

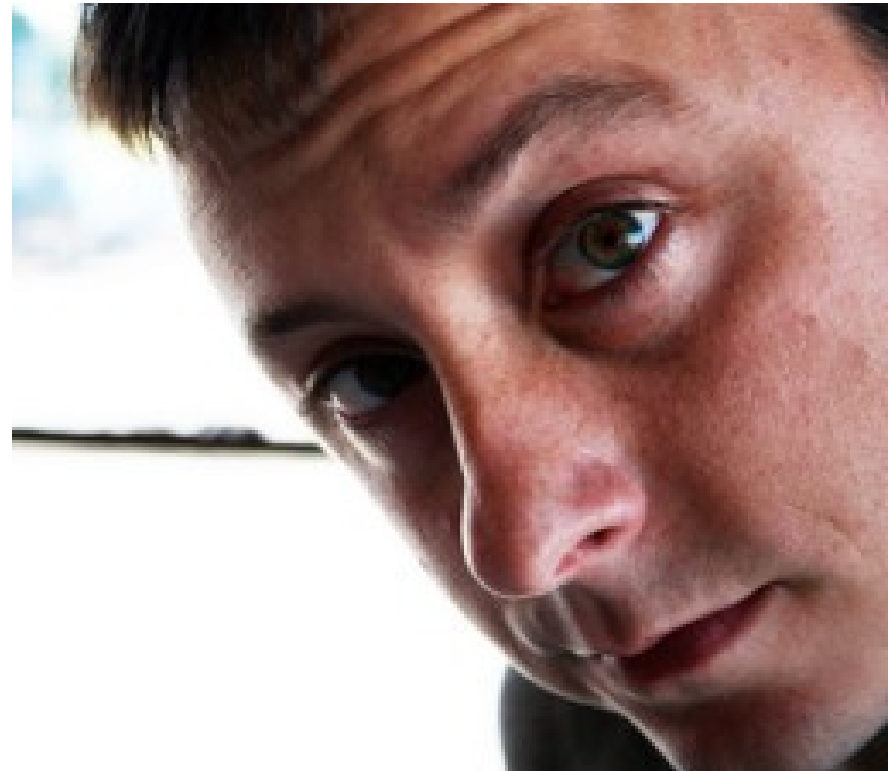
Build A ColdFusion Powered AIR Application The Centaur Edition

Dan Wilson

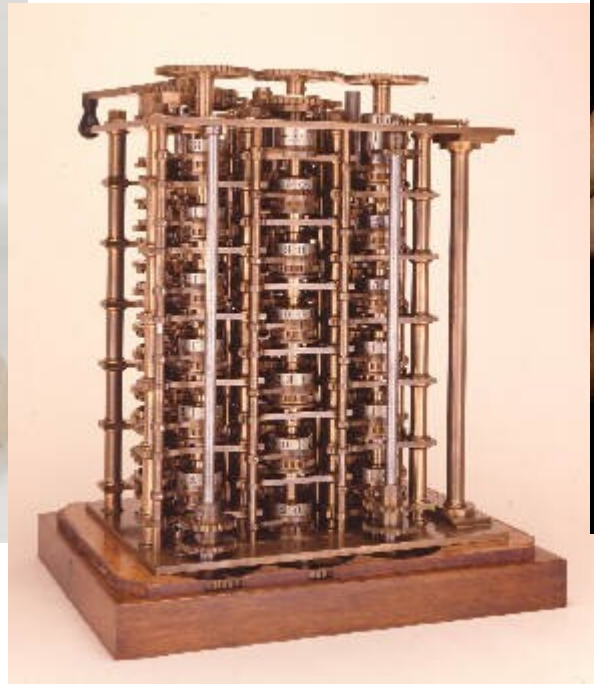


About Dan Wilson

- Principal Partner of DataCurl LLC
- TheHealthChallenge.com
- Adobe Certified Expert
- Adobe Community Expert
- Head of Model-Glue Project
- Manager of TACFUG
- Terrible Typist



Early Computing



DATA  **URL**

50s - Mid 80s Dawn of the Client



DATA**URL**

Mid 80s – Mid 90s



DATA**URL**

The Flood



© 2002 National Geographic Society. All rights reserved.

NATIONALGEOGRAPHIC.COM

DATA  **URL**

Internet Browser As A Client



DATA  **URL**

Ajax Stuff

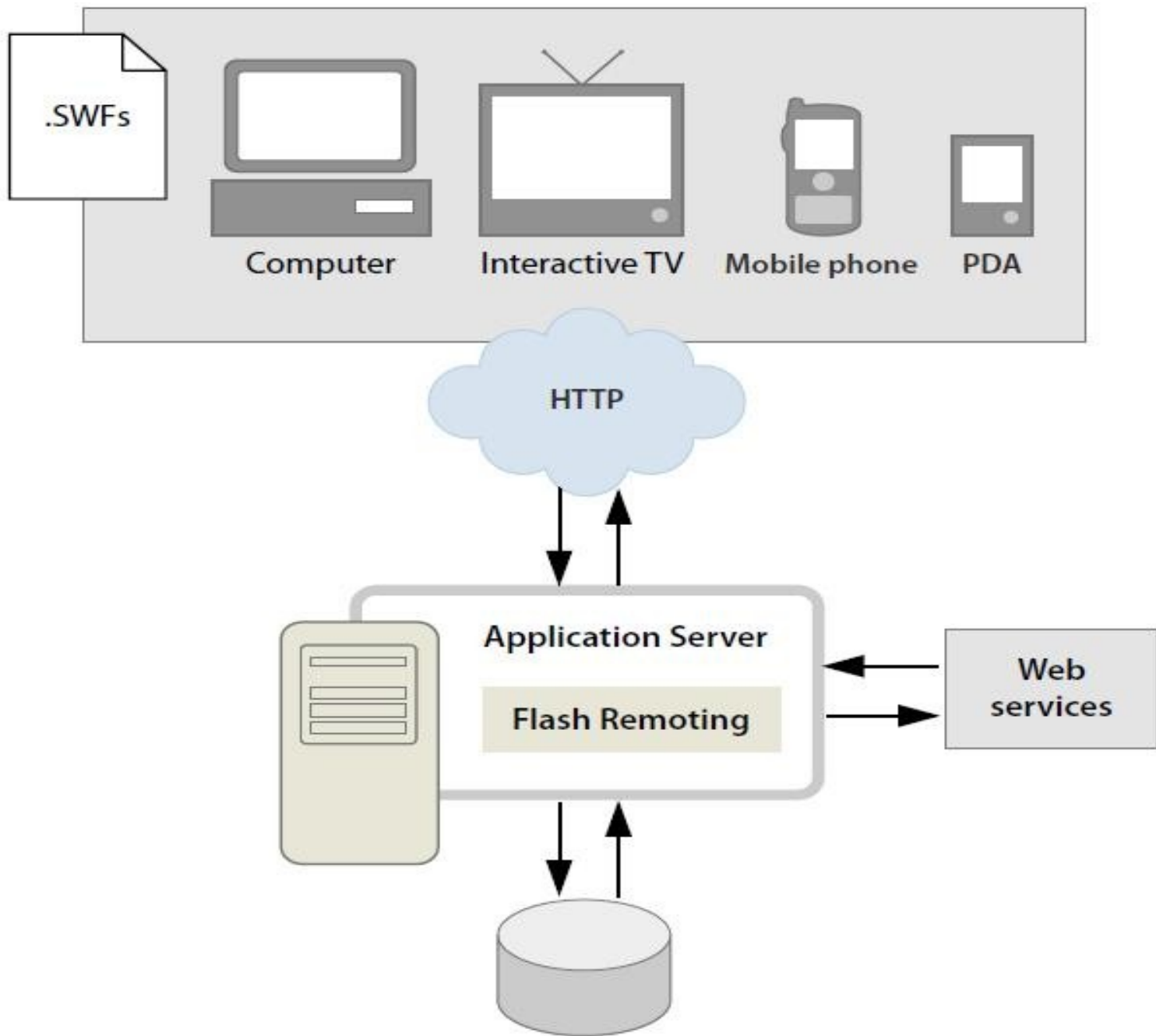


DATA  URL

AIR



DATA**URL**



DATA  **URL**

Let the Feline out of the Container

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml"  
    initialize="init()" title="CF AIR Example" layout="absolute"  
    xmlns:view="view.*" height="600" width="800">  
  
<!-- Air Code Goes Here -->  
  
</mx:WindowedApplication>
```



Start The Show

```
<mx:Script>
<![CDATA[
private function init():void
    var dbFile:File = File.applicationStorageDirectory;
    dbFile = dbFile.resolvePath("LocalDatabase.db");
    if (!dbFile.exists) {
        var defaultDB:File = File.applicationDirectory;
        defaultDB = defaultDB.resolvePath("assets/LocalDatabase.db");
        defaultDB.copyTo(dbFile);
    }
}
]]>
</mx:Script>
```



eVenting a little

```
import flash.events.Event;

public class DataRetrievedEvent extends Event{

    public var data:ListCollectionView;

    public var dataSource:String;

    public function DataRetrievedEvent(type:String, dataSource:String,
data:ListCollectionView){

        super(type, true, false);

        this.dataSource = dataSource;

        this.data = data;

    }

    override public function clone():Event {

        return new DataRetrievedEvent(this.type, this.dataSource, this.data);

    }

}
```



eVenting More

```
<view:Commenters id="commentersView" dataRetrieved="cacheData(event)"/>
```

....

```
private function resultHandler(e:ResultEvent):void {  
    remoteObjectData = e.result as ArrayCollection;  
  
    // send this to be cached  
  
    var dataEvent:DataRetrievedEvent = new  
DataRetrievedEvent("dataRetrieved", "commenters", remoteObjectData);  
  
    dispatchEvent(dataEvent);  
}
```



3 Ways To Get Data

Rest/XML

<?xml?>

WebServices



RemoteObject



DATA **URL**

Rest / XML - ColdFusion

```
<cfsavecontent variable="theXML">
<?xml version="1.0" encoding="UTF-8" ?>
<cfoutput><query rows="#topCommentedCategories.recordcount#"></cfoutput>
<cfoutput query="topCommentedCategories">
  <row>
    <amount>#Amount#</amount>
    <category>#Category#</category>
  </row>
</cfoutput>
</query>
</cfsavecontent>
<cfcontent reset="true" type="text/xml"><cfoutput>#trim(theXML)#</cfoutput>
```



REST/XML – XML Data

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<query rows="10">
```

```
  <row>
```

```
    <amount>330</amount>
```

```
    <category>ColdFusion</category>
```

```
  </row>
```

```
  <row>
```

```
    <amount>278</amount>
```

```
    <category>Rapid Development</category>
```

```
  </row>
```

```
  ....
```

```
</query>
```



Rest/XML - MXML

```
<mx:HTTPService id="restService" method="get" resultFormat="e4x"  
    url="http://cfair.local/Rest.cfm" result="resultHandler(event)"/>
```

```
<mx:Script>
```

```
public function init():void {
```

```
    restService.send();
```

```
}
```

```
private function resultHandler(e:ResultEvent):void {
```

```
    restData = new XMLListCollection(e.result.row as XMLList);
```

```
}
```

```
</mx:Script>
```



Webservice/SOAP - ColdFusion

```
<cffunction name="loadV0" output="false" access="remote" returntype="DataRow[]"
hint="I return an array of Value Objects">

    <cfset var data = loadQuery() />

    <cfset var result = arrayNew(1) />

    <cfset var V0 = "" />

    <cfoutput query="data">

        <cfset V0 = createobject("component", "DataRow") />

        <cfset V0['category'] = Category />

        <cfset V0['amount'] = Amount />

        <cfset arrayAppend( result, V0 ) />

    </cfoutput>

    <cfreturn result />

</cffunction>
```



Webservice/SOAP – XML Data

```
<?xml version="1.0" encoding="UTF-8"?  
  
<wsdl:definitions targetNamespace="http://DefaultNamespace"  
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://DefaultNamespace"  
xmlns:intf="http://DefaultNamespace"  
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"  
xmlns:tns1="http://rpc.xml.coldfusion" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  
  <wsdl:types>  
  
    <schema targetNamespace="http://DefaultNamespace"  
xmlns="http://www.w3.org/2001/XMLSchema">  
  
      <import namespace="http://rpc.xml.coldfusion"/>  
  
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />  
  
      <complexType name="DataRow">  
  
        <sequence>  
  
..... to line number 104
```



Webservice/SOAP - MXML

```
<mx:WebService id="webService" wsdl="http://cfair.local/WebService.cfc?wsdl">  
    <mx:operation name="loadVO" resultFormat="object" result="resultHandler(event)"/>  
</mx:WebService>
```

```
<mx:Script>  
public function init():void {  
    webService.loadVO.send();  
}  
private function resultHandler(e:ResultEvent):void {  
    webServiceData = e.result as ArrayCollection;  
}  
</mx:Script>
```



Remote Object - ColdFusion

```
<cffunction name="loadV0" output="false" access="remote" returntype="DataRow[]"
hint="I return an array of Value Objects">

    <cfset var data = loadQuery() />

    <cfset var result = arrayNew(1) />

    <cfset var V0 = "" />

    <cfoutput query="data">

        <cfset V0 = createobject("component", "DataRow") />

        <cfset V0['category'] = Category />

        <cfset V0['amount'] = Amount />

        <cfset arrayAppend( result, V0 ) />

    </cfoutput>

    <cfreturn result />

</cffunction>
```



Remote Object - Data

1010101010101110011000010100110011001001
0010010011100101001110101001010001011010
01100111010100111010101010010000100101101
10111010101110110101011001010101010101110
0110000101001100110010010010010011100101
00111010100101000101101001100111010100111
01010101001000010010110110111010101110110
10101100101010101010111001100001010011001
10010010010010011100101001110101001



Remote Object - MXML

```
<mx:RemoteObject id="remoteObjectService" destination="ColdFusion"
source="RemoteObject" showBusyCursor="true">

    <mx:method name="loadVO" result="resultHandler(event)"/>

</mx:RemoteObject>

<mx:Script>

public function init():void {

    remoteObjectService.loadVO();

}

private function resultHandler(e:ResultEvent):void {

    remoteObjectData = new ArrayCollection();

    for (var key:String in e.result as Array){

        var dataRow:DataRow = new DataRow( e.result[key].category, e.result[key].amount );

        remoteObjectData.addItem( dataRow );

    }

}

</mx:Script>
```



Remote Object – ColdFusion Query

```
<cffunction name="loadRemoteQuery" output="false" access="remote" returntype="query"
hint="I return a query">
    <cfreturn loadQuery() />
</cffunction>
```



Remote Object – MXML CFQuery

```
<mx:RemoteObject id="remoteObjectService" destination="ColdFusion"
source="RemoteQueryObject" showBusyCursor="true">
    <mx:method name="loadRemoteQuery" result="resultHandler(event)"/>
</mx:RemoteObject>

<mx:Script>
public function init():void {
    remoteObjectService.loadRemoteQuery();
}
private function resultHandler(e:ResultEvent):void {
    remoteObjectData = e.result as ArrayCollection;
}
</mx:Script>
```



AIR SQLite DB

```
<mx:Script>
```

```
public function init():void {
```

```
    var dbStatement:SQLStatement = new SQLStatement();
```

```
    dbStatement.sqlConnection = localDB;
```

```
    var sqlString:String = "SELECT * FROM dataCache";
```

```
    dbStatement.text = sqlString;
```

```
    dbStatement.execute();
```

```
    var result:SQLResult = dbStatement.getResult();
```

```
    var cachedData:ArrayCollection = new ArrayCollection(result.data);
```

```
    dataGrid.dataProvider = cachedData;
```

```
}
```

```
</mx:Script>
```



AIR – CFImage

```
<cf:Config id="conf" cfServer="127.0.0.1" cfPort="80"  
    servicePassword="CFAASPass" serviceUserName="CFAASUser"/>
```

```
<cf:Image id="imgFlip" action="rotate" angle="90"  
    source="http://cfair.local/bentaur-source.jpg"/>
```

```
<cf:Image id="imgBorder" action="addborder" color="red" thickness="5"  
    source="http://cfair.local/bentaur-source.jpg" />
```



AIR - CFMail

```
<cf:Config id="conf" cfServer="localhost" cfPort="80"  
    servicePassword="CFAASPass" serviceUserName="CFAASUser" />  
  
<cf:Mail id="mailtest" server="localhost"  
    to="{to.text}" bcc="{bcc.text}" cc="{cc.text}"  
    failTo="undeliverable@nodans.com" replyTo="admin@nodans.com"  
    subject="{subject.text}" content="{mailbody.text}"  
    from="{from.text}" type="text" charset="utf-8" mailerId="CF"  
    priority="1" timeout="60" useTLS="true" wrapText="5" />
```



AIR – SyncManager - ColdFusion

```
<cfcomponent implements="CFIDE.AIR.ISyncManager">  
    <cffunction name="fetch" returnType="Array" access="remote">  
        <cfset cus = EntityLoad("Customer")>  
    <cfreturn cus>  
    </cffunction>  
    <cffunction name="sync" returnType="any">  
        <cfargument name="operation" type="string" required="true">  
        <cfargument name="clientobject" type="any" required="true">  
        <cfargument name="originalobject" type="any" required="false">  
        <cfif operation eq "INSERT">  
            <cfset EntitySave(clientobject)>  
            ...  
        </cfif>  
    </cffunction>  
</cfcomponent>
```



AIR – Customer.cfc - ColdFusion

```
<cfcomponent persistent="true" alias="Customer">  
    <cfproperty name="cid" fieldtype="id" >  
    <cfproperty name="name" >  
    <cfproperty name="street" >  
</cfcomponent>
```



AIR – SyncManager - MXML

```
private function init():void
{
    syncmanager = new SyncManager();
    ...
    var token:AsyncToken= syncmanager.fetch("fetch");
    token.addResponder(new mx.rpc.Responder(fetchSuccess, fetchFault));
}
private function fetchSuccess(event:SyncResultEvent):void
{
    var cus:Array = event.result as Array;
    var sessiontoken:SessionToken =syncmanager.openSession(dbFile,017915);
    sessiontoken.addResponder(new mx.rpc.Responder(connectSuccess,connectFault));
}
```



AIR – Customer.as - MXML

```
package onetoone
{
    [Bindable]
    [Entity]
    [RemoteClass(alias="Customer")]
    public class Customer
    {
        [Id]
        public var cid:int;
        public var name: String;
        public var street: String;
    }
}
```



Credits

Shannon Hicks (iotashan.com)

helped with the Flex code

Terry Ryan (numtopia.com)

gave me permission to show CF9 features

The TACFUG Crew (tacfug.org)

let me bore them with a rough version of this presentation





BENTAUR

Questions?



BENTAUR

My Company: <http://www.datacurl.com>

My Blog: <http://www.nodans.com>

Free account at The Health Challenge

<http://thehealthchallenge.com>

Invite Code: CFObjective

DATA**URL**