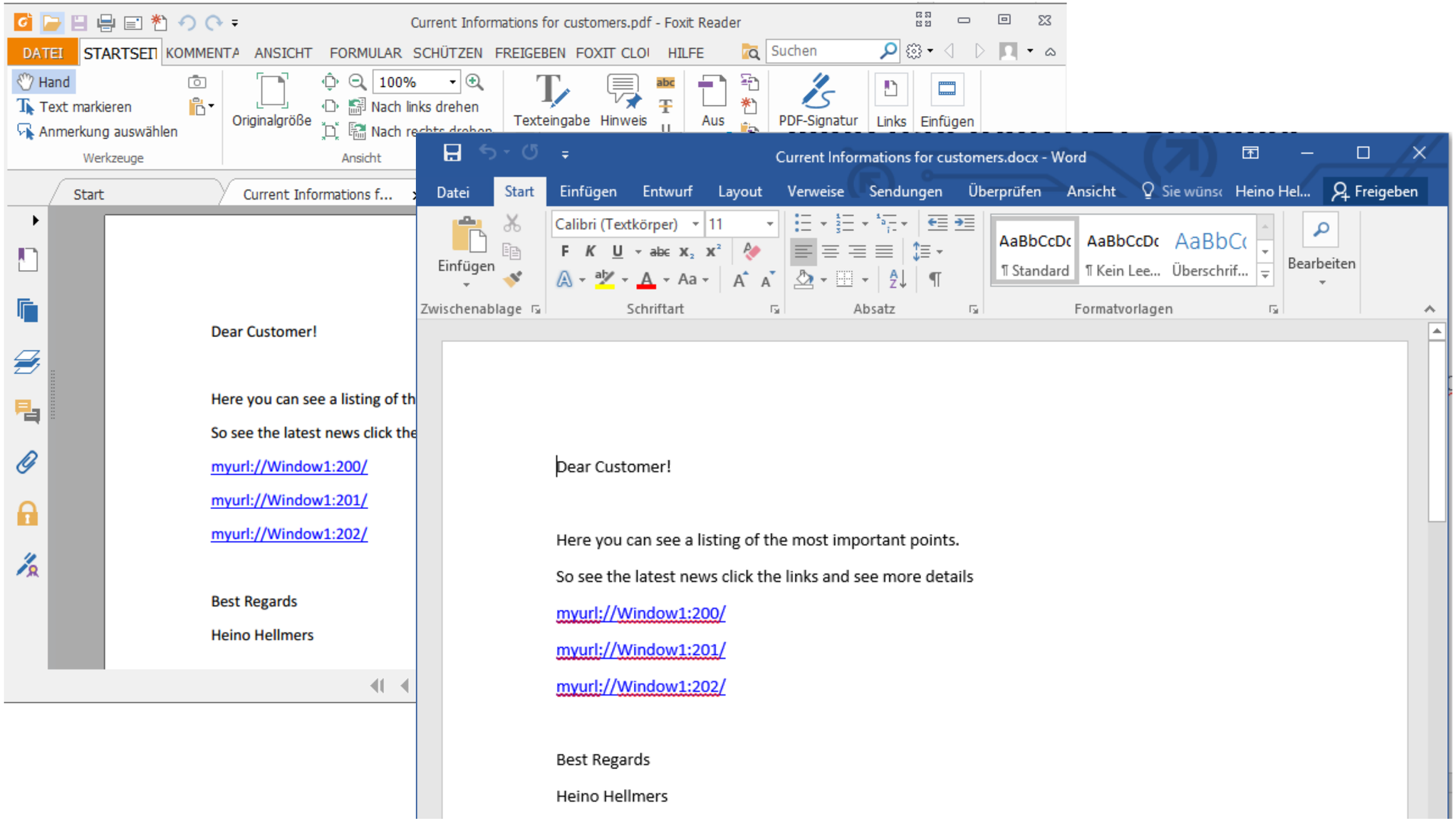


Web App



MS Outlook with clickable link

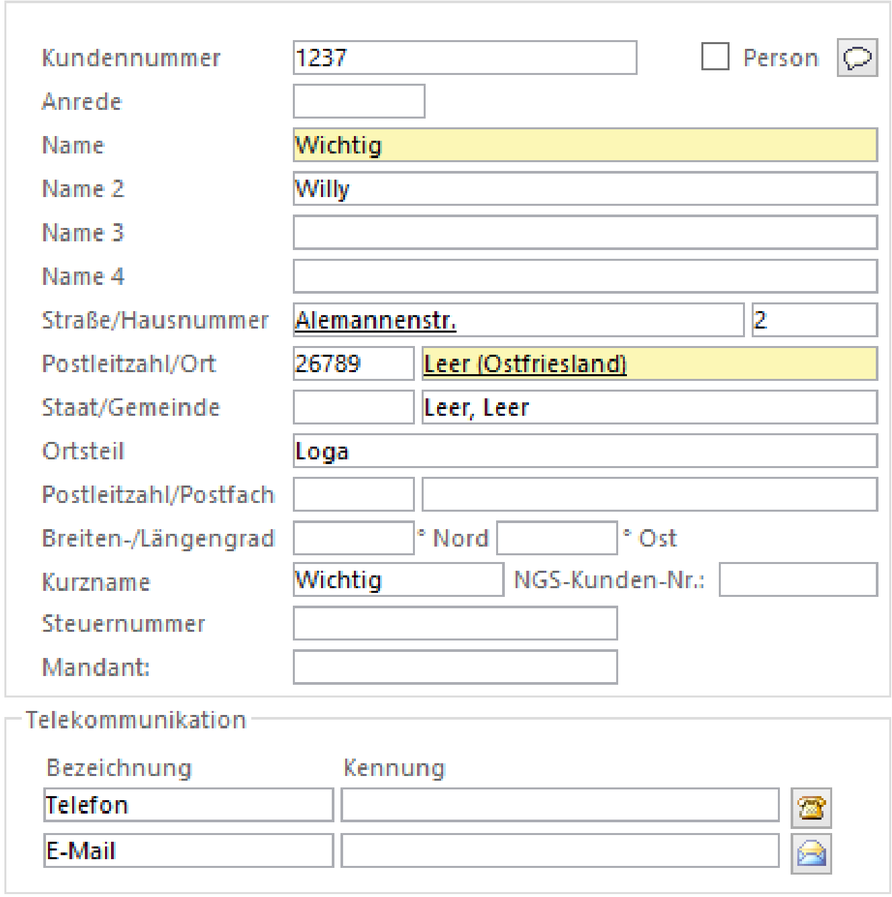
Word/PDF Documents with Custom URL‘s



Interactive Reports with Custom URLs

Kundenumsatzstatistik Top 10

Kunde	Nummer	Anteil
Wichtig, Willy, 26789 Leer	1237	32,48%
Bernd Untermann, 26789 Leer	1059	29,43%
Tollmann, GmbH & Co. KG, 26823 Aurich	1029	18,49%
Tankstelle Tankgut, 67435 Neustadt	1146	9,80%
Adamovic GmbH, 52080 Aachen	1094	3,82%
Bunten, yyy, 26632 Ihlow	1073	1,82%
Baat, 26736 Krummhörn	1170	1,79%
Maus, Manfred, 26789 Leer	1173	1,12%
diverse Barzahler, 26789 Leer	4001209	0,68%
Hallo, Heinrich, 26789 Leer	1177	0,57%
		Gesamtsumme:



Inter-App Communication by using URI Schemes (Basics)

Our Solution:

Inter-App Communication by using URI Schemes

But what are URI Schemes?

Inter-App Communication by using URI Schemes (Basics)

- A **Uniform Resource Identifier (URI)** is a compact sequence of characters used to identify a resource.
- The most common form "**Uniform Resource Locator" (URL)** refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism.
- **Schemes** specifying a concrete **syntax** and associated protocols define each URI.
- Such identification enables interaction with representations of the resource using specific **protocols**.

See https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

Inter-App Communication by using URI Schemes (Basics)

- In general the Syntax of a **URL Scheme** starts with the name of the scheme followed by “://” and followed by further information’s.
  URI Scheme Example **myurl://abc …**

- **Protocols and Schemes** can be divided into two different types: Standard and Application.

- **Standard Protocols/Schemes** are mostly used and are common known. This includes http://, https://, ftp://, and file://

Inter-App Communication by using Custom Protocols (Basics)

Application Protocols/Schemes

- Application-specific, which are usually fairly simple.
- Instead of sending a content to the browser, these allow a browser or other program to start another (local or network) third-party application.
- The URL can be passed to this third-party application as a parameter.
- This is often referred to as "Custom Schemes“ or “Custom URLs”.
- There is the possibility for developers to expand the operating system with additional protocols (custom protocols / custom schemes).

Inter-App Communication by using Custom Protocols (Basics)

- Creating your own ‘custom’ URL Schemes is supported on all major operating systems like Windows, Android or iOS.
- Calling (custom) URL Schemes is possible from Desktop, Web and Mobile Apps.

Outcome:

Inter-App Communication (directly or time-delayed) is possible to implement by using custom URL schemes

But how to do?

Calling (custom) URI Schemes

Different Situations for Calling a (custom) URL Scheme

1. Calling a Desktop App (using PowerBuilder)
2. Calling a WebSite (using plain HTML)
3. Calling a Web App (using PowerServer Web)
4. Calling a Mobile App (using or not using PowerServer Mobile)

Calling (custom) URI Schemes
Implementation | Components

CustomUrlDemo - Target

- Possibility to create/embed URL‘s though the clipboard
- App to call commands and also used to receive commands
- Additional use of registry settings for commands and further informations

AppCaller - Target

- App that is associated with the custom URL Scheme (myurl://) and is startet by calling the custom URL
- Helper application to redirect commands from customURLDemo (sender) to customURLDemo (receiver)
- Registration of the custom URL (e.g. myurl://) can be done using the App

Calling (custom) URI Schemes

create/embed URLs - working with URLs

```
// Sample Link
ls_hyperlink = "myurl://Window1:200"
// replace special charcters
ls_hyperlink = gnv_app.inv_string.of_utf8urlencode( ls_hyperlink )
// RTF Text
ls_hyperlinkRTF  = '{\rtf1\ansi\ansicpg1252 {\colortbl ;\red0\green0\blue255;}{\field{\*\fldinst
HYPERLINK "' + ls_hyperlink + '"}{\fldrslt{\cf1\ul ' + ls_hyperlink + '}}}}'
// Html Text
ls_hyperlinkHtml = '<a href="' + ls_hyperlink + '">' + ls_hyperlink + '</a>'
// write the different formats to the clipboard
lnv_clip.of_copyToClipboard ( ls_hyperlink, ls_hyperlinkRTF, ls_hyperlinkHtml )
```

Calling (custom) URI Schemes
Calling a Desktop App

1.1 from Web App (using PowerServer Web)

- no reload for Desktop App
- by custom URL Scheme e.g myurl://abc….
- call custom URL by external function *shellExecuteW*
- use App (*AppCaller*) associated with the custom URL Scheme
- *postmessage* to Desktop App handle() and pb_custom event by *AppCaller*

1.2 from Website (using plain HTML)

- see 1.1
- Instead of calling custom URL by external function shellExecuteW
  call custom URL custom URL Link

1.3 from Desktop App (using PowerBuilder)

- see 1.1

Calling (custom) URI Schemes
Calling a Web App (using PowerServer Web)

2.1 from Web App (using PowerServer Web)

- no reload for Web App
- by custom URL Scheme e.g myurl://abc….
- call custom URL by external function *shellExecuteW*
- use App (*AppCaller*) associated with the custom URL Scheme
- *postmessage* to Desktop App handle() and wm_activate event (ID=6) by *AppCaller*

2.2 from Website (using plain HTML)

- see 2.1
- instead of calling custom URL by external function shellExecuteW
  call custom URL custom URL Link

2.3 from Desktop App (using PowerBuilder)

- see 2.1

Calling (custom) URI Schemes

App Caller - Part 1

```
// Write Values to the Registry, so the application can read  them.
RegistrySet( is_reg_path, "CustomURLDemo.Kommando", RegString!, ls_kommando )
RegistrySet( is_reg_path, "CustomURLDemo.Parameter", RegString!, ls_parameter )

// Get the Handle of the current application from registry; where to send the message
ll_ret = RegistryGet ( is_reg_path, "CustomURLDemo.Handle", ReguLong!, lul_handle )
```

Calling (custom) URI Schemes

App Caller - Part 2

```
// get information if Receiver App is an Appeon Web App – e.g. from Registry
ll_ret = RegistryGet ( is_reg_path, "CustomURLDemo.Clienttyp", RegString!, ls_ClientTyp )

// Workaround if Appeon Web because pb_custom is not supported , or use a timer with Web Apps
IF ls_ClientTyp="WEB" THEN
        ll_eventid=6   // wm_activate
ELSE
        ll_eventid=1033   //  just  an id (default) to send using postmessage;
                          //   pb_custom event for desktop apps
END IF
PostMessage ( lul_handle, ll_eventid, 1000, 100 )
```

Calling (custom) URI Schemes

Calling a Desktop or Web App

```
// post message to handle of Receiver App (CustomURLDemo) called by the AppCaller

FUNCTION boolean PostMessage(ulong whandle,UINT wmsg,ulong wParam,ulong lParam)
LIBRARY "user32.dll" alias for "PostMessageA;Ansi“


PostMessage ( lul_handle, ll_eventid, 1000, 100 )


// for calling a (custom) URL scheme from CustomURLDemo  or AppCaller

FUNCTION long ShellExecuteW( ulong hWnd, string Operation, string lpFile, string lpParameters,
string lpDirectory, int nShowCmd ) LIBRARY "shell32.dll" alias for "ShellExecuteW“


gnv_app.shellExecuteW ( ll_null, 'OPEN', ls_link,"", ls_null, 5 ) // SW_SHOW =5
```

Calling (custom) URI Schemes
Calling a Web Site (using plain HTML)

3.1 from Web App (using PowerServer Web)

- by URL Scheme http://localhost/...
- call URL by external function *shellExecuteW*

3.2 from Website (using plain HTML)

- see 3.1
- Instead of callling URL by external function shellExecuteW
  call URL HTML Link

3.3 from Desktop App (using PowerBuilder)

- see 3.1

3.4 from mobile App (using PowerServer Mobile)

- by URL Scheme http://localhost/...
- Call URL by *HyperLinktoURL* from inet PB object

Calling URI Schemes

Calling from Mobile App

```
// HyperLinkToURL

inet inv_inet

inv_inet = CREATE inet
getContextService ( "Internet", inv_inet )


inv_inet.HyperlinkToURL ( ls_link )
```

Calling (custom) URI Schemes
Calling a Mobile App

4.1 from mobile App using PowerServer Mobile

- no reload for Mobile App
- by custom URL Scheme e.g myurl://abc….
- Call custom URL by *HyperLinktoURL* from inet PB object

4.2 from Website using HTML

- See 4.1
- Instead of calling custom URL by HyperLinktoURL from inet PB object
  call custom URL by custom URL Link

Creating custom URI Schemes

You also can create your own **custom** URL Scheme

This is possible for:

1. Desktop Apps
2. Mobile Apps (using PowerServer Mobile)

Creating custom URI Schemes
for Desktop Apps

- A One-Time Registration is needed for a new custom URL scheme (like myurl://) by setting some registry keys.
- Admin rights are needed to set the registry keys.
- It can be done by an App and also by a Registry File.

Creating custom URI Schemes

Base Windows registry keys for Custom URL scheme (e.g. “myurl”)

```
ls_app   = "" + as_apppath + ""   // complete path of executable
// description of the protocol – scheme => ls_proto
RegistrySet ( 'HKEY_CLASSES_ROOT\' + ls_proto, '', RegString!, "URL:My Own Protocol" )

// special empty key for a  Custom URL
RegistrySet ( 'HKEY_CLASSES_ROOT\' + ls_proto, 'URL Protocol', RegString!, "" )

// Standard-Icon, Display of  ICON,  EXE  +  Index of Icon, separated by comma
RegistrySet ( 'HKEY_CLASSES_ROOT\' + ls_proto + '\DefaultIcon', '', RegString!, ls_app + ",1" )

ls_app   = ls_app + " "%1"   // add parameters
RegistrySet ( 'HKEY_CLASSES_ROOT\' + ls_proto + '\shell\open\command', '', RegString!, ls_app )
```

Creating custom URI Schemes
for Mobile Apps (PowerServer Mobile)

- Support for custom URL Scheme is a new feature in PowerServer Mobile
- By Customizing and packaging Appeon Workspace you can set your own custom URL scheme. The standard Appeon Workspace scheme is appeonaws://
- You can use a special Syntax to call a single app from inside the Appeon WorkSpace e.g. appeonaws://?url=http://172.20.10.11/custurl_m&command=window1+parameter=1 where http://172.20.10.11/custurl_m is the installed App
- &command=window1+parameter=1 is the parameter name and value pair(s) to be passed to the destination app. It is optional.
- When it is passed to the app the event “oe_urlschemesucceed” is triggered and it can be obtained by the app using “of_geturlschemeparm”

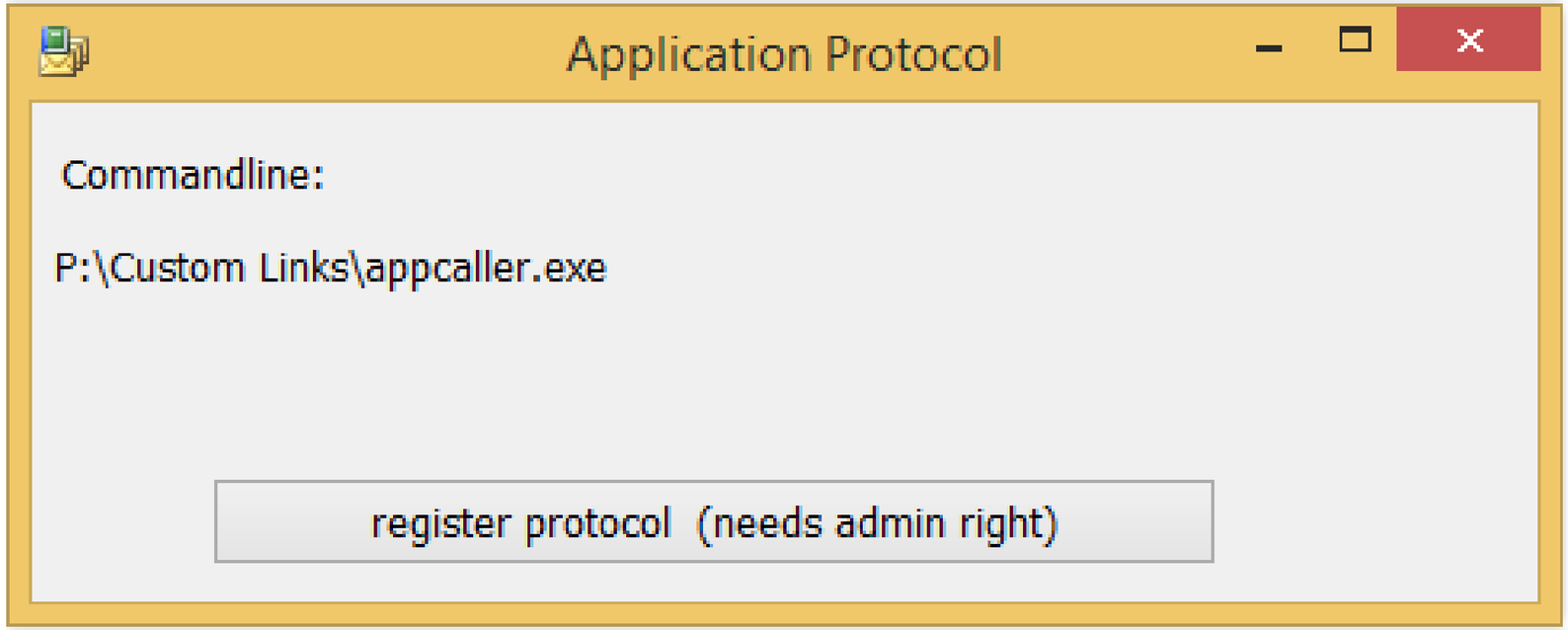
Creating custom URI Schemes

receiving a custom URL call for mobile Apps (PowerServer Mobile)

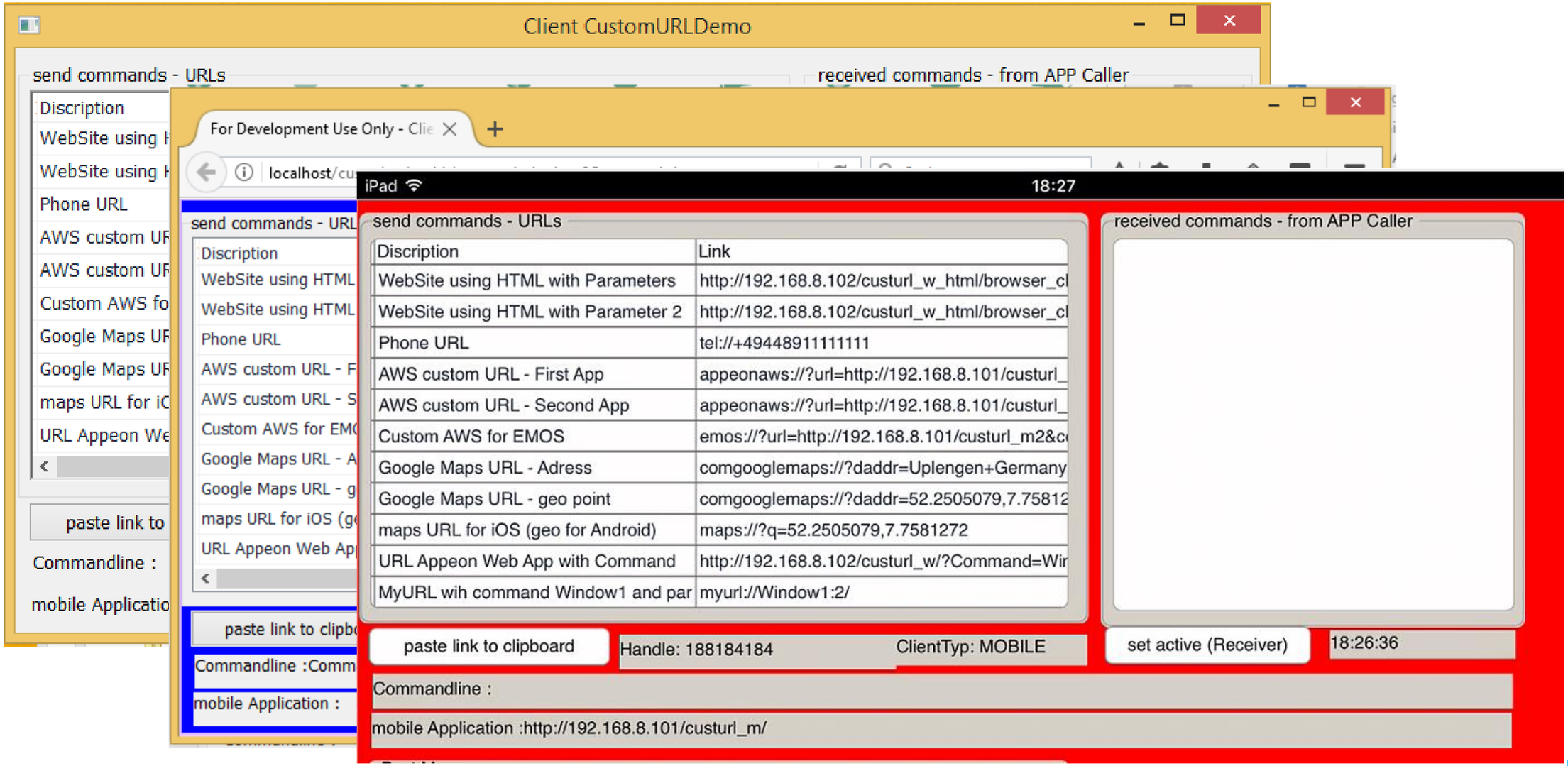
```
// binding custom URL to object and event as needed
eon_mobile_awsex inv_aws
inv_aws = create eon_mobile_awsex
inv_aws.of_register(this,"ue_customevent") // bounding event oe_urlschemesucceed


// create event ue_customevent for the bounded object
STRING ls_customurl
inv_aws.of_geturlschemeparm ( ls_customurl ) // get parameters by reference
```

Demo

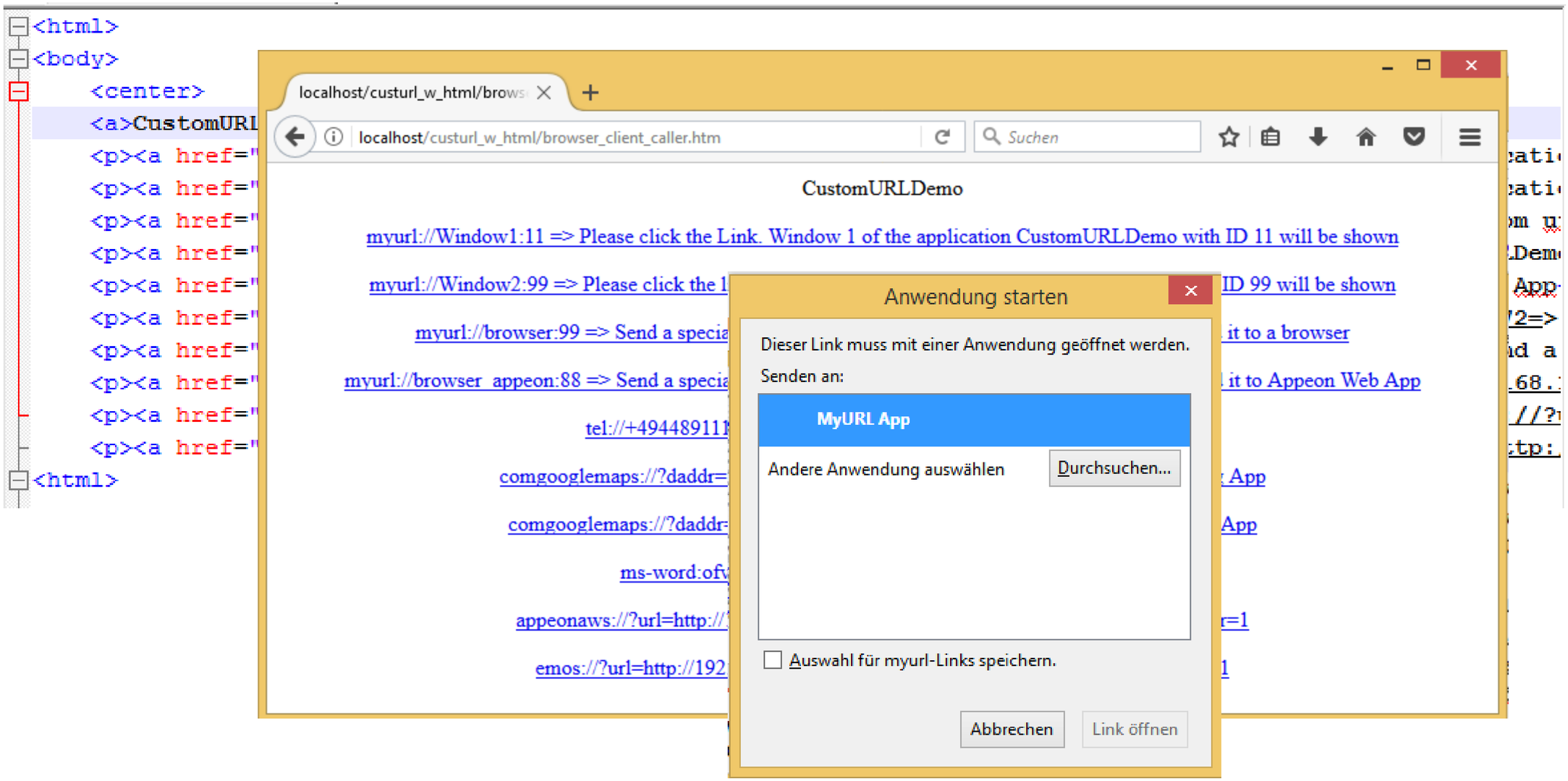
AppCaller – the custom URL App



Using URL Schemes with Desktop, Web and Mobile Apps



WebSite with (custom) URL Schemes browser_client_caller.htm (using plain HTML)



Sourcecode

IF you are interested in the complete Source Code for the CustomURLDemo and the AppCaller then send me an email.

I will send it to you for free

Conclusion

- Inter-App Communication can be implemented by using (custom) URL schemes and can be used with Desktop, Web and Mobile Apps
- Inter-App communication by using (custom) URL schemes are a standardized way to extend an application's functionality and to interact with other applications. It is also language and device agnostic
- The presentation shows an excerpt of possibilities by using (custom) URL schemes, which today can be used with little effort and cross platform.
- It is widely used today by many native desktop, mobile or web applications.

Inter-App Communication based on (custom) URL schemes can be used as an independent and low budget solution to implement a communication between all PowerBuilder and non-PowerBuilder applications.

References and Sources

[Microsoft] - Registering an Application to a URI Scheme
http://msdn.microsoft.com/en-us/library/aa767914(v=vs.85).aspx

[Microsoft Protocols] - Understanding Protocols (Blog Artikel bei MSDN))
http://blogs.msdn.com/b/ieinternals/archive/2011/07/14/url-protocols-application-protocols-and-asynchronous-pluggable-protocols-oh-my.aspx

[Clarify] - Launching Clarify via a custom clarify:// URL
http://clarify.dovetailsoftware.com/gsherman/2012/06/04/launching-clarify-via-a-custom-clarify-url/

[Microsoft Clipboard1] - MSDN - Beliebige Datenformate in die Zwischenablage kopieren
http://msdn.microsoft.com/de-de/library/bb979249.aspx

[Microsoft Clipboard2] - Using the Clipboard
https://msdn.microsoft.com/en-us/library/windows/desktop/ms649016(v=vs.85).aspx

[Microsoft Clipboard3] - Clipboard Formats
https://msdn.microsoft.com/en-us/library/windows/desktop/ms649013(v=vs.85).aspx

Thank You