

# **IQX Web Referencing Setup Guide**

**V1.02**

**IQX Release 2.16.2 onwards**

**Gareth Johnstone and Ivan Baldwin**



## Table of Contents

What you need to run it.....	3
IQX Database Update.....	3
SQL for the database table.....	4
Woodpecker File Update. ....	6
System Administrator Setup .....	7
IQXNet Setup.....	7
Switchable Views Setup .....	8
Enabling Web Reference Settings .....	8
Setup Branding.....	9
Reference Questionnaire Setup.....	11
Job Runner Scheduling.....	13
Web Reference – Sending out the reference requests.....	14
Reference Request Job - Send Web Reference Emails.xml .....	14
Reference Request Job - Send Web Reference Reminder Email.xml .....	19
The Summary – Send Web Reference Summary.xml. ....	24

## What you need to run it

- A version of IQX 2.16.2 and above with Agency Setup Branding onwards.
- IQXHub (latest) and associated IQXWeb functionality
- nodeJS

IQX will set up IQX Hub and IQXWeb along with associated database and wpk changes. This will be a chargeable item.

## IQX Database Update

**This information is provided for information purposes only. Changes to the database should only be carried out by experienced personnel. Before any changes please ensure a copy of the database has been made.**

Table ReferenceRequest should already be in the database. However, if the table is incomplete it should be deleted and the following SQL run to recreate the table.

Columns	Constraints	Referencing Constraints	Indexes	Triggers	Dependent Views	Data					
PKey	Name	ID	Obj. ID	Data Type	Size	Scale	Cmp.	Nulls	Uniq.	Value	Comment
<input checked="" type="checkbox"/>	ReferenceRequestID	1	11626	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	PersonID	2	11627	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	DepartmentID	3	11628	char	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	RefereeAddressName	4	11629	char	60		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	RefereeSalutation	5	11630	char	60		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	RefereePosition	6	11631	char	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	CompanyName	7	11632	char	60		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Keyname	8	11633	char	60		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Addr1	9	11634	char	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Addr2	10	11635	char	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Addr3	11	11636	char	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Town	12	11637	char	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	County	13	11638	char	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Country	14	11639	char	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	PostCode	15	11640	char	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	HomePhone	16	11641	char	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WorkPhone	17	11642	char	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	MobilePhone	18	11643	char	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Fax	19	11644	char	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Email	20	11645	char	250		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Position	21	11646	char	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	PositionNotes	22	11647	long varchar			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	URLRequestCode	23	11648	char	250		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	Processed	24	11649	smallint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	
<input type="checkbox"/>	StaffID	25	11650	char	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WhenCreated	26	11651	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	current timest...	
<input type="checkbox"/>	WhenCompleted	27	11652	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WhenProcessed	28	11653	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WhoProcessed	29	11654	char	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WhenSent	30	11655	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	WhenReturned	31	20172	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	EmploymentID	32	20173	char	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	RefType	33	20174	smallint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	StartDate	34	41832	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	EndDate	35	41833	timestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	brandid	36	68683	char	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	ReferenceConfirmationRequir...	37	68684	tinyint			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

## SQL for the database table

Copy and paste this SQL into SQL Tool in the IQX program.

```
CREATE TABLE "pears"."ReferenceRequest" (  
    "ReferenceRequestID" CHAR(20) NOT NULL,  
    "PersonID" CHAR(20) NOT NULL,  
    "DepartmentID" CHAR(2) NOT NULL,  
    "RefereeAddressName" CHAR(60) NOT NULL,  
    "RefereeSalutation" CHAR(60) NOT NULL,  
    "RefereePosition" CHAR(40) NOT NULL,  
    "CompanyName" CHAR(60) NULL,  
    "Keyname" CHAR(60) NOT NULL,  
    "Addr1" CHAR(40) NULL,  
    "Addr2" CHAR(40) NULL,  
    "Addr3" CHAR(40) NULL,  
    "Town" CHAR(30) NULL,  
    "County" CHAR(30) NULL,  
    "Country" CHAR(30) NULL,  
    "PostCode" CHAR(20) NULL,  
    "HomePhone" CHAR(100) NULL,  
    "WorkPhone" CHAR(100) NULL,  
    "MobilePhone" CHAR(100) NULL,  
    "Fax" CHAR(100) NULL,  
    "Email" CHAR(250) NOT NULL,  
    "Position" CHAR(100) NOT NULL,  
    "PositionNotes" LONG VARCHAR NULL,  
    "URLRequestCode" CHAR(250) NULL,  
    "Processed" SMALLINT NULL DEFAULT 0,  
    "StaffID" CHAR(20) NULL,  
    "WhenCreated" TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,  
    "WhenCompleted" TIMESTAMP NULL,  
    "WhenProcessed" TIMESTAMP NULL,
```

```
"WhoProcessed" CHAR(20) NULL,  
"WhenSent" TIMESTAMP NULL,  
"WhenReturned" TIMESTAMP NULL,  
"EmploymentID" CHAR(20) NULL,  
"RefType" SMALLINT NULL,  
"StartDate" TIMESTAMP NULL,  
"EndDate" TIMESTAMP NULL,  
"brandid" Char(20) NULL,  
"ReferenceConfirmationRequired" SMALLINT DEFAULT 0,  
PRIMARY KEY ( "ReferenceRequestID" ASC )  
 ) IN "system";  
  
ALTER TABLE "pears"."ReferenceRequest" ADD CONSTRAINT "Person" NOT NULL FOREIGN KEY (  
"PersonID" ASC ) REFERENCES "pears"."Person" ( "personid" );  
  
ALTER TABLE "pears"."ReferenceRequest" ADD CONSTRAINT "Department" NOT NULL FOREIGN KEY  
( "DepartmentID" ASC ) REFERENCES "pears"."Department" ( "departmentid" );  
  
ALTER TABLE "pears"."ReferenceRequest" ADD CONSTRAINT "staff" NOT NULL FOREIGN KEY (  
"StaffID" ASC ) REFERENCES "pears"."staff" ( "staffid" );
```



## System Administrator Setup

This section includes System Administrator setup activities.

### IQXNet Setup

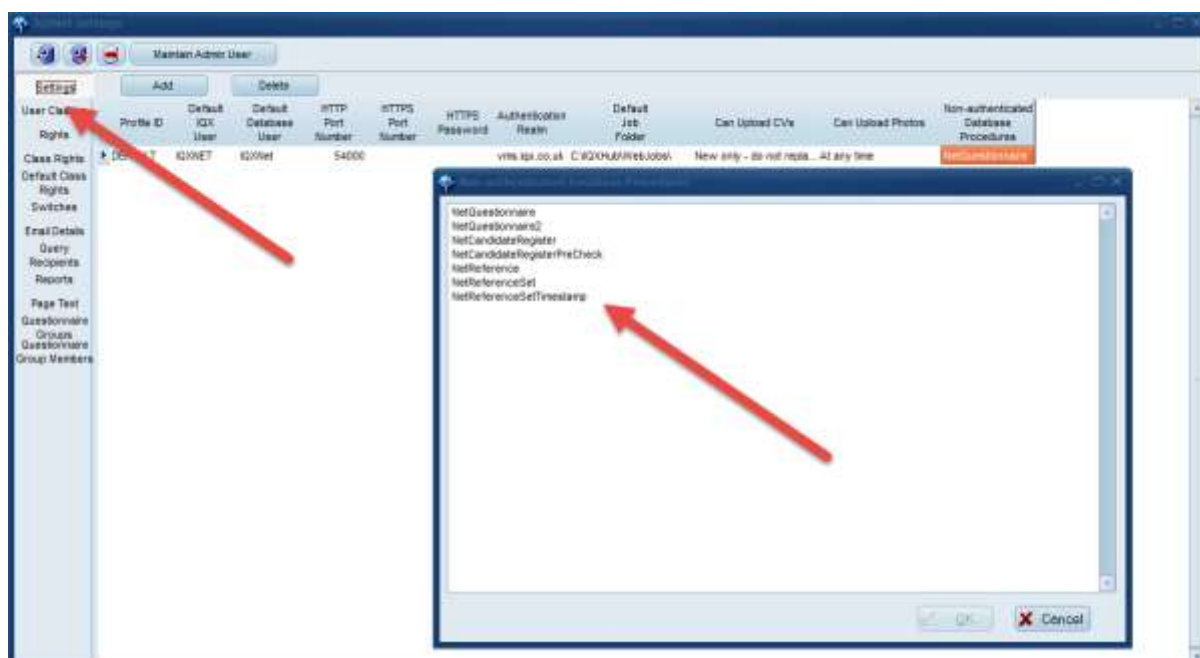
Once the IQXHub etc. has been set up clients will need to set up their IQXWeb Settings.

Under Maintenance > IQXNet Setup

Click on the settings tab then scroll along to Non-authenticated Database Procedures

Double click on the entry and make sure the following are included:

- NetQuestionnaire
- NetQuestionnaire2
- NetCandidateRegister
- NetCandidateRegisterPreCheck
- NetReference
- NetReferenceSet
- NetReferenceSetTimestamp



Whilst in IQXNet Settings set up access rights and other settings.

## Switchable Views Setup

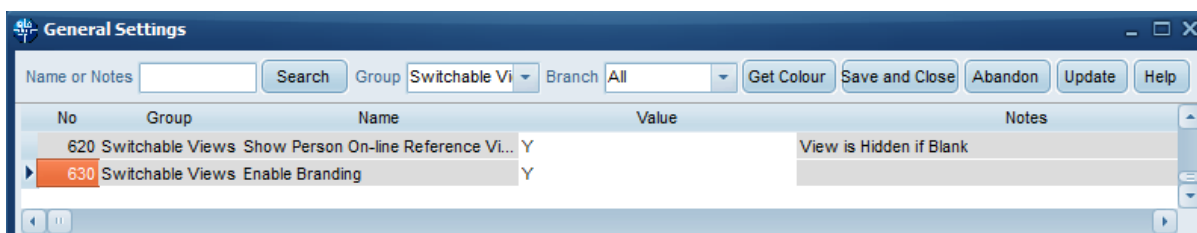
Under Maintenance > General Settings

Enabling Web Reference Settings

Set up the web reference settings as indicated below:

No 620 is required to allow viewing of the web reference.

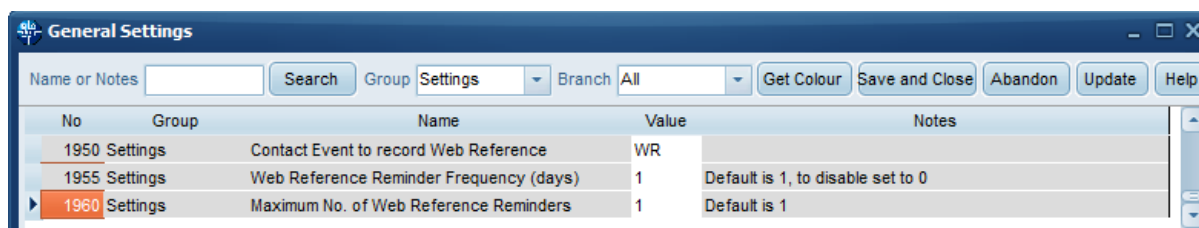
No 630 Enables Branding used in the web reference.



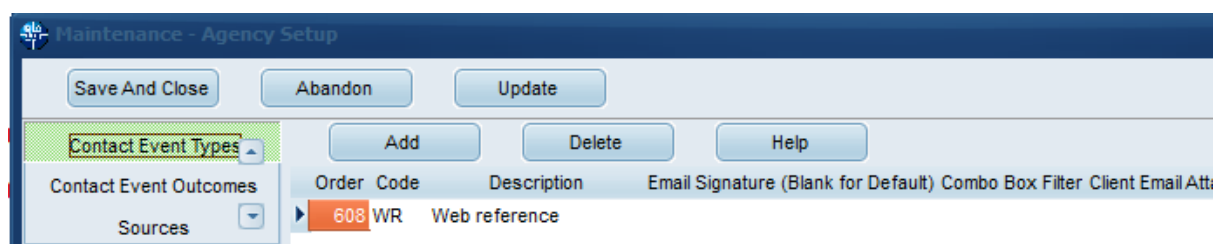
No 1950 activates a contact event option to record web references. WR in this case.

No 1955 determines the frequency web reminders are checked and sent out as necessary.

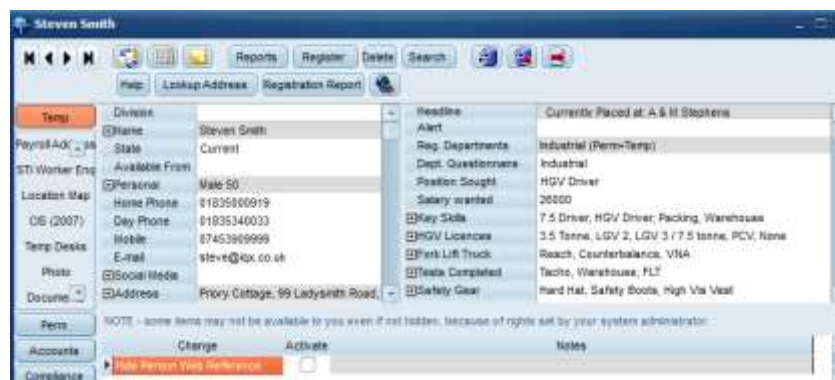
No 1960 sets the maximum number of reference request reminders that will be sent to a referee.



Below is an example of a contact event type set up with the code WR.



In Person – Modify Screen ensure the Hide Person Web Reference box is unticked.





## Setup Branding

- In Agency Setup, locate the branding view, below 'Vacancy Roles'
- Select the Brand function, and press Open. WEBREFERENCE is for companies without full Branding set up. Templates set up in WEBREFERENCE can be included in a company brandS (e.g.TESTBRAND) for web referencing to work within that brand



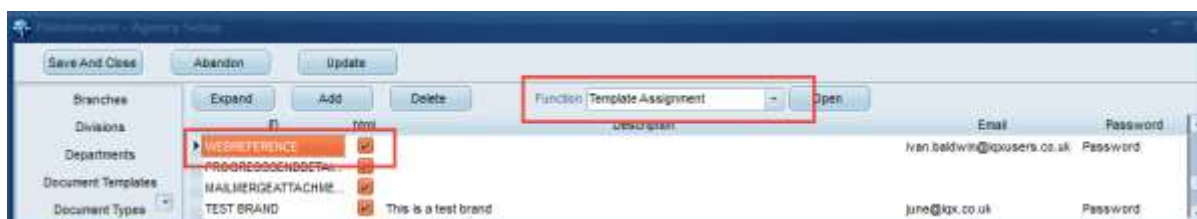
And set up the function as below

```
(
  in persid char(20),in vacid char(20),in ClientOrTemp char(1),in Use char(20) )
returns char(20)
begin
  declare RVal char(20);
  if Use = 'WEBREFERENCE' then
    set rval = 'WEBREFERENCE'
  end if;
  return RVal
end
```

- Back to branding, create a new brand called 'WEBREFERENCE'



- Double click WEBREFERENCE, and create a new template
- You need to assign a template to 'WEB REFERENCE', this can be done by selecting



Once open, set up the function as below and save

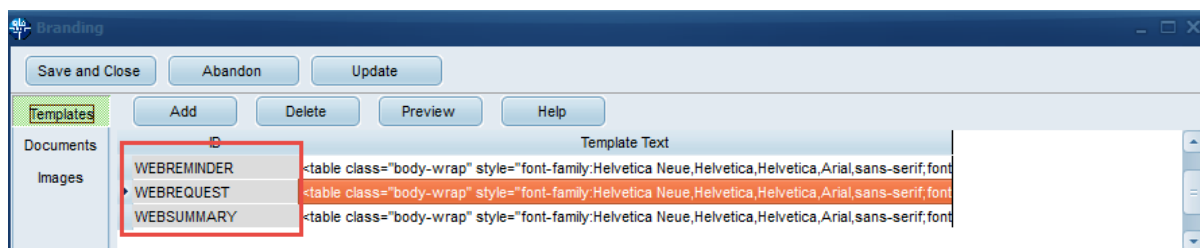
```
(
```

```

in "BehaviourID" char(1),in "BrndID" char(20),in "EventID" char(1),in "TargetType"
char(1),in "ShiftsID" char(1000),in "PlacementID" char(20) )
returns long varchar
begin
-- TargetType C or T (client/temp)
-- behaviourid P, S, E (Post, SMS or Email)
-- EventID C, X or O (Confirmation, Cancellation or Offer)
declare "RVal" long varchar;
declare "ID" char(20);
select "visibleid" into "ID" from "EN_brand" where "brandid" = "BrndID";
if("ID" = 'WEBREFERENCE') and("eventid" = 'D') then
select "brandtext" into "RVAL" from "en_template" where "brandid" = "BrndID" and
"visibleid" = 'REQUEST';
return "RVAL"
end if;
if("ID" = 'WEBREFERENCE') and("eventid" = 'S') then
select "brandtext" into "RVAL" from "en_template" where "brandid" = "BrndID" and
"visibleid" = 'SUMMARY';
return "RVAL"
end if;
return "RVAL"
end

```

- f) Double click on WEBREFERENCE
- g) Create 3 templates as shown below



- h) Xml jobs for sending out reference requests, reminders and a summary of the reference to referees will use the above SQL to pull in the standard web signature,
- i) The standard html wrap shown in summary against each template is reproduced in full below.

```

<table class="body-wrap" style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0; padding: 20px;width: 100%;"bgcolor="#f6f6f6">
    <tr>
    <td></td>
    <td class="container" style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0;padding: 20px;clear: both !important;display: block !important;Margin: 0 auto !important;max-width: 600px !important;border: 1px solid #f0f0f0;" bgcolor="#FFFFFF">
        <div class="content" style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0;width: 100%;display: block;margin: 0 auto;max-width: 600px;">
            <table>
        </div>
    </td>
</tr>

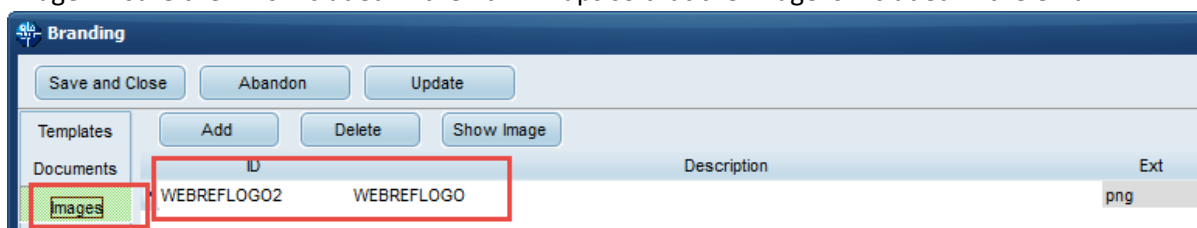
```

```

<td style="padding: 10px 0px 40px 0px;">
  <IMG SRC="{BRD_WEBREFLOGO2}">
</td>
</tr>
<tr>
<td>
  <p style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0;font-size: 14px;font-weight: normal;margin-bottom: 10px;">Dear {REFEREADDRESSNAME}</p>
  <p style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0;font-size: 14px;font-weight: normal;margin-bottom: 10px;">We have been given your details as a reference contact for <strong>{APP_NAME}</strong>.</p>
  <p style="font-family:Helvetica Neue,Helvetica,Helvetica,Arial,sans-serif;font-size:100%;line-height:1.6em;margin:0;padding:0;font-size: 14px;font-weight: normal;margin-bottom: 10px;">Please follow the link below to access the secure on-line reference.</p>
  <br />
  <table class="btn-primary" style="Margin-bottom: 10px;width: auto !important;margin: 0 auto !important;"cellpadding="0" cellspacing="0" border="0">
    <tr>
      <td style="background-color: #348eda; border-radius: 25px;font-family: 'Helvetica Neue', Helvetica, Arial, 'Lucida Grande', sans-serif; font-size: 14px; text-align: center;vertical-align: top; ">
        <a style="background-color: #348eda;border: solid 1px #348eda;border-radius: 25px;border-width: 10px 20px;display: inline-block;color: #ffffff;cursor: pointer;font-weight: bold;line-height: 2;text-decoration: none;" href="{yourURL}/#/webReference?{URLRequestCode}">Click here to access your reference</a>
      </td>
    </tr>
  </table>

```

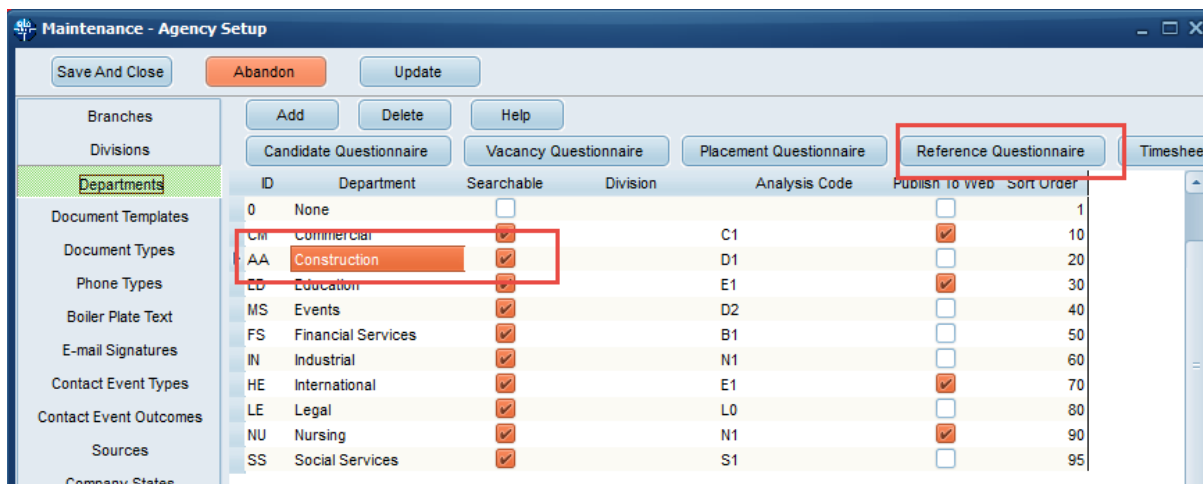
j) Store images used in web referencing in Images. Select Add and find the location of the required image. Ensure the ID is included in the html wraps so that the image is included in the email.



Refer to the IQX Web Referencing User Guide for information on using Web Referencing.

## Reference Questionnaire Setup

The questionnaire must be set up via Maintenance – Agency Setup – Department – Select a department – Click Publish to Web – Reference Questionnaire.



The next window will allow the creation of questions relevant to the Department. Web Publish and Web View must be clicked for this feature to be displayed on the Web. The long description is the question that will be displayed in the reference request.



Each department will need separate Reference Questionnaires set up.

## Job Runner Scheduling

To set up the automatic jobs for emailing the reference request and a summary of their completed reference to the referee got to Maintenance – Database setup.

In Job Runner Schedule Groups create a group for web referencing along with the default directory address housing the two xml jobs.

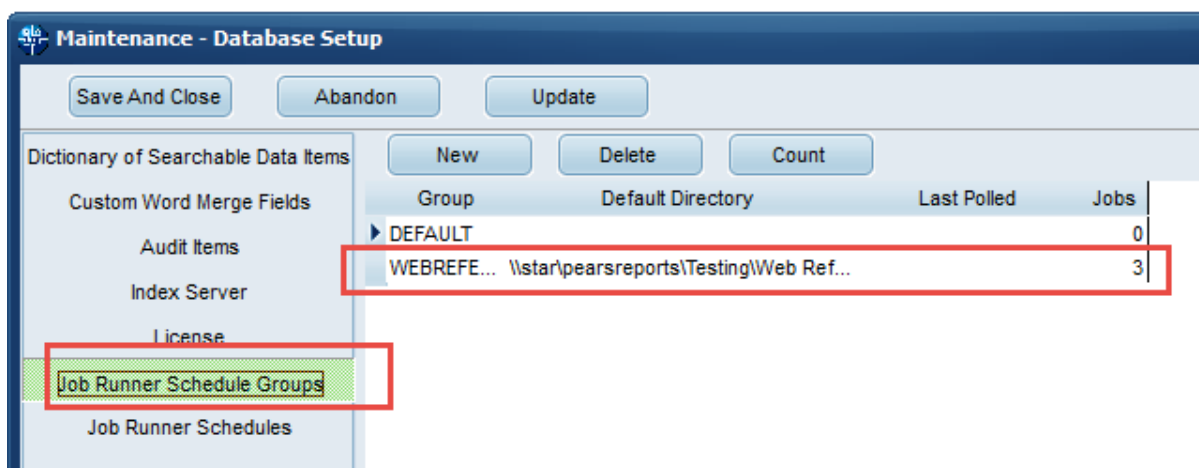
Send Web Reference Emails.xml,

Send Web Reference Reminders.xml and

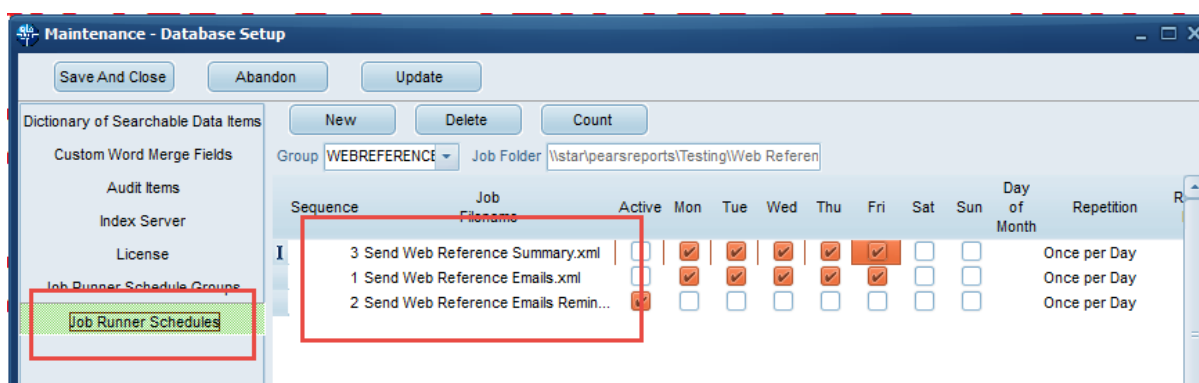
Send Web Reference Summary.xml.

The xml code for these are included at the end of this document.

In Maintenance – Database Setup – Job Runner Schedule Groups – Create a new group and insert the folder location of the xml files.



In Job Runner Schedules schedule the insert the names of the xml jobs and decide on the frequency they will be applied. Note – Active means the jobs will be run as and when they are required.



For testing each job can be run manually via Maintenance – Run a Job and locating the respective xml jobs.

## Web Reference – Sending out the reference requests

All the web reference requests are stored in the database marked as ready to be emailed out.

The job below is can then scheduled to run every night or at intervals set by the System Administrator. It finds all the reference requests that need to be processed, creates an email for each request and then emails it out to the referee with a unique URL so that the referee can access an online form to fill in.

## Reference Request Job - Send Web Reference Emails.xml

NOTE Highlighted text will need populating by the System Administrator.

```
<?xml version="1.0"?>
<Job title="Email Reference Requests" autoclose="yes" dateformat="dd/mm/yyyy" delimiter=","
stringquote="&#34;">
  <Library>
    <SetEmailBody smb_text="" smb_wrap="">
      <SetVariable name="Emailbodytext" type="S" value="{smb_text}" />
      <If x1="{emailhtml}" x2="Yes">
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_text, char(13)+char(10),
'&lt;BR/&gt;');" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_wrap, '{{bodytext}',
:EmailBodyText)" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{signature}', :mysignature) " />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{origmessage}', '')" />
      </If>
    </SetEmailBody>
    <GetBoilerPlateText gbt_text="" gbp_person="" gbp_vacancy="" gbp_company=""
gbp_placement="" gbp_employment="" gbp_brand="">
      <SetVariable name="BPText" value="" />
      <StaticFunction service="CONVERT_BOILERPLATETEXT" person="{gbp_person}"
company="gbp_company" employment="{gbp_employment}" placement="{gbp_placement}"
vacancy="{gbp_vacancy}" brand="{gbp_brand}" text="{gbt_text}" />
      <SetVariable name="BPText" type="S"><![CDATA[{{WebServiceResult}}]></SetVariable>
    </GetBoilerPlateText>
    <DoBranding br_person="" br_vacancy="" br_use="" br_type="" br_id="">
```

```

<if x1="{br_id}" x2="" comparison="!=">
  <SetVariable name="brandid" value="{br_id}"/>
</if>
<else>
  <SetVariable name="brandid" sql="select BrandID from en_Brand where VisibleID =
GetBrand(:br_person, :br_vacancy,:br_type, :br_use)" />
</else>
<!-- work out the template -->
<SetVariable name="emailbody" sql="select GetNotifyDocuments(:br_type, :brandid, 'D','W',','")"
/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{REFEREEADDRESSNAME}}',
:REFEREEADDRESSNAME)"/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{yourURL}}', :yourURL)"/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{URLRequestCode}}',
:URLRequestCode)"/>
<!-- convert the body-->
<Call gbt_text="{emailbody}" gbp_person="{br_person}" gbp_vacancy="{br_vacancy}"
gbp_brand="{brandid}">GetBoilerPlateText</Call>
<!-- slot body, default signature into html wrap-->
<Call smb_text="{BPText}" smb_wrap="{shtmlwrap}">SetEmailBody</Call>
<!-- <messagedialog>{EmailBodyText}</messagedialog> -->
</DoBranding>
</Library>
<SetVariable type="S" name="CRLF" sql="select string(char(13),char(10))"/>
<SetVariable type="S" name="TAB" sql="select string(char(9))"/>
<SetVariable type="S" name="SMTPHost" value="INSERT SMTP HOST" />
<SetVariable type="S" name="SMTPPort" value="INSERT SMTP PORT" />
<SetVariable type="S" name="SMTPTLS" value="INSERT YES OR NO" />
<SetVariable type="S" name="yourURL" value="INSERT URL TO IQXWeb" />
<ChangeDirectory path="{TempDir}"/>
<SQLExec ignoreerror="YES">drop table zzzIQXREF_TEMP_Refs</SQLExec>
<SQLExec>DECLARE LOCAL TEMPORARY TABLE zzzIQXREF_TEMP_Refs (
  ReferenceRequestID char(20),

```

```
Email char(250), primary key (ReferenceRequestID))
on commit preserve rows;
</SQLExec>
<!-- Get default signature -->
<StaticFunction service="GetEmailSignature"/>
<SetVariable name="mysignature" type="S"><![CDATA[{{WebServiceResult}}]></SetVariable>
<!-- get htmlwrap -->
<SetVariable name="emailhtml" type="S" value="Yes"/>
<SetVariable name="shtmlwrap" type="S" sql="select signature from emailsignature where
emailsignatureid='HTML_WRAP'"/>
<IfSQL condition="'shtmlwrap'=''">
  <SetVariable name="emailhtml" type="S" value="No"/>
</IfSQL>
<SQLExec><![CDATA[
insert into zzzIQXREF_TEMP_Refs
SELECT
ReferenceRequestID, replace(trim(Email), ' ', '') as TrimEmail
FROM "pears"."ReferenceRequest"
where whensent is null and whencompleted is null and whenprocessed is null and
TrimEmail like '%@%.' and
TrimEmail not like '% %' and TrimEmail not like '%..%' and TrimEmail not like '%.@"%' and
TrimEmail not like '%.' and TrimEmail not like '%.' and TrimEmail not like '%@%@"%'
]]>
</SQLExec>
<SQLQuery>
<SQLSelect><![CDATA[
select ReferenceRequest.ReferenceRequestID as refid, person.name as PersonName,
RefereeAddressName, RefereePosition, ReferenceRequest.CompanyName as
RefereeCompanyName,
z.Email as RefereeEmail, ReferenceRequest.URLRequestCode,
staff.email as SMTPSender, staff."name" as Staffname, person.personid as pid,
ReferenceRequest.brandid as webbrand,
```



```

en_brand.email as brandemail,
en_brand.visibleid , en_brand.smtppwd
FROM ReferenceRequest key join (staff, person, en_brand),
ReferenceRequest inner join zzzIQXREF_TEMP_Refs z on (z.ReferenceRequestID =
ReferenceRequest.ReferenceRequestID)
order by ReferenceRequest.WhenCreated
  ]]>
</SQLSelect>
<IfAnyRows>
  <ForEachRow>
    <Block>
      <Call br_person="{pid}" br_use="WEBREFERENCE" br_id="{webbrand}"
>DoBranding</Call>
      <SendEmail subject="{Visibleid} Reference Request" smtptls="{SMTPTLS}"
smtpport="{SMTPPort}" smtphost="{SMTPHost}" smtpuserid="{brandemail}" html="YES"
smtpsender="{brandemail}" smtppassword="{smtppwd}">
        <EmailRecipients>
          <WriteLn>{RefereeEmail}</WriteLn>
        </EmailRecipients>
        <EmailBody>{EmailBodyText}</EmailBody>
        <EmailPlainBody>
          <WriteLn><![CDATA[{EmailBodyText}]]></WriteLn>
        </EmailPlainBody>
      </SendEmail>
      <SetVariable name="emailFailed" value="" />
    </Block>
    <Catch><SetVariable name="emailFailed" value="{Exception}" />
      <MessageDialog>{emailFailed}</MessageDialog>
    </Catch>
  </IfSQL condition="{emailFailed} = "">
  <SQLExec>

```

```
        update ReferenceRequest set WhenSent = current timestamp where ReferenceRequestID =
:RefID
    </SQLExec>
</IfSQL>
</ForEachRow>
</IfAnyRows>
</SQLQuery>
</Job>
```

## Reference Request Job - Send Web Reference Reminder Email.xml

This job can be used to send emailed reminders to referees.

NOTE Highlighted text will need populating by the System Administrator.

```
<?xml version="1.0"?>
<Job title="Email Reference Requests" autoclose="yes" dateformat="dd/mm/yyyy" delimiter=","
stringquote="&#34;">
  <Library>
    <SetEmailBody smb_text="" smb_wrap="">
      <SetVariable name="Emailbodytext" type="S" value="{smb_text}" />
      <If x1="{emailhtml}" x2="Yes">
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_text, char(13)+char(10),
'&lt;BR/&gt;')" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_wrap, '{{bodytext}',
:EmailBodyText)" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{signature}', :mysignature)" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{origmessage}', '')" />
      </If>
    </SetEmailBody>
    <GetBoilerPlateText gbt_text="" gbp_person="" gbp_vacancy="" gbp_company=""
gbp_placement="" gbp_employment="" gbp_brand="">
      <SetVariable name="BPText" value="" />
      <StaticFunction service="CONVERT_BOILERPLATETEXT" person="{gbp_person}"
company="gbp_company" employment="{gbp_employment}" placement="{gbp_placement}"
vacancy="{gbp_vacancy}" brand="{gbp_brand}" text="{gbt_text}" />
      <SetVariable name="BPText" type="S"><![CDATA[{{WebServiceResult}}]]></SetVariable>
    </GetBoilerPlateText>
    <DoBranding br_person="" br_vacancy="" br_use="" br_type="" br_id="">
      <if x1="{br_id}" x2="" comparison="!=">
        <SetVariable name="brandid" value="{br_id}"/>
      </if>
  </Library>
</Job>
```

```

<else>
    <SetVariable name="brandid" sql="select BrandID from en_Brand where VisibleID =
GetBrand(:br_person, :br_vacancy,:br_type, :br_use)" />
</else>
<!-- work out the template -->
<SetVariable name="emailbody" sql="select GetNotifyDocuments(:br_type, :brandid, 'R','W',','")"
/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{REFEREEADDRESSNAME}}',
:REFEREEADDRESSNAME)"/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{yourURL}}', :yourURL)"/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{URLRequestCode}}',
:URLRequestCode)"/>
<!-- convert the body-->
<Call gbt_text="{emailbody}" gbp_person="{br_person}" gbp_vacancy="{br_vacancy}"
gbp_brand="{brandid}">GetBoilerPlateText</Call>
<!-- slot body, default signature into html wrap-->
<Call smb_text="{BPText}" smb_wrap="{shtmlwrap}">SetEmailBody</Call>
<!-- <messagedialog>{EmailBodyText}</messagedialog>-->
</DoBranding>
</Library>
<SetVariable name="WebFreq" value="{WPK_G_WEBREFFREQ}"/>
<if x1="{WebFreq}" x2="">
    <SetVariable name="WebFreq" value="1"/>
</if>
<IfSQL condition=" '{WebFreq}' &lt; 0 ">
    <Finish/>
</IfSQL>
<SetVariable name="Weblimit" value="{WPK_G_WEBREFLIMIT}"/>
<if x1="{Weblimit}" x2="">
    <SetVariable name="Weblimit" value="1"/>
</if>
<IfSQL condition=" '{Weblimit}' &lt; 0 ">
    <Finish/>

```

```

</IfSQL>
  <SetVariable type="S" name="CRLF" sql="select string(char(13),char(10))"/>
<SetVariable type="S" name="TAB" sql="select string(char(9))"/>
<SetVariable type="S" name="SMTPHost" value="INSERT SMTP HOST" />
<SetVariable type="S" name="SMTPPort" value="INSERT SMTP PORT" />
<SetVariable type="S" name="SMTPTLS" value="INSERT YES OR NO" />
<SetVariable type="S" name="yourURL" value="INSERT URL TO IQXWeb" />
<ChangeDirectory path="{TempDir}"/>
<!-- Get default signature -->
<StaticFunction service="GetEmailSignature"/>
<SetVariable name="mysignature" type="S"><![CDATA[{{WebServiceResult}}]]></SetVariable>
<!-- get htmlwrap -->
<SetVariable name="emailhtml" type="S" value="Yes"/>
<SetVariable name="shtmlwrap" type="S" sql="select signature from emailsignature where
emailsignatureid='HTML_WRAP'"/>
  <IfSQL condition=""shtmlwrap="">
    <SetVariable name="emailhtml" type="S" value="No"/>
  </IfSQL>
<SetVariable name="Webce" sql=" select classcode from contactclass where classcode =
'{WPK_G_WEBREFCE}'"/>
<SQLQuery>
  <SQLSelect><![CDATA[
    select ReferenceRequest.ReferenceRequestID as refid, person.name as PersonName,
    RefereeAddressName, RefereePosition, ReferenceRequest.CompanyName as
    RefereeCompanyName,
    replace(trim(ReferenceRequest.Email), ' ', '') as RefereeEmail,
    ReferenceRequest.URLRequestCode,
    staff.email as SMTPSender, staff."name" as Staffname, person.personid as pid,
    (if (datediff(day, cast( whensent as date) , current date) / {webfreq}) = {weblimit} then 1 else 0
    endif) as reminder,
    en_brand.email as brandemail, en_brand.visibleid, en_brand.smtppwd,
    ReferenceRequest.brandid as webbrand
    FROM ReferenceRequest key join (staff, person, en_brand)
  ]>

```

```

    where ReferenceRequest.whensent between dateadd (day, -({webfreq}*{weblimit}), current
date) and dateadd(day, -{webfreq}+1, current date)

    and ReferenceRequest.WhenCompleted is null

    and reminder = 1

    and (
        RefereeEmail like '%@%.' and
        RefereeEmail not like '% %' and RefereeEmail not like '%..%' and RefereeEmail not like '%.@%'
and
        RefereeEmail not like '.%' and RefereeEmail not like '%.' and RefereeEmail not like '%@%@%'
    )

    order by ReferenceRequest.WhenCreated

    ]]>
</SQLSelect>
<IfAnyRows>
  <ForEachRow>
    <Block>
      <Call br_person="{pid}" br_use="WEBREFERENCE" br_id="{webbrand}" >DoBranding</Call>

      <SendEmail subject="{visibleid} Reference Request" smtpport="{SMTPPort}"
smtppls="{SMTPTLS}" smtphost="{SMTPHost}" smtpuserid="{brandemail}" html="YES"
smtpsender="{brandemail}" smtppassword="{smtppwd}">

        <EmailRecipients>
          <WriteLn>{RefereeEmail}</WriteLn>
        </EmailRecipients>

        <EmailBody>{EmailBodyText}</EmailBody>

        <EmailPlainBody>
          <WriteLn><![CDATA[{EmailBodyText}]]></WriteLn>
        </EmailPlainBody>

      </SendEmail>

      <SetVariable name="emailFailed" value=""/>
    </Block>
  </ForEachRow>
</IfAnyRows>
<Catch>
  <SetVariable name="emailFailed" value="{Exception}"/>
</Catch>

```

```
</Catch>
<IfSQL condition="{emailFailed}" = "">
  <SQLExec>
    update ReferenceRequest set WhenSent = current timestamp where ReferenceRequestID =
:RefID
  </SQLExec>
  <if x1="{webce}" x2="" comparison="!=">
    <SQLExec><![CDATA[insert into contactevent (contacteventid, staffid, personid,
contactdate, contacttime, classcode,
      who, description, notes) values (uniquekey(:pid), userstaffid , :pid, current date,
current time, '{webce}',
      'A', 'Web Reference Reminder sent to {RefereeAddressName}') ]]></SQLExec>
  </if>
</IfSQL>
</ForEachRow>
</IfAnyRows>
</SQLQuery>
</Job>
```

## The Summary – Send Web Reference Summary.xml.

If a referee has asked for a summary email of their reference and when this job is run it will identify those referees and send a copy to them by email.

NOTE Highlighted lines need to be populated by the System Administrator.

```
<?xml version="1.0"?>
<Job title="Email Reference Requests Summary" autoclose="yes" dateformat="dd/mm/yyyy"
delimiter="," stringquote="&#34;">
  <Library>
    <SetEmailBody smb_text="" smb_wrap="">
      <SetVariable name="Emailbodytext" type="S" value="{smb_text}" />
      <If x1="{emailhtml}" x2="Yes">
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_text, char(13)+char(10),
'&lt;BR/&gt;')" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:smb_wrap, '{{bodytext}},
:EmailBodyText)" />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{signature}}, :mysignature) " />
        <SetVariable name="EmailBodyText" type="S" sql="select replace(:EmailBodyText,
'{{origmessage}}, '')" />
      </If>
    </SetEmailBody>
    <GetBoilerPlateText gbt_text="" gbp_person="" gbp_vacancy="" gbp_company=""
gbp_placement="" gbp_employment="" gbp_brand="">
      <SetVariable name="BPText" value="" />
      <StaticFunction service="CONVERT_BOILERPLATETEXT" person="{gbp_person}"
company="gbp_company" employment="{gbp_employment}" placement="{gbp_placement}"
vacancy="{gbp_vacancy}" brand="{gbp_brand}" text="{gbt_text}" />
      <SetVariable name="BPText" type="S"><![CDATA[{{WebServiceResult}}]></SetVariable>
    </GetBoilerPlateText>
    <DoBranding br_person="" br_vacancy="" br_use="" br_type="" br_id="">
      <!-- locate your brand based on a new notify status -->
      <if x1="{br_id}" x2="" comparison="!=">
        <SetVariable name="brandid" value="{br_id}" />
      </if>
      <else>
        <SetVariable name="brandid" sql="select BrandID from en_Brand where VisibleID =
GetBrand(:br_person, :br_vacancy, :br_type, :br_use)" />
      </else>
    </DoBranding>
  </Library>
</Job>
```



```

</else>
<!-- work out the template -->
<SetVariable name="emailbody" sql="select GetNotifyDocuments(:br_type, :brandid, 'S','W',',' )"
/>
<SetVariable name="emailbody" sql="select replace(:emailbody,
'{{JOB_Email_Question_and_Answers}}, :Email_Question_and_Answers)"/>
<SetVariable name="emailbody" sql="select replace(:emailbody, '{{JOB_RefereeAddressName}},
:RefereeAddressName)"/>
<!-- convert the body -->
<Call gbt_text="{emailbody}" gbp_person="{br_person}" gbp_vacancy="{br_vacancy}"
gbp_brand="{brandid}">GetBoilerPlateText</Call>
<!-- slot body, default signature into html wrap-->
<Call smb_text="{BPText}" smb_wrap="{shtmlwrap}">SetEmailBody</Call>
<!-- <messagedialog>{EmailBodyText}</messagedialog>-->
</DoBranding>
</Library>
<SetVariable type="S" name="CRLF" sql="select string(char(13),char(10))"/>
<SetVariable type="S" name="TAB" sql="select string(char(9))"/>
<SetVariable type="S" name="SMTPHost" value="INSERT SMTP HOST" />
<SetVariable type="S" name="SMTPPort" value="INSERT SMTP PORT" />
<SetVariable type="S" name="SMTPTLS" value="INSERT YES OR NO" />
<SetVariable type="S" name="yourURL" value="INSERT URL TO IQXWeb" />
<ChangeDirectory path="{TempDir}"/>
<!-- Get default signature -->
<StaticFunction service="GetEmailSignature"/>
<SetVariable name="mysignature" type="S"><![CDATA[{{WebServiceResult}}]></SetVariable>
<!-- get htmlwrap -->
<SetVariable name="emailhtml" type="S" value="Yes"/>
<SetVariable name="shtmlwrap" type="S" sql="select signature from emailsignature where
emailsignatureid='HTML_WRAP'"/>
<IfSQL condition="shtmlwrap=''">
<SetVariable name="emailhtml" type="S" value="No"/>
</IfSQL>
<SQLQuery>
<Message>Fetching all completed references</Message>
<SQLSelect><![CDATA[

```

```

    select ReferenceRequest.DepartmentID, ReferenceRequest.ReferenceRequestID,
ReferenceRequest.Email,
    ReferenceRequest.RefereeAddressName, staff.name as Staffname, person.name as PersonName,
person.personid as pid,
    ReferenceRequest.brandid as webbrand,
    en_brand.email as brandemail, en_brand.visibleid , en_brand.smtppwd
FROM ReferenceRequest key join (person, staff, en_brand)
WHERE ReferenceRequest.ReferenceConfirmationRequired = 1 AND
ReferenceRequest.WhenCompleted is not null
    order by ReferenceRequest.WhenCreated
    ]]>
</SQLSelect>
<IfAnyRows>
    <ForEachRow> <!-- for each reference request -->
        <Message>Fetching {PersonName}'s answers</Message>
        <SetVariable type="S" name="Email_Question_and_Answers" value="" />
        <SQLQuery>
            <SQLSelect><![CDATA[
                SELECT isnull( t.LongDescription, t.description) as question,
                (if t.tagtype= 'D' THEN
dateformat(getquestdate(string('R',:DepartmentID),t.tagid,:ReferenceRequestID),'dd/mm/yyyy')
                else getquestanswer(string('R',:DepartmentID),t.tagid,:ReferenceRequestID) endif) as answer
                FROM tag t
                WHERE t.taglocation = 'R{DepartmentID}' order by t.sortorder
            ]]>
            </SQLSelect>
        </IfAnyRows>
        <ForEachRow> <!-- for each question + answer -->
            <SetVariable type="S" name="Email_Question_and_Answers" value="
                {Email_Question_and_Answers}
                &lt;tr&gt;
                    &lt;td&gt; &lt;strong&gt;{question}&lt;/strong&gt; &lt;/td&gt;
                    &lt;td&gt;{answer}&lt;/td&gt;
                &lt;/tr&gt;
                " />
            </ForEachRow>
        </IfAnyRows>

```

```
</SQLQuery>
<Call br_person="{pid}" br_use="WEBREFERENCE" br_id="{webbrand}" >DoBranding</Call>
<Block>
  <Message>Sending Email to {RefereeAddressName} / {Email}</Message>
  <SendEmail subject="{VisibleID} Reference Summary" smtpport="{SMTPPort}"
smtppls="{SMTPTLS}" smtphost="{SMTPHost}" smtpuserid="{brandemail}" html="YES"
smtpsender="{brandemail}" smtppassword="{smtppwd}">
  <EmailRecipients>
    <WriteLn>{Email}</WriteLn>
  </EmailRecipients>
  <EmailBody>{EmailBodyText}</EmailBody>
  <EmailPlainBody>
    <WriteLn><![CDATA[{EmailBodyText}]]></WriteLn>
  </EmailPlainBody>
</SendEmail>
<SetVariable name="emailFailed" value=""/>
<Message>Updating Table</Message>
  <SQLExec><![CDATA[UPDATE ReferenceRequest SET ReferenceConfirmationRequired = 0
WHERE ReferenceRequestID = '{ReferenceRequestID}']]></SQLExec>
</Block>
<Catch>
  <SetVariable name="emailFailed" value="{Exception}"/>
</Catch>
</ForEachRow>
</IfAnyRows>
</SQLQuery>
</Job>
```