How did we break it in the first place?

How the back button works..

> Browser records navigation actions
>> Each page reload becomes a history item
>> Going back simply means loading the previous URL

> Classic Web Applications rely on full page reloads
>> CustomerOverview.asp
>> Customer.asp?number=12
>> Order.asp?number=121

> http://localhost/Order_Entry_17/CustomerReport.asp

AJAX Applications

> Rely less and less on page reloads
>> Refresh parts of the page with data loaded using AJAX Calls

> Single page apps
>> Do not reload the page at all
>> Browsers do not record history at all
>> Do not support deep links
>> DataFlex WebApp Framework is such a single page app

> http://localhost/WebOrderMobile_19/

How browsers solved this..

> Modern browsers have JavaScript API's for manually maintaining history
>> history.pushState
>> history.replaceState
>> popState event

> History back & forward operations are handled in JavaScript as well

> Used to make the back button 'work'

> Can change the displayed URL without page load

WebApp Framework

> Exposed low level API's

> Standard URL generation
>> For both drill-down and desktop-style

> Standard history management

> http://localhost/WebOrderMobile_19_1/

Deep link URL support

> Turned out to be the key in the project
>> After we got that, the back button was easy

> System that works automatically
>> Drill-down style
>>> Tightly integrated with drill-down model
>>> [...SalesPersons/Orders-CM/Order-2172](#)
>> Desktop style
>>> Easy
>>> Modal dialogs are not part of the URL
>>> [...Inventory-BEARS](#)

State Hash

> String that represents the state of the application
> Includes
>> View names   (customizable by property)
>> Record ids   (based on main index, defaults to primary key)
>> Custom data   (implement event to add)
> Per view
>> {viewname}-{recordid}-{custom}
>> Example: Product-BEARS
> Drill-down
>> {view hash}/{view hash}
>> Inventory/Product-BEARS

Behavior

> History represents history of user actions
>> Not tied to the breadcrumb

> No data loss warnings when using the back button

> Changes might get lost when navigating back

> Changes are always lost when navigating forward

> Desktop-style webapps do not refind the record when going back
>> Record-id is in the URL so you can copy-paste

> Select views do not always reselect the same record (depends on the invoking view)

Drill-down navigation

> Loading a deep link URL restores the entire navigation path
>> It goes through entire navigation process
>> View loading is optimized
>>> Invoking views are loaded asynchronously in the background
>> Navigation type is figured out based on the URL
>>> Invoking objects can't be figured out and need to be provided explicitly sometimes
>>>> OnGetDefaultInvokingObject (cWebView)

> Navigation type determines which record ID is in the URL

Custom navigation

> Generate custom state hash

> > OnDefineCustomStateHash

> > > Fired each time URL is generated

> Use custom hash inside OnNavigateForward

> > CustomStateHash

> > > Returns the custom hash

> Or exclude from the URL using peStateMode

Customization

> Modes
> > Off
> > History only
> > URLs & History

> Customize generated URLs
> > Change view names
> > Change record id generation
> > Add custom details

> Choose between using the location hash (#..) and inclusion in the URL itself

Custom URLs without hash

> Override URL generation logic
>> Override StateHashToUrl procedure and StateHash function

> Use IIS URL Rewrite module to rewrite URL's

> Add <base element to the HTML fix relative includes

> Configure session manager to write cookies at higher level

> http://localhost:82/

EDUC 2018
European DataFlex users' conference - Edinburgh, Scotland

DataFlex to New Heights

Thank you!

Are there any questions?