

Using IntraWeb

by John Evans

In 2002 I gave a presentation at a DDG/BUG meeting about a case study that I had built over the last two years that made use of the web broker and page producer components. In 2002 I watched the arrival of the Web Snap components...I came to a Masterclass and never touched them again. Then, during 2003, I came to the Masterclass for IntraWeb, and really liked the look of what it could do. I decided to use it in anger on a project for a company I partly own.

This article is based on my presentation to the Developers Group in February 2005. It was not a "How to do it talk", it was a "what I did talk", containing my solutions to an eclectic list of problems. I should say at this point that if any of you know any of my assumptions to be incorrect, please do email me, I am not proud.

The website is www.face2facenetworking.co.uk. Please do have a look around it, watch the URL change, use the back button (try to!), look at the images used in the menu and in grids...the article will make more sense if you have taken a brief look at the website. The reader is expected to understand some of the basics of IntraWeb: I urge you to go a read up on it if you aren't familiar with it.

Problem 1 - Form inheritance

According to the IntraWeb notes, it is not supported. I really needed it so I tried it anyway. It works just fine, and it is now more or less acknowledged as being supported.

Problem 2 - The datamodule

This was a change of mindset for me; I am used to developing desktop applications. IntraWeb provides something called the usersession form. Use this for your datamodule and all will be well.

Put all of your session variables in the usersession module. Since we were taking membership details, or bookings for events, the number of the session details got quite large. Making good use of the usersession module makes session control a breeze, and removes one of the biggest headaches from the web broker model.

Here is the code for when a member logs on. It demonstrates that various usersession properties are set for the rest of the session:

Code Snippet 1

```
procedure TIWfrmMain.LogonButtonClick(Sender: TObject); //also in the baseform
begin
  usersession.qMember.Open;
  usersession.qNewCompany.open;
  if
    usersession.qMember.Locate('usercode;password', vararrayof([edusercode.Text, edpassword.Text]), [])
  then
    begin
      if usersession.qNewCompany.Locate('pkey', usersession.qMemberCompanypkey.AsInteger, [])
      then
        begin
          usersession.isloggedon := true;
          usersession.pmemberpkey := usersession.qMemberpkey.asinteger;
          usersession.useremail := usersession.qMemberemail.AsString;
          usersession.logondate := datetostr(date);
          usersession.fullname := usersession.qMemberFname.AsString+
            '+usersession.qMemberLname.AsString;
          IWRegionMember.Visible := true;
          IWRegionlogon.Visible := false;
          textWelcome.Lines[0] := usersession.fullname;
        end else
          webapplication.ShowMessage('Error with database, please email face2face');
        end else
          webapplication.ShowMessage('Invalid usercode or password');
    end;
end;
```

The base form and mainform also have the following OnCreate event:

```
procedure TIWfrmBase.IWAppFormCreate(Sender: TObject);
begin
  buildmenu;
  if ((usersession.isloggedon) and (strtodate(usersession.logondate)=date)) then
  begin
    IWRegionMember.Visible := true;
    IWRegionlogon.Visible := false;
    textWelcome.Lines[0] := usersession.fullname;
  end
  else
  begin
    IWRegionMember.Visible := false;
    IWRegionlogon.Visible := true;
  end;
end;
```

Problem 3 - The back button

One of the great things about the IntraWeb model is that it manages session state so well. Unfortunately, this makes the back button very difficult/impossible to use in a way that is conducive with the user's existing experience. The main problem with not having a back button is that every Internet user in the world is used to it and is frustrated if they can't use it. I had to consider how to navigate "back" wherever the user was. Explaining to users that not having a back button is actually quite a good idea does not go down well!

Problem 4 – Using Images

Of course using an image in the IntraWeb image components is easy, but my web designer wanted more than that. *'Can't we replace the clunky text in the grids and menus with nice images?'* Well you can, and it was quite hard to work out. In particular I wanted the main menu to display a different image (orange instead of blue) for the current page, so this list has to get built each time the page form is created. The menu is in a frame and the click event for the menu is:

Code snippet 2A

```
procedure TFrame2.menuClick(Sender: TObject; ItemIdx: Integer);
var IWfrmMain:TIWfrmMain;
    IWfrmAboutUs:TIWfrmAboutUs ;
    IWfrmMembership:TIWfrmMemberShip ;
    IWfrmJoin:TIWfrmJoin ;
    IWfrmSearch:TIWfrmSearch ;
    IWfrmEvents:TIWfrmEvents ;
    IWfrmMemberspage:TIWfrmMemberspage ;
    IWfrmContactUs:TIWfrmContactUs;
begin
  case Menu.SelectedIndex of
    //not showing all cases
    0:
      begin
        usersession.menuitem := 0;
        IWfrmMain := TIWfrmMain.create(WebApplication);
        IWfrmMain.show;
      end;
    1:
      begin
        usersession.menuitem := 1;
        IWfrmAboutUs := TIWfrmAboutUs.create(WebApplication);
        IWfrmAboutUs.Show;
      end;
    2:
      begin
        usersession.menuitem := 2;
        IWfrmMembership := TIWfrmMembership.create(WebApplication);
        IWfrmMembership.Show;
      end;
  end; //case
end;
```

For the static menu, the onClick event has to be in the menu frame, but the creation of the images has to be in the form AFTER if has been created.

Code snippet 2B

```
procedure TIWfrmBase.buildmenu;
begin
  menuframe.menu.Items[0].Caption:='<IMG src="/isapi/face2face.dll/Files/home.gif">';
  menuframe.menu.Items[1].Caption:='<IMG src="/isapi/face2face.dll/Files/about.gif">';
  menuframe.menu.Items[2].Caption:='<IMG src="/isapi/face2face.dll/Files/membership.gif">';
  menuframe.menu.Items[3].Caption:='<IMG src="/isapi/face2face.dll/Files/join.gif">';
  menuframe.menu.Items[4].Caption:='<IMG src="/isapi/face2face.dll/Files/search.gif">';
  menuframe.menu.Items[5].Caption:='<IMG src="/isapi/face2face.dll/Files/diary.gif">';
  menuframe.menu.Items[6].Caption:='<IMG src="/isapi/face2face.dll/Files/myhome.gif">';
  menuframe.menu.Items[7].Caption:='<IMG src="/isapi/face2face.dll/Files/contact.gif">';

  case usersession.menuitem of
    0: menuframe.menu.Items[0].Caption:='<IMG src="/isapi/face2face.dll/Files/homeclick.gif">';
    1: menuframe.menu.Items[1].Caption:='<IMG src="/isapi/face2face.dll/Files/aboutclick.gif">';
    2: menuframe.menu.Items[2].Caption:='<IMG src="/isapi/face2face.dll/Files/membershipclick.gif">';
    3: menuframe.menu.Items[3].Caption:='<IMG src="/isapi/face2face.dll/Files/joinclick.gif">';
    4: menuframe.menu.Items[4].Caption:='<IMG src="/isapi/face2face.dll/Files/searchclick.gif">';
    5: menuframe.menu.Items[5].Caption:='<IMG src="/isapi/face2face.dll/Files/diaryclick.gif">';
    6: menuframe.menu.Items[6].Caption:='<IMG src="/isapi/face2face.dll/Files/myhomeclick.gif">';
    7: menuframe.menu.Items[7].Caption:='<IMG src="/isapi/face2face.dll/Files/contactclick.gif">';
    99: //do nothing the page shown is not on the menu
  end; //case
end;
```

The reference to the image file location also has to be explicit for Internet Information Services (IIS), which makes local testing in the IntraWeb web server a real pain.

Problem 5 – Launching Separate Pages

There are two ways to launch a new browser page: one is to use the URL component, the other is to add JavaScript in the page 'onload' event.

We discovered that the JavaScript worked fine, *except* it is regarded by Norton Personal firewall (and probably other things) as a pop-up and hence is blocked for many users. Of course the user doesn't tell you this...they simply tell you it just isn't working.

To launch it from the URL meant that the page must already exist, but we wanted the user to see information changing from our site, for example the list and details of all people booked onto the next networking event. This would change every time someone booked for the event. So I went back to something I still hold dear, the PageBuilder component.

Code snippet 3

```
procedure TIWUserSession.qEventBookingAfterPost(DataSet: TDataSet);
begin
  buildeventpage(qeventbookingguesteventpkey.AsInteger);
end;

procedure TIWUserSession.buildeventpage(event: integer);
var  FileName : String;
     newFileLocal : TextFile;
begin
  FileName := 'D:\websites\face2face\attfiles\' + inttostr(event) + '.htm' ;
  AssignFile(NewFileLocal,Filename);
  usersession.qPrinteventdetails.Close;
  usersession.qPrinteventdetails.Parameters.ParamValues['peventpkey']:=Event;
  usersession.qPrinteventdetails.Open;
  usersession.qPrinteventguests.Close;
```

```

usersession.qPrinteventguests.Parameters.ParamValues['peventpkey']:=Event;
usersession.qPrinteventguests.Open;
if FileExists(FileName) then //reset(NewFileLocal) else
Rewrite(NewFileLocal);
Writeln(NewFileLocal,usersession.pgprintattendeess.Content);
closefile(NewfileLocal);
end;

procedure TIWfrmEventAttendees.AttendeesButtonClick(Sender: TObject);
begin
  WebApplication.NewWindow(lURL,'attendees',600,800, [woscrollbars]);
end;

```

In my usersession, I have AfterPost events for the event booking queries. This makes sense because the page is only rebuilt based on actual changes to the bookings. The filename uses the primary key value of the event, so the filename to look for is easy. I then use the PageBuilder component to rebuild a physical file on the web server every time the bookings for the event change.

The URL link on the IntraWeb page is built whenever the page is opened, using the primary key value of the event (to build the page variable lURL):

```

lURL:='http://www.face2facenetworking.co.uk/attfiles/'+
      intostr(usersession.qEventDetailspkey.asinteger)+'.htm';

```

Problem 6 – Office Interface

One of the things I wanted to do as members joined or booked an event was to send them an email. This is pretty simple (using the Indy components). Code snippet 4 should give you a flavour of how this can be achieved.

Code snippet 4 (heavily cut down to show the basics)

```

procedure TIWUserSession.SendEmail(emailaddr:string; emailtype:integer; bodytext:string);
var attach:string;
idattachment:Tidattachment;
begin
  email.Body.Clear; //email is an idmessage component
  email.ReplyTo.Clear;
  email.Recipients.Clear;
  email.From.Address :=qparamsF2FEmailFrom.AsString;
  email.Sender.Address:=qparamsF2FEmailFrom.AsString;
  email.ReplyTo.Add.address:= qparamsF2FEmailReplyTo.AsString;
  email.Recipients.Add.address :=trim(emailaddr);
  email.Body.Add(qparamsF2FEmailHeader.AsString);
  case emailtype of
    1: // Thanks for New Join
      begin
        email.Subject:='Face2Face Membership';
        email.Body.Add(qparamsEmailWelcomeMembers.AsString);
        email.Body.Add('');
        email.Body.Add(bodytext);
        if usersession.qNewMemberGuest.AsString='Y' then else
          begin
            attach:=createInvoice(1); //see snippet 5
            TIdAttachment.Create(email.MessageParts, attach );
          end;
        end;
      end;
  end;
  mailsender.Username := qparamsSMTPUsercode.AsString; /mailsender is an idSMTP component
  mailsender.Password := qparamsSMTPPassword.AsString;
  mailsender.Host := qparamsSMTPIPaddress.AsString;
  mailsender.Connect;
  try
    mailsender.Send(email);
  finally
    mailsender.Disconnect;
    idattachment.Free;
  end;
end;

```

Next I wanted to build and attach an invoice. My first attempt was to use Word templates, mail merge and the good old Office Partner components. This worked a treat locally; however deploying it under IIS was another problem. It gave all sorts of wacky errors [*Ed. Technical Term*]. It was not happy about the DCOM part, and after weeks of frustration I read a posting from someone who said it would never work.

I looked around for an alternative. I bought the component set called edocEngine from Gnostice. Basically it can translate Delphi created reports including those from ReportBuilder, into external formats like Excel, PDF or in my case RTF.

Here is the code for building the invoice:

Code snippet 5

```
function TIWUserSession.CreateInvoice(invType:integer):string;
var
  i : Integer;
  ThisInvoice:string;
begin
  qparams.Open;
  qparams.Edit;
  qparamsNextInvoice.AsInteger:=qparamsNextInvoice.AsInteger+1;
  qparams.Post;
  ThisInvoice := 'D:\f2fdocs\invoice'+qparamsNextInvoice.asString+'.rtf' ;
  gtRTFEngine1.FileName:= ThisInvoice;
  gtRBExportInterfacel.Engine := gtRTFEngine1;
  if invtype=1 then
  begin
    Qsinglecompany.close;
    Qsinglecompany.Parameters.ParamValues['pkey']:=usersession.qNewMemberpkey.AsInteger;
    Qsinglecompany.Open; //this query is used by the RB report
  //Exporting a ReportBuilder Report
    gtRBExportInterfacel.RenderDocument(Repinv); ///repinv is the RB report
  end;
  result:=thisinvoice;
end;
end;
```

Problem 7 – Worldpay Interface

Over the course of this article, I have been building up to my biggest headache. I had never interfaced anything to Worldpay (or similar) before and it was conceptually quite challenging. I have to say the support of Delphi developer John Hair was invaluable, so thank you John.

There are three stages/parts to the interface.

Stage 1

Sending the information to Worldpay.

Code snippet 6

(this is heavily cut down, and the instid.value has been changed to protect our bank account!)

```
procedure TIWfrmPay.PopulateWorldPayScript;
var sl: TStringList;
    Year, month, day:word;
begin
  sl := TStringList.Create;
  try
    begin
      TIWfrmPay(WebApplication.FindComponent('IWfrmPay')).HiddenFields.Text :=
        'instId=1' + #13#10 +
        'cartId=1' + #13#10 +
        'amount=1' + #13#10 +
        'currency=1' + #13#10 +
        'testMode=1' + #13#10 +
        'name=1' + #13#10 +
        'email=1' + #13#10 ;
      sl.Add('document.SubmitForm.email.value = '' + UserSession.PayEmail + '';');
      sl.Add('document.SubmitForm.name.value = '' + UserSession.PayFullname + '';');
      sl.Add('document.SubmitForm.amount.value = '' + FloatToStrF(UserSession.TotalCost,
        ffFixed, 18, 2) + '';');
    end;
  end;
```

```

sl.Add('document.SubmitForm.action = "https://select.worldpay.com/wcc/purchase";');
sl.Add('document.SubmitForm.method = "POST";');
sl.Add('document.SubmitForm.instId.value = "12345";'); // ***** Your installation id
sl.Add('document.SubmitForm.cartId.value = "` + Usersession.worldpayID + `";');
sl.Add('document.SubmitForm.currency.value = "GBP";');
sl.Add('document.SubmitForm.testMode.value = "0";');
sl.Add('document.SubmitForm.submit();');

    btnWorldPay.ScriptEvents.Values['OnClick'] := sl.Text; //btnworldpay is a button
finally
    sl.Free;
end;
end;

```

Looking through code snippet 6, you can see that this is fairly easy. You can have the hidden fields defined as a form property but we have more than one case so we build on the fly. Keep in mind that the user has actually left our site. Worldpay will process the transaction and either reject or accept it.

Stage 2

I wanted the user to come back to the site, so I worked out how to use a Worldpay feature called redirect, as shown in code snippet 7:

Code snippet 7 (the URL called by the Worldpay redirect).

<http://www.face2facenetworking.co.uk/isapi/face2face.dll?redirect='12345'&transstatus='Y'>

```

(a part of the mainform OnCreate event)
if checkforRedirect then
begin
doRedirect;
end;

function TIWfrmMain.CheckForRedirect: Boolean;
var sl: TStringList;
begin
Result := False;
sl := TStringList.Create;
try
if (WebApplication.RunParams.Count > 0) then
begin
sl.Text := WebApplication.RunParams.Text;
if sl.Values['Redirect'] <> '' then
begin
Result := True;
Usersession.WorldpayID := sl.Values['Redirect'];
Usersession.OrderAccepted := sl.Values['transStatus'] = 'Y';
end;
end;
finally
sl.Free;
end;
end;

```

We pass over a variable, unique to this transaction and get it passed back to us. This means we need to modify Stage 1. Before we send the user to Worldpay we write a record to our database that includes our unique transaction number, and any session data that we might need.

The Worldpay redirect function launches a URL that starts a new session of our IntraWeb ISAPI application. This session has no knowledge whatsoever of the previous session, so we need to pass it the parameters. We interrogate the IWMain form OnCreate event to see if it has a redirect parameter. If it has, we look up in our database to find out the session information that we had before we left the site. We then set the various usersession variables as required. We now have the user back with us in their new session. *[Ed. Magic!]*

At this point I thought I was finished, but I was missing a big BUT, and John Hair patiently explained it to me about ten times until the penny dropped. Here is the big "Gotcha"; please read it carefully.

Stage 3

When the user is at Worldpay, there is no control over where they go next. So how does *my* database know that the Worldpay transaction was successful or not?

The answer is that Worldpay provides a function called callback. It invokes a URL to the IntraWeb application that executes the ISAPI code, but is not visible (read this a few times). So a URL, similar to the redirect, (but passing callback as the parameter) is called, and handled like this:

Code snippet 8

```
procedure TIWfrmMain.IWAppFormCreate(Sender: TObject);
begin
  buildmenu;
  if checkforcallback then
  begin
    doUpdate; // this will get executed, then the application session closes
    exit;
  end;
  if checkforRedirect then
  begin
    IWRegionThanks.Visible:=true;
    doRedirect;
  end;
  if ((usersession.isloggedon) and (strtodate(usersession.logondate)=date)) then
  begin
    IWRegionMember.Visible:=true;
    IWRegionlogon.Visible:=false;
    textWelcome.Lines[0]:=usersession.fullname;
  end
  else
  begin
    IWRegionMember.Visible:=false;
    IWRegionlogon.Visible:=true;
  end;
end;

procedure TIWfrmMain.doUpdate;
var  emailbody:string;
      eventbody:string;
begin
  usersession.qWorldpay.open;
  // Get the OrderID for this order
  if UserSession.qWorldpay.Locate('pkey',strtoint(usersession.WorldPayID),[]) then
  begin
    case usersession.qWorldpaytranstype.AsInteger of
      1,2:
        begin
          usersession.qNewMember.Open;
          usersession.pmemberpkey:=usersession.qWorldpayuserpkey.AsInteger;
          if usersession.qNewMember.Locate('pkey',usersession.pmemberpkey,[]) then
          begin
            if usersession.OrderAccepted then
            begin
              if usersession.qWorldpayreferrerpkey.AsInteger=0 then  else
              begin
                usersession.qAddReferral.Open;
                usersession.qAddReferral.Append;
                usersession.qAddReferralMemberpkey.AsInteger:=usersession.qWorldpayreferrerpkey.AsInteger;
                usersession.qAddReferralDateRef.AsDateTime:=date;
                usersession.qAddReferralDateJoined.AsDateTime:=date;
                usersession.qAddReferralNewMemberpkey.AsInteger:=usersession.pmemberpkey;
                usersession.qAddReferral.Post;
                usersession.qLookupmembers.Open;
                if
                usersession.qLookupmembers.Locate('pkey',usersession.qWorldpayreferrerpkey.AsInteger,[])
                then
                begin
                  emailbody:='Dear '+usersession.qLookupmembersfname.AsString+ #13+#10+
                    'Thank you for referring
                    '+usersession.qNewMemberFullname.AsString+
                    ' of '+usersession.qNewMemberCompany.AsString+ #13+#10+
                    'to face2facenetworking.'+ #13+#10+
                    'You are now entitled to £25 off your next annual subscription!'+
                    #13+#10+
                end
            end
          end
        end
    end
  end
end;
```

```

        'You do not need to do anything else; the calculation will be made
automatically when you renew your membership.'+ #13+#10+
        'Many thanks for your contribution!' + #13+#10;
        usersession.sendEmail(usersession.qlookupMembersemail.AsString,3,emailbody);
    end;
end;
usersession.qNewMember.edit;
usersession.qNewMemberpaidOK.asstring:='Y';
usersession.qNewMember.post;
emailbody:=#13+#10+'Your Membership number is
'+inttostr(usersession.qNewMemberMembershipNo.AsInteger)+
        #13+#10+'Your Usercode is '+usersession.qNewMemberUserCode.AsString+
        #13+#10+'Your Password is
'+usersession.qNewMemberPassword.AsString+#13+#10;
usersession.sendEmail(usersession.qNewMemberemail.AsString,1,emailbody);
end else
begin
    usersession.qLookupmembers.open;
    if
    usersession.qLookupmembers.Locate('pkey',usersession.qWorldpayuserpkey.AsInteger,[])
    then
    begin
        usersession.qLookupmembers.delete;
    end;
    end;
end;
end;
end; //case
end;

end;

```

In the first procedure we see some of the OnCreate event of the main form. If docallback is true (it will be), it executes the second procedure and then closes the application.

The second procedure checks the database to find the (same) session record, uses this to update the database to say if the user paid or not, and if they paid, it then creates and sends the email (using the code seen earlier).

To summarise: ALL Worldpay transactions will execute a callback, *some* will execute a redirect. The callback happens automatically, the redirect only if the user clicks the appropriate button in Worldpay. Interestingly, when I presented to the Developers Group, several people had the same confusion as I originally did before the penny dropped.

Support

There is no doubt that IntraWeb is a great idea with plenty of great features. It is also in its infancy in an area that is changing fast. I found lots of faults, and lots of instability. AtoZed had a policy of issuing a fix release almost every month. This was a bad idea: there were several cases of one fix causing further faults, or reintroducing previously fixed faults.

I use the TMS components, which again have some great features, and have faults, but importantly are only qualified against specific versions of IntraWeb, so the whole fix upgrade process becomes more complicated. While it might be possible to use IntraWeb without third party components, I think a richness of content and ease of use would be lost without them. The efforts of AtoZed support staff and the Team B members has been to the usual high standard: they are just working in an area where they have so little control over the end user environment.

Browsers

The single biggest area of ongoing problems is with browser incompatibility. Much of this site looks odd under Firefox or Netscape. It has serious problems under Mac browsers. The technical response is often "It is the fault of the browser, the browser implements non-standard xyz." This may be true, but cuts no ice with users who insist that they get on fine with most other web sites in the same browser.

Creating code examples to send to these guys to prove my problems is very complicated and time consuming, especially if the fault occurs on someone else's machine, when it runs under IIS and looks fine to me.

Conclusions

I really like many of the features of IntraWeb. I really like that fact that it is so similar to developing desktop applications. For anyone who has managed session control before, the ease of that now is wonderful.

If it is to compete with other development environments, it has to have the features that are offered by TMS and by Arcana as standard. Third party components should be nice to have as add-ons. I do not think this is the case here. Commercial projects are therefore reliant on TMS as much as anything else. I would like to see all three companies merged (or their IntraWeb parts), under funding from Borland, and to spend some time getting what they have to be stable. I know this is unlikely; it is just my thoughts.

If I was to quote now for a client project where they wanted an *intranet*, with the browser choice controlled and the desktop environment controlled, I would immediately use IntraWeb. If it is a complex *Internet* project, I would not.

If any readers need a little more advice on any of these topics, then please email me on John@Clear-Advantage.co.uk. It gives me a chance to give back some of the support I have been given.

Links

www.AtoZed.com

www.TMSsoftware.com

www.arcanatech.com

www.Gnostice.com

John Evans runs Clear Advantage, based in the Cotswolds. www.clear-advantage.co.uk

