

Go to Layout [ "lay\_companiesandcontacts" (organisation) ]

Show/Hide Menubar  
[ Hide ]

Show/Hide Toolbars  
[ Lock; Hide ]

Show All Records

Enter Find Mode [ ]

Exit Script [ ]

Go to Layout [ "lay\_site" (site) ]

Show/Hide Menubar  
[ Hide ]

Show/Hide Toolbars  
[ Lock; Hide ]

Show All Records

Enter Find Mode [ ]

Exit Script [ ]

Go to Layout [ "lay\_development" (development) ]

Show/Hide Menubar  
[ Show ]

Show/Hide Toolbars  
[ Lock; Hide ]

Show All Records

Enter Find Mode [ ]

Exit Script [ ]

**New Window** [ Style: Floating Document; Name: "My Dashboard"; Height: 500; Width: 895; Top: 200; Left: 200; Close: Yes; Minimize: Yes; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_taskdashboard" (jndevelopmenttasks) ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Lock; Hide ]

**Move/Resize Window** [ Name: "My Dashboard"; Current file; Height: 500; Width: 895; Top: 200; Left: 200]

**Enter Find Mode** [ ]

**Set Field** [ site::wildstoneowner; Get ( AccountName ) ]

**Set Field** [ site::FKsitestatus\_id; 1 ]

**Set Error Capture** [ On ]

**Perform Find** [ ]

**Sort Records by Field** [ Descending; site::dateadded ]

**Set Variable** [ \$site; Value:site::PKid ]

**Select Window** [ Name: "Wildstone Development Tracker"; Current file ]

**Show/Hide Menubar**  
[ Hide ]

**Show/Hide Toolbars**  
[ Lock; Hide ]

**Go to Layout** [ "lay\_site" (site) ]

**Enter Find Mode** [ ]

**Set Field** [ site::PKid; \$site ]

**Perform Find** [ ]

**Move/Resize Window** [ Name: "My Dashboard"; Current file; Height: 500; Width: 895; Top: 200; Left: 200]

**Enter Find Mode** [ ]

**Set Field** [ development::wildstonedeveloperorcontact; Get ( AccountName )]

**Set Error Capture** [ On ]

**Perform Find** [ ]

**Sort Records by Field** [ Descending; site::createddate ]

**Set Variable** [ \$devlp; Value:development::PKid ]

**Select Window** [ Name: "Wildstone Development Tracker"; Current file ]

**Show/Hide Menubar**  
[ Hide ]

**Show/Hide Toolbars**  
[ Lock; Hide ]

**Go to Layout** [ "lay\_development" (development) ]

**Enter Find Mode** [ ]

**Set Field** [ development::PKid; \$devlp ]

**Perform Find** [ ]



**Move/Resize Window** [ Name: "My Dashboard"; Current file; Height: 500; Width: 895; Top: 200; Left: 200]

**Enter Find Mode** [ ]

**Set Field** [ development::wildstoneplanner; Get ( AccountName )]

**Set Error Capture** [ On ]

**Perform Find** [ ]

**Sort Records by Field** [ Descending; site::createddate ]

**Move/Resize Window** [ Name: "My Dashboard"; Current file; Height: 500; Width: 895; Top: 200; Left: 200]

**Enter Find Mode** [ ]

**Set Field** [ indevelopmenttasks::assignedto; Get ( AccountName ) ]

**Set Field** [ indevelopmenttasks::complete; "Incomplete" ]

**Set Error Capture** [ On ]

**Perform Find** [ ]

**Move/Resize Window** [ Name: "My Dashboard"; Current file; Height: 500; Width: 895; Top: 200; Left: 200]

**Enter Find Mode** [ ]

**Set Field** [ indevelopmenttasks::createdby; Get ( AccountName )]

**Set Field** [ indevelopmenttasks::complete; "Incomplete" ]

**Set Error Capture** [ On ]

**Perform Find** [ ]

**Set Variable** [ \$dev; Value:jndevelopmenttasks::FKdevelopment\_id ]

**Select Window** [ Name: "Wildstone Development Tracker"; Current file ]

**Show/Hide Menubar**  
[ Show ]

**Show/Hide Toolbars**  
[ Show ]

**Go to Layout** [ "lay\_development" (development) ]

**Enter Find Mode** [ ]

**Set Field** [ development::PKid; \$dev ]

**Perform Find** [ ]

**Go to Object** [ Object Name: "prtTasksAndContacts"; Repetition: 1 ]

**Show Custom Dialog** [ Title: "Send Task Complete Notification"; Message: "Do you want to notify the assignor of this task that it has been completed?"; Default Button: "Yes", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Send Mail** [ Send via E-mail Client; To: useraccounts\_developmenttaskcreator::email; Subject: "Task Complete - " & development::name & " at " & site::name; Message: "The following task has been completed" & ¶ & ¶ &

"Task Type:" & tasktype::action & ¶ & ¶ &

"Task:" & jndevelopmenttasks::notes & ¶ & ¶ &

"Completed:"& jndevelopmenttasks::completed & ¶ & ¶ &

"By:" & jndevelopmenttasks::assignedto ]

**End If**

**Perform Script** [ "My Dashboard Tasks Layout Enter"]

## Sites Layout: Update Site Owner / Agent Name & refresh Location Map

---

*#Used as a script trigger on site layout load to refresh the site owner name and site agent name in case they have been updated at company level. Required due to the using the type ahead look up to get owner and agent details.*

**Set Field** [ site::siteownername; organisation\_site::name ]

**Set Field** [ site::siteagentname; organisation\_agent::name ]

**Set Web Viewer** [ Object Name: "location"; Action: Reset ]

**Show Custom Dialog** [ Title: "Duplicate Sites"; Message: "It's important that duplicate sites are not added to the database." ]

& ¶ & ¶ & "Prior to entering a new site, perform a search on this form or in the 'Site Mapping' section to see if it already exists. Otherwise, continue by clicking 'Add'"; Default Button: "Search", Commit: "No"; Button 2: "Add", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Enter Find Mode** [ ]

**Go to Object** [ Object Name: "SiteNameSite"; Repetition: 1 ]

**Exit Script** [ ]

**Else If** [ Get ( LastMessageChoice ) = 2 ]

**New Record/Request**

**Go to Object** [ Object Name: "SiteOwnerSite"; Repetition: 1 ]

**End If**

New Record/Request

Go to Field [ [site::FKorganisation\\_id\\_siteowner](#) ]



## Sites Layout: Menu: Sites Menu: View Other Sites With Same Site Owner

---

**Set Variable** [ \$owner; Value:site::FKorganisation\_id\_siteowner ]

**Go to Layout** [ "lay\_site" (site) ]

**Enter Find Mode** [ ]

**Set Field** [ site::FKorganisation\_id\_siteowner; \$owner ]

**Perform Find** [ ]

**If** [ Get ( FoundCount ) = 1 ]

**Show Custom Dialog** [ Title: "Developments at this site"; Message: "The site owner doesn't have any other sites listed"; Default Button: "OK", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Exit Script** [ ]

**End If**

**Else** [ ]

**If** [ Get ( FoundCount ) > 1 ]

**Show Custom Dialog** [ Title: "Developments at this site"; Message: "The site owner has "&(Get (FoundCount) - 1)&" other sites listed. Do you wish to view them now?"; Default Button: "View", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Sort Records by Field** [ Ascending; site::FKsitestatus\_id ]

**Else** [ ]

**Show All Records**

**End If**

**End If**

**End If**

Go to Layout [ original layout ]

Show All Records

Close Popover

Enter Find Mode [ ]



**Show All Records**

**Set Error Capture** [ On ]

**New Window** [ Style: Document; Name: "Sites List Report"; Height: 800; Width: 800; Top: 50; Left: 50; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

**Go to Layout** [ "sites report" (site) ]

**Set Variable** [ \$ID; Value:site::PKid ]

**Go to Layout** [ "lay\_development" (development) ]

**New Record/Request**

**Set Field** [ development::FKsite\_id; \$ID ]

**Go to Object** [ Object Name: "DevelopmentType"; Repetition: 1 ]

**Set Error Capture** [ On ]

**Set Variable** [ \$siteid; Value:site::PKid ]

**Go to Layout** [ "lay\_development" (development) ]

**Show All Records**

**Enter Find Mode** [ ]

**Set Field** [ development::FKsite\_id; \$siteid ]

**Perform Find** [ ]

**Set Variable** [ \$found; Value:Get ( FoundCount ) ]

**Go to Layout** [ original layout ]

**If** [ \$found = 0 ]

**Show Custom Dialog** [ Title: "Developments at this site"; Message: "No developments were found at this site"; Default Button: "Close", Commit: "No" ]

**Exit Script** [ ]

**End If**

**Show Custom Dialog** [ Title: "Developments at this site"; Message: Case ( (\$found = 1) ; \$found &" development was found at this site. Do you wish to view it now?"; (\$found > 1);\$found &" developments were found at this site. Do you wish to view them now?"); Default Button: "View", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Go to Layout** [ "lay\_development" (development) ]

**Else** [ ]

**Go to Layout** [ original layout ]

**End If**

Go to Layout [ "lay\_development" (development) ]

Show All Records

Enter Find Mode [ ]

Go to Layout [ "lay\_companiesandcontacts" (organisation) ]

Show All Records

Enter Find Mode [ ]



**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "New Site Owner"; Top: 100; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_companiesandcontacts" (organisation) ]

**New Record/Request**

**Set Field** [ organisation::type; "Site Owner / Client" ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "New Site Agent"; Top: 100; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_companiesandcontacts" (organisation) ]

**New Record/Request**

**Set Field** [ organisation::type; "Agent" ]

## Sites Layout: Site Tab: Not used: Set Site Name

---

*#Decided not to use this*

**If** [ IsEmpty ( site::name ) ]

**Set Field** [ site::name; address\_site::line1 ]

**Else If** [ not IsEmpty ( site::name ) ]

**Show Custom Dialog** [ Title: "Modify Site Name"; Message: "Do you want to update the Site Name to show this change?"; Default  
        Button: "Yes", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Set Field** [ site::name; address\_site::line1 ]

**End If**

**End If**

**Set Variable** [ \$path; Value:"file:" & Get ( DesktopPath ) & site::name & ".pdf" ]

**Set Error Capture** [ On ]

**Allow User Abort** [ Off ]

**Go to Layout** [ original layout ]

**Print Setup** [ Orientation: Landscape; Paper size: 8.26" x 11.69"; Scaling: 55% ]  
[ Restore ]

**Save Records as PDF** [ File Name: "\$path"; Create folders:No; Current record ]

[ *Document* - ]

[ *Pages* - Number Pages From: 1; Include: All pages ]

[ *Security* - Printing: High Resolution; Editing: Any except extracting pages; Enable copying; Enable Screen Reader ]

[ *Initial View* - Show: Pages Panel and Page; Page Layout: Single Page; Magnification: 100% ]

[ Restore ]

**Go to Layout** [ original layout ]

**Show Custom Dialog** [ Title: "Get Coordinates"; Message: "This button fetches the Longitude and Latitude coordinates based on the address input. Results will not always be accurate (i.e. if it's a 'land at' address) or it may not find a result if address details are limited.

¶¶¶Once they have been set, check the 'Location' tab (to the right) to see if the marker has been placed in the correct place.

¶¶¶If these are not accurate, you can manually add the exact coordinates by following these steps;

¶¶ - Open Google Maps in your browser

¶ - Zoom in close to the site

¶ - Click on the location of the site

¶- The coordinates will be displayed below the address under the search bar

¶ - Copy and paste the coordinates into the fields, ignoring the comma separator"; Default Button: "OK", Commit: "No" ]

## Sites Layout: Site Tab: Get Coordinates for this Address Site

---

**If** [ not IsEmpty(address\_site::longitude) ]

**Show Custom Dialog** [ Title: "Coordinates Already Exist"; Message: "Coordinates already exist for this site and will be overwritten.";  
Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 2 ]

**Exit Script** [ ]

**End If**

**End If**

**Go to Layout** [ original layout ]

**Insert from URL** [ site::inserturl; "http://maps.googleapis.com/maps/api/geocode/json?address=" & Substitute(address\_site::line1; ""," ") &  
" " & Substitute(address\_site::FKtownsandcities\_id; ""," ")& " " & Substitute(address\_site::FKcounties\_PKid; ""," ")& " " & Substitute  
(address\_site::country; ""," ")& " " & Substitute(address\_site::postcode; ""," ") & "&output=csv&key=" & settings::  
gGoogle\_API\_Key&"&sensor=true" ]  
[ Select; No dialog ]

**Commit Records/Requests**

[ No dialog ]

**Set Variable** [ \$response\_from\_google; Value:site::inserturl ]

**Set Variable** [ \$LongLatChunk; Value:Extract ( \$Response\_From\_Google ; "\"location\" : {\" ; "\"location\_type\""" ) ]

**Set Field** [ address\_site::latitude; Filter(Trim(Extract ( \$LongLatChunk ; "\"lat\" : \" ; \" , \" )); "0123456789-." ) ]

**Set Field** [ address\_site::longitude; Filter(Trim(Extract ( \$LongLatChunk ; "\"lng\" : \" ; \" , \" )); "0123456789-." ) ]

**Set Web Viewer** [ Object Name: "GPSWebService"; URL: "" ]

**Set Error Capture** [ On ]

**Freeze Window**

*#Copy the Account Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_townsandcities" (townsandcities) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ townsandcities::Name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Account record*

**Set Variable** [ \$id; Value:townsandcities::PKid ]

**Go to Layout** [ original layout ]

*#Link the Invoice record to the Account record*

**Set Field** [ address site::FKtownsandcities id; \$id ]

*#Normally you would navigate to the next desired field, say the invoice date or the first line item.*

// **Go to Object** [ ]

**Set Error Capture** [ On ]

**Freeze Window**

*#Copy the Account Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_counties" (counties) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ counties::name; "=" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Account record*

**Set Variable** [ \$id; Value:counties::PKid ]

**Go to Layout** [ original layout ]

*#Link the Invoice record to the Account record*

**Set Field** [ address site::FKcounties PKid; \$id ]

*#Normally you would navigate to the next desired field, say the invoice date or the first line item.*

// **Go to Object** [ ]



**Set Error Capture** [ On ]

**Freeze Window**

*#Copy the Local Authority Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_organisation LA" (organisation) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ organisation::name; "=" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Organisation record*

**Set Variable** [ \$id; Value:organisation::PKid ]

**Go to Layout** [ original layout ]

*#Link the Site record to the Local Authority record*

**Set Field** [ site::FKorganisation id localauthority; \$id ]

*#Normally you would navigate to the next desired field*

**// Go to Object** [ ]

Go to Related Record [ From table: "development"; Using layout: "lay\_development" (development) ]

**Sites Layout: Site Contacts Tab: Delete Site Agent Portal Record**

---

**Show Custom Dialog** [ Title: "Remove Site Agent Contact "; Message: "Are you sure you want to remove this agent contact from this site?";  
Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**  
        [ No dialog ]

**End If**

**Sites Layout: Site Contacts Tab: Delete Site Owner Portal Record**

---

**Show Custom Dialog** [ Title: "Remove Site Owner Contact "; Message: "Are you sure you want to remove this contact from this site?";  
Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

**If** [ IsEmpty ( site::dialog agent ) ]

**Exit Script** [ ]

**Else If** [ IsEmpty(site::FKorganisation\_id\_siteagent) ]

**Go to Field** [ site::FKorganisation\_id\_siteagent ]

**Exit Script** [ ]

**Else If** [ IsEmpty(jnsiteagentcontacts::FKcontact\_id) ]

**Set Variable** [ \$Agent; Value:site::FKorganisation\_id\_siteagent ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

[ Resize to Fit ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Hide ]

**New Record/Request**

**Set Field** [ contact::FKorganisation\_id; \$Agent ]

**Go to Field** [ contact::salutation ]

**End If**

**Exit Script** [ ]

**Set Variable** [ \$Agent; Value:site::FKorganisation\_id\_siteagent ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

[ Resize to Fit ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Hide ]

**New Record/Request**

**Set Field** [ contact::FKorganisation\_id; \$Agent ]

**Go to Field** [ contact::salutation ]

**Exit Script** [ ]

**If** [ IsEmpty ( site::dialog owner ) ]

**Exit Script** [ ]

**Else If** [ IsEmpty(site::FKorganisation\_id\_siteowner) ]

**Go to Field** [ site::FKorganisation\_id\_siteowner ]

**Exit Script** [ ]

**Else If** [ IsEmpty(jnsiteownercontacts::FKcontact\_id) ]

**Set Variable** [ \$Owner; Value:site::FKorganisation\_id\_siteowner ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

[ Resize to Fit ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Hide ]

**New Record/Request**

**Set Field** [ contact::FKorganisation\_id; \$Owner ]

**Go to Field** [ contact::salutation ]

**End If**

**Exit Script** [ ]

## Sites Layout: Site Contacts Tab: Add Site Owner Contacts Button

---

**Set Variable** [ \$Owner; Value:site::FKorganisation\_id\_siteowner ]

### Commit Records/Requests

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

### Adjust Window

[ Resize to Fit ]

### Show/Hide Menubar

[ Hide ]

### Show/Hide Toolbars

[ Hide ]

### New Record/Request

**Set Field** [ contact::FKorganisation\_id; \$Owner ]

**Go to Field** [ contact::salutation ]

**Exit Script** [ ]



**Show Custom Dialog** [ Title: "Add Photos"; Message: "Use the 'Camera' button to import multiple photos for this site (the photos should be added to a single folder first).

¶¶Alternatively, drag and drop single photos to the white boxes below.

¶¶Once you've added your photos select the 'Main Photo' you wanted to appear as a reference throughout the database. This can be changed at any time."; Default Button: "OK", Commit: "No" ]

## Sites Layout: Site Photos Tab: Not Used: Import Multiple Photos

---

**Allow User Abort** [ Off ]

**Set Error Capture** [ On ]

**Freeze Window**

**Set Variable** [ \$siteid; Value:site::PKid ]

**Go to Layout** [ "tbl\_sitephotos\_scriptstep" (sitephotos) ]

**Import Records** [ File Type: Picture and movie files; Target: "sitephotos"; Method: Add; Character Set: "Mac Roman"; Field Mapping:

Source field 1 import to sitephotos::image(Do Auto-Enter);

Source field 2 import to sitephotos::filename;

Source field 3 import to sitephotos::filepath;

Source field 4 import to sitephotos::imagethumbnail;

sitephotos::PKid(Do Auto-Enter);

sitephotos::comment(Do Auto-Enter);

sitephotos::createdby(Do Auto-Enter);

sitephotos::modifiedby(Do Auto-Enter);

sitephotos::modifiedtimestamp(Do Auto-Enter);

sitephotos::createddate(Do Auto-Enter); ]

[ Data contains column names ]

**If** [ (Get(LastMessageChoice) = 1) ]

**Go to Layout** [ original layout ]

**Exit Script** [ ]

**Else** [ ]

**End If**

**Replace Field Contents** [ sitephotos::FKsite\_id; Replace with calculation: \$siteid ]

[ No dialog ]

**Go to Layout** [ original layout ]

**Allow User Abort** [ Off ]

**Set Error Capture** [ On ]

**Freeze Window**

**Set Variable** [ \$siteid; Value:site::PKid ]

**Go to Layout** [ "tbl\_sitephotos\_scriptstep" (sitephotos) ]

**Import Records** [ File Type: Picture and movie files; Target: "sitephotos"; Method: Add; Character Set: "Mac Roman"; Field Mapping:

Source field 1 import to sitephotos::image(Do Auto-Enter);

Source field 2 import to sitephotos::filename;

Source field 3 import to sitephotos::filepath;

Source field 4 import to sitephotos::imagethumbnail;

sitephotos::PKid(Do Auto-Enter);

sitephotos::comment(Do Auto-Enter);

sitephotos::createdby(Do Auto-Enter);

sitephotos::modifiedby(Do Auto-Enter);

sitephotos::modifiedtimestamp(Do Auto-Enter);

sitephotos::createddate(Do Auto-Enter); ]

[ Data contains column names ]

**If** [ (Get(LastMessageChoice) = 1) ]

**Go to Layout** [ original layout ]

**Exit Script** [ ]

**Else** [ ]

**End If**

**Replace Field Contents** [ sitephotos::FKsite\_id; Replace with calculation: \$siteid ]

[ No dialog ]

**Go to Layout** [ original layout ]

**If** [ not sitephotos::PKid ]

**Exit Script** [ ]

**End If**

**Set Field** [ site::FKsitephoto\_id; Let (

    id = sitephotos::PKid ;

    Case ( id = site::FKsitephoto\_id ; "" ; id )

)]

**Commit Records/Requests**

[ Skip data entry validation; No dialog ]

**Refresh Window**

[ Flush cached join results; Flush cached external data ]

## Sites Layout: Site Photos Tab: Delete Site Photo Portal Record

---

**Show Custom Dialog** [ Title: "Delete Photo"; Message: "Are you sure you want to delete this photo?"; Default Button: "Yes", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**Set Variable** [ \$photoid; Value:sitephotos::PKid ]

**If** [ Get (LastMessageChoice) = 2 ]

**Exit Script** [ ]

**Else** [ ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

**If** [ site::FKsitephoto\_id = \$photoid ]

**Clear** [ site::FKsitephoto\_id ]

[ Select ]

**End If**

**End If**

**Allow User Abort** [ Off ]

**Set Error Capture** [ On ]

**Set Field** [ [sitephotos::g\\_largephoto](#); GetField ( Get ( ScriptParameter ) ) ]

**New Window** [ Style: Dialog; Name: "Site Photo " & site::name; Height: 665; Width: 963; Top: 0; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_large photos" (sitephotos) ]

**Show/Hide Toolbars**  
[ Hide ]

## Sites Layout: Title Plan Tab: Open Title Plan

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (site::landregistymap));"remote:."; "" ) ]

**Export Field Contents** [ site::landregistymap; "\$temp"; Automatically open; Create folders:No ]

**Sites Layout: Title Plan Tab: Delete Title Plan**

---

**Show Custom Dialog** [ Title: "Delete Title Plan"; Message: "Are you sure you want to delete this Title Plan?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ site::landregistymap ]  
        [ Select ]

**End If**



Go to Field [ [site::landregistymap](#) ]

Insert PDF [ ]

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (site::landregistrydoc));"remote:."; "" ) ]

**Export Field Contents** [ site::landregistrydoc; "\$temp"; Automatically open; Create folders:No ]

Go to Field [ [site::landregistrydoc](#) ]

Insert PDF [ ]

**Show Custom Dialog** [ Title: "Delete Register of Title"; Message: "Are you sure you want to delete this Register of Title?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ site::landregistrydoc ]  
        [ Select ]

**End If**

Go to Layout [ "lay\_site" (site) ]

Show All Records

Enter Find Mode [ ]

Go to Layout [ "lay\_development" (development) ]

Show All Records

Enter Find Mode [ ]

## Mapping Layout: Menu: Go to Layout

---

If [ Get(ScriptParameter)="Solution 2 Data Entry View" ]

**Go to Layout** [ "Solution 2 Data Entry View" (site) ]

End If

If [ Get(ScriptParameter)="Solution 2 Map View" ]

**Go to Layout** [ "Solution 2 Map View" (settings) ]

    If [ /\*not IsEmpty(Settings::gRequested\_URL)\*/ ]

**Set Web Viewer** [ Object Name: "GoogleWebViewer"; URL: /\*Settings::gRequested\_URL\*/ ]

    End If

End If

If [ Get(ScriptParameter)="Solution 2 Settings" ]

**Go to Layout** [ "Solution 2 Settings" (settings) ]

End If

**Adjust Window**

    [ Resize to Fit ]

## Mapping Layout: Active Solution: Generate Map: Local Authority Name Type Ahead - Mapping

---

*#Script is used for type ahead functionality to allow user to select a local authority in the mapping filters*

**Set Error Capture** [ On ]

**Freeze Window**

**If** [ not IsEmpty ( filterfields::gLocalAuthorityLookup ) ]

*#Copy the Local Authority Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_organisation LA" (organisation) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name in quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ organisation::name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Organisation record*

**Set Variable** [ \$id; Value:organisation::PKid ]

**Go to Layout** [ original layout ]

*#Link the Site record to the Local Authority record*

**Set Field** [ filterfields::gLocalAuthority; \$id ]

*#Normally you would navigate to the next desired field*

*// Go to Object* [ ]

**Else** [ ]

**If** [ IsEmpty ( filterfields::gLocalAuthorityLookup ) ]

**Clear** [ filterfields::gLocalAuthority ]  
[ Select ]

**End If**

**End If**



## Mapping Layout: Active Solution: Generate Map: Site Owner Name Type Ahead - Mapping

---

*#The script gives users type ahead functionality when selecting a Site Owner filter on the mapping layout*

**Set Error Capture** [ On ]

**Freeze Window**

**If** [ not IsEmpty(filterfields::gOwnerLookup) ]

*#Copy the Site Owner Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_organisation LA" (organisation) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ organisation::name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Organisation record*

**Set Variable** [ \$id; Value:organisation::PKid ]

**Go to Layout** [ original layout ]

*#Link the Site record to the Site Owner record*

**Set Field** [ filterfields::gOwner; \$id ]

*#Navigates to next field*

*// Go to Object* [ Object Name: "btnFilterSites"; Repetition: 1 ]

**Else** [ ]

**If** [ IsEmpty(filterfields::gOwnerLookup) ]

**Clear** [ filterfields::gOwner ]  
[ Select ]

**End If**

**End If**

## Mapping Layout: Active Solution: Generate Map: Site Agent Name Type Ahead - Mapping

---

*#The script gives users type ahead functionality when selecting a Site Agent in the filter fields on the mapping layout*

**Set Error Capture** [ On ]

**Freeze Window**

**If** [ not IsEmpty(filterfields::gAgentLookup) ]

*#Copy the Site Owner Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_organisation LA" (organisation) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ organisation::name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Organisation record*

**Set Variable** [ \$id; Value:organisation::PKid ]

**Go to Layout** [ original layout ]

*#Link the Site record to the Site Owner record*

**Set Field** [ filterfields::gAgent; \$id ]

*#navigate to the next desired field*

*// Go to Object* [ ]

**Else** [ ]

**If** [ IsEmpty(filterfields::gAgentLookup) ]

**Clear** [ filterfields::gAgent ]  
[ Select ]

**End If**

**End If**

## Mapping Layout: Active Solution: Generate Map: Town Name Type Ahead - Mapping

---

*#The script gives users type ahead functionality when selecting a town/city in the filters on the mapping layout*

**Set Error Capture** [ On ]

**Freeze Window**

**If** [ not IsEmpty ( filterfields::gCityLookup ) ]

*#Copy the Town / City Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_townsandcities" (townsandcities) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name in quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ townsandcities::Name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Account record*

**Set Variable** [ \$id; Value:townsandcities::PKid ]

**Go to Layout** [ original layout ]

*#Link the records*

**Set Field** [ filterfields::gCity; \$id ]

*#Normally you would navigate to the next desired field*

*// Go to Object* [ ]

**Else** [ ]

**If** [ IsEmpty ( filterfields::gCityLookup ) ]

**Clear** [ filterfields::gCity ]  
[ Select ]

**End If**

**End If**

## Mapping Layout: Active Solution: Generate Map: Clear Filter Fields

---

*#Script clears all filter fields use on the mapping layout*

**Set Field** [ filterfields::gOwner; "" ]

**Set Field** [ filterfields::gOwnerLookup; "" ]

**Set Field** [ filterfields::gAgent; "" ]

**Set Field** [ filterfields::gAgentLookup; "" ]

**Set Field** [ filterfields::gProject; "" ]

**Set Field** [ filterfields::gDeveloperCompany; "" ]

**Set Field** [ filterfields::gDeveloperContact; "" ]

**Set Field** [ filterfields::gSiteStatus; "" ]

**Set Field** [ filterfields::gDevelopmentStatus; "" ]

**Set Field** [ filterfields::gPlanningScore; "" ]

**Set Field** [ filterfields::gApplicationStatus; "" ]

**Set Field** [ filterfields::gPanelType; "" ]

**Set Field** [ filterfields::gPanelSize; "" ]

**Set Field** [ filterfields::gCity; "" ]

**Set Field** [ filterfields::gCityLookup; "" ]

**Set Field** [ filterfields::gLocalAuthority; "" ]

**Set Field** [ filterfields::gLocalAuthorityLookup; "" ]

**Set Field** [ filterfields::gDealType; "" ]

**Set Field** [ filterfields::gDevelopmentType; "" ]

**Go to Layout** [ "Solution 2 Data Entry View" (site) ]

**Show All Records**

## Mapping Layout: Active Solution: Generate Map: FindSet

---

**If** [ not Count (filterfields::gOwner ; filterfields::gProject ; filterfields::gDeveloperCompany ; filterfields::gDeveloperContact ; filterfields::gSiteStatus ; filterfields::gDevelopmentStatus ; filterfields::gPlanningScore ; filterfields::gApplicationStatus ; filterfields::gPanelType ; filterfields::gPanelSize ; filterfields::gCity ; filterfields::gLocalAuthority ; filterfields::gDealType ;filterfields::gAgent;filterfields::gDevelopmentType) ]

**Exit Script** [ ]

**End If**

**Enter Find Mode** [ ]

**Set Field** [ site::FKorganisation id siteowner; filterfields::gOwner ]

**Set Field** [ site::FKorganisation id siteagent; filterfields::gAgent ]

**Set Field** [ indevelopmentprojects::FKproject id; filterfields::gProject ]

**Set Field** [ development::FKorganisation id developercompany; filterfields::gDeveloperCompany ]

**Set Field** [ development::wildstonedeveloperorcontact; filterfields::gDeveloperContact ]

**Set Field** [ development::FKdealttype id; filterfields::gDealType ]

**Set Field** [ development::FKdevelopmenttype id; filterfields::gDevelopmentType ]

**Set Field** [ site::FKsitestatus id; filterfields::gSiteStatus ]

**Set Field** [ development::FKdevelopment statusid; filterfields::gDevelopmentStatus ]

**Set Field** [ development::planningscore; filterfields::gPlanningScore ]

**Set Field** [ development::FKstatus ID advertapplication; filterfields::gApplicationStatus ]

**Set Field** [ developmentdetail::FKpaneltype id; filterfields::gPanelType ]

**Set Field** [ developmentdetail::FKpanelsize id; filterfields::gPanelSize ]

**Set Field** [ address site::FKtownsandcities id; filterfields::gCity ]

**Set Field** [ site::FKorganisation id localauthority; filterfields::gLocalAuthority ]

**Set Error Capture** [ On ]

**Perform Find** [ ]

```

If [ IsEmpty(address_site::longitude) ]
Else If [ IsEmpty(address_site::latitude) ]
    Exit Script [ ]
Show Custom Dialog [ Title: "Empty Longitude of Latitude"; Message: "One of the sites is missing a Longitude or Latitude coordinate.
    These must be added before generating the map, otherwise no sites will show on the map."; Default Button: "OK", Commit: "Yes" ]

End If
Set Variable [ $DisplayMap; Value:Get(ScriptParameter) ]
If [ $DisplayMap ="In Database" ]
    #Skip Dialog Box
Else [ ]
    Show Custom Dialog [ Title: "View in a Browser or in FileMaker"; Message: "Would you like to view this data in a web browser or
        within the database?"; Default Button: "Browser", Commit: "Yes"; Button 2: "Database", Commit: "No"; Button 3: "Cancel", Commit:
        "No" ]
    Set Variable [ $DisplayMap; Value:Case(
        Get(LastMessageChoice)=1; "In Web Browser";
        Get(LastMessageChoice)=2; "In Database";
        Get(LastMessageChoice)=3; "Cancel"
    ) ]
End If
If [ $DisplayMap="Cancel" ]
    #'Cancel' selected.
    Halt Script
End If
Perform Script [ "Check for '.htm' in File Name" ]
Go to Layout [ "Solution 2 Data Entry View" (site) ]
#set up the $json header.
Set Variable [ $json; Value:"var data =[" ]
Go to Record/Request/Page
    [ First ]
Loop [ Flush: Always ]
    #Make sure the longitude and latitude are numbers in the appropriate range.
    If [ not IsEmpty(address_site::latitude) and not IsEmpty(address_site::longitude) ]
        If [ address_site::longitude <180 and address_site::latitude <90

            //I switched this to test for less than 180 for longitude (360 around the world, split in half E/W -/+) and less than 90 for Latitude
            ]

        If [ not IsEmpty(address_site::line1) ]
            #Put any additional information (aside from the label field) in here that you want displayed on the data point.
            Set Variable [ $infoURL; Value:"<a href=\"\"&"FMP://\" &Get(SystemIPAddress)&"/\"& Get ( FileName )& "?script=Open-
                Detail-Map&param=" & address_site::PKid&"\"></a>" ]
            Set Variable [ $info; Value:Substitute(
                Trim(address_site::line1)&" ", "&Trim(address_site::FKtownsandcities_id)&" "&Trim(address_site::FKcounties_PKid)
                &" "&Trim(address_site::postcode)&" "&Trim(address_site::country)&$infoURL;
                ["¶";" "];
                [" ";""];
                [" ";""];
                ["\\"", " ") ]
            #create JSON record with longitude, latitude, label and info
            Set Variable [ $json; Value:$json

                &"{"

                &"longitude\": " & address_site::longitude

```

```

& ", \"latitude\": " & address_site::latitude
& ", \"label\": \"\"& address_site::Address_Label
& "\", \"info\": \"\"& $info
& "\", " ]

```

**Set Variable** [ \$locCount; Value:\$locCount+1 ]

**End If**

**End If**

**End If**

**Go to Record/Request/Page**

[ Next; Exit after last ]

**End Loop**

**If** [ \$locCount≤0 ]

**Show Custom Dialog** [ Title: "No Sites Found"; Message: "No sites were found meeting the filter criteria. If you are sure there are sites matching this criteria make sure they all have valid Longitude and Latitude co-ordinates. Missing or incorrect values can stop the map generating properly."; Default Button: "OK", Commit: "Yes" ]

**Else** [ ]

**Set Variable** [ \$json; Value:Substitute(Substitute(Left(\$json; Length(\$json)-1) & "];"; "[]"; "" ); Char ( 9 ); " " ]

**Set Field** [ settings::g\_tmp; "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<title>" & settings::gMap\_Title& "</title>

<script src="http://maps.googleapis.com/maps/api/js?key="&settings::gGoogle\_API\_Key&"&sensor=false" type="text/javascript" charset="utf-8"></script>"&

Case(settings::gInclude\_Javascript="Reference External Link"; "<script src="\"&settings::gURL\_Marketcluster\_Javascript&" charset="utf-8"></script>";

settings::gInclude\_Javascript="Include"; "<script type="text/javascript">" & settings::gMarketcluster\_Javascript & "</script>" )

&"<script type="text/javascript">

" & \$json & "

```

var infowindow;
function initialize() {

```

```

    var myOptions = {

```

```

        center: new google.maps.LatLng("&settings::gCentre_latitude&","&settings::gCentre_longitude&"),
        zoom: "&settings::gMap_Zoom_Level&",

```

```

        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

```

```

    var map = new google.maps.Map(document.getElementById("map"),
        myOptions);

```

```

    infowindow = new google.maps.InfoWindow({
        content: "...",
        maxWidth:50
    });

```

```

    var markers = [];

```

```

    for (var i = 0; i < data.length; ++i) {

```

```

        var latlng = new google.maps.LatLng(data[i].latitude, data[i].longitude);

```

```

        var marker = new google.maps.Marker({position: latlng, map: map, title:data[i].label, info:data[i].info});

```

```

    google.maps.event.addListener(marker, 'click', function () {

```

```

        var titletag = document.createElement("H3");

```

```

        var contenttag = document.createElement("div");

```

```

        var infotag = document.createElement('p');

```

```

        titletag.innerHTML = this.title;

```

```

        infotag.innerHTML = this.info;

```

```

        contenttag.appendChild(titletag);

```

```

        contenttag.appendChild(infotag);

```

```

        infowindow.setContent(contenttag);
    });

```

```

        infowindow.setPosition(this.position);
        infowindow.open(map, this);
        map.setCenter(this);
    });

    markers.push(marker);
}
var markerCluster = new MarkerClusterer(map, markers);
}
</script>
</head>
<body onload="initialize()" onunload="GUnload()">
    <h3> "&settings::gMap_Title&" </h3>
    <h4>Generated by " & Get ( AccountName ) & " on " & Get ( CurrentTimestamp ) & " </h4>
    <p>The map shows " & $locCount & " sites meeting your search criteria</p>
    <div id="map" style="width:"&settings::gMap_Width&"px;height:"&settings::gMap_Height&"px;"></div>
</body>
</html>" ]

```

**Set Variable** [ \$FileName; Value:settings::gFile\_Name ]

**Set Variable** [ \$Path; Value:Case(

```

    settings::gSave_File_Location="Desktop Path"; Get(DesktopPath);
    settings::gSave_File_Location="Temporary Path"; Get(TemporaryPath);
    settings::gSave_File_Location="Same Directory as Database File"; "";
    settings::gSave_File_Location="FileMaker Folder"; Get ( FileMakerPath );
    settings::gSave_File_Location="Documents Folder"; Get ( DocumentsPath )
)]

```

**Set Variable** [ \$tmpPath; Value:\$Path&\$FileName ]

*#The New Window routine allows us to isolate a single record for export routine AND maintain the Found Set.*

**New Window** [ Style: Document; Name: "Export"; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

**Show All Records**

**Omit Record**

**Show Omitted Only**

**If** [ \$DisplayMap ="In Web Browser" ]

*#Open in a Browser selected.*

**Export Records** [ File Name: "\$tmpPath"; Create folders:No; Character Set: "Unicode (UTF-8)"; Field Order: settings::g tmp ]  
[ No dialog ]

**Set Variable** [ \$localPath; Value:Case(

```

    Abs(Get(SystemPlatform))=1; "file://" & Substitute($tmpPath ; " " ; "%20" );
    Abs(Get(SystemPlatform))=2; "file://" & Substitute($tmpPath ; " " ; "%20" ))]

```

**Close Window** [ Name: "Export"; Current file ]

**Open URL** [ \$localPath ]  
[ No dialog ]

**Halt Script**

**Else If** [ \$DisplayMap ="In Database" ]

*#View in FileMaker Web Viewer selected.*

**Export Records** [ File Name: "\$tmpPath"; Create folders:No; Character Set: "Unicode (UTF-8)"; Field Order: settings::g tmp ]  
[ No dialog ]

*#Exporting a single record allows you to select UTF-8 output which turns out to be crucial to the Javascript interpreting the file correctly.*

**Close Window** [ Name: "Export"; Current file ]

**Go to Layout** [ "Solution 2 Map View" (settings) ]

**Adjust Window**  
[ Resize to Fit ]

*#FileMaker's path routine for a Web Viewer is a little different than the web browsers. It can't handle spaces and needs the 'file:/' prefix.*



**Set Variable** [ \$localPath; Value:Case(

Abs(Get(SystemPlatform))=1; "file:/" & Substitute(\$tmpPath ; " " ; "%20" );

Abs(Get(SystemPlatform))=2; "file://" & Substitute(\$tmpPath ; " " ; "%20" )]

**Set Field** [ settings::gRequested\_URL; \$localPath ]

**Set Web Viewer** [ Object Name: "GoogleMapView"; URL: \$localPath ]

**End If**

**End If**

## Mapping Layout: Active Solution: Generate Map: Display Map

---

**Show Custom Dialog** [ Title: "View in a Browser or in FileMaker"; Message: "Would you like to view this data in a web browser or within FileMaker's Web Viewer?¶¶¶Display Map with out regenerating data. If you experience any problems with this, use the previous button first. 'Generate HTML File and Display Map'"; Default Button: "Browser", Commit: "Yes"; Button 2: "FileMaker", Commit: "No"; Button 3: "Cancel", Commit: "No" ]

**Set Variable** [ \$FileName; Value:settings::gFile\_Name ]

**Set Variable** [ \$Path; Value:Case(  
settings::gSave\_File\_Location="Desktop Path"; Get(DesktopPath);  
settings::gSave\_File\_Location="Temporary Path"; Get(TemporaryPath);  
settings::gSave\_File\_Location="Same Directory as Database File"; "";  
settings::gSave\_File\_Location="FileMaker Folder"; Get ( FileMakerPath );  
settings::gSave\_File\_Location="Documents Folder"; Get ( DocumentsPath )  
) ]

**Set Variable** [ \$tmpPath; Value:\$Path&\$FileName ]

**Set Variable** [ \$localPath; Value:Case(  
Abs(Get(SystemPlatform))=1; "file://" & Substitute(\$tmpPath ; " " ; "%20" );  
Abs(Get(SystemPlatform))=2; "file://" & Substitute(\$tmpPath ; " " ; "%20" )) ]

**If** [ Get(LastMessageChoice)=1 ]

*#Open in a Browser selected.*

**Open URL** [ \$localPath ]  
[ No dialog ]

**Else If** [ Get(LastMessageChoice)=2 ]

*#View in FileMaker Web Viewer selected.*

**Go to Layout** [ "Solution 2 Map View" (settings) ]

**Adjust Window**  
[ Resize to Fit ]

**Set Field** [ settings::gRequested\_URL; \$localPath ]

**Set Web Viewer** [ Object Name: "GoogleMapView"; URL: \$localPath ]

**Else If** [ Get(LastMessageChoice)=3 ]

*#'Cancel' selected.*

**Halt Script**

**End If**

**Set Variable** [ \$SearchParam; Value:Get(ScriptParameter) ]

**New Window** [ Style: Document; Name: "Detail Map"; Left: Get ( WindowLeft)+100; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

**Go to Layout** [ "Solution 2 Map Detail" (site) ]

**Adjust Window**  
[ Restore ]

**Perform Find** [ ]  
[ Restore ]

## Mapping Layout: Active Solution: Generate Map: Perimeter Search

---

**Show Custom Dialog** [ Title: "Perimeter Distance"; Message: "Enter a value in Miles to search and map locations around this location."; Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No"; Input #1: settings::gPerimeterDistance, "Miles:" ]

**If** [ Get(LastMessageChoice)=1 and not IsEmpty(settings::gPerimeterDistance) ]

**Set Variable** [ \$\$PerimeterBox; Value://Convert kilometers to miles for Custom Function by dividing field gPerimeterDistance by 1.609344

//Change field label in dialog box as well.

GeoBox ( address\_site::latitude ; address\_site::longitude ; settings::gPerimeterDistance ) ]

**Enter Find Mode** [ ]

**Set Field** [ address\_site::latitude; GetValue ( \$\$PerimeterBox ; 1 ) & "..."& GetValue ( \$\$PerimeterBox ; 2 )]

**Set Field** [ address\_site::longitude; GetValue ( \$\$PerimeterBox ; 3 ) & "..."& GetValue ( \$\$PerimeterBox ; 4 )]

**Perform Find** [ ]

**Set Error Capture** [ On ]

**If** [ Get(FoundCount)>0 ]

**New Window** [ Style: Dialog; Name: "Perimeter Search - Records Found: " & Get(FoundCount); Top: Get ( WindowTop )+10; Left: Get ( WindowLeft )+100; Close: Yes; Minimize: No; Maximize: Yes; Resize: Yes ]

**Perform Script** [ "Generate JSON, HTML and Display Map"; Parameter: "In FileMaker" ]

**Set Variable** [ \$\$BoundingBox; Value:"Bounding Box: "& "Latitude 1:" & GetValue(\$\$PerimeterBox; 1) & " Longitude 1: "& GetValue(\$\$PerimeterBox; 3) & " Latitude 2:" & GetValue(\$\$PerimeterBox; 2) & " Longitude 2: "& GetValue(\$\$PerimeterBox; 4) ]

**End If**

**End If**

*#Currently not used on layouts as would overwrite existing latitude and longitude co-ordinates*

**Show Custom Dialog** [ Title: "Warning: Slow Process"; Message: "This process takes a while, looping through each record ("&Get FoundCount)&" Records in the Current Found Set) and querying Google Maps API for a set of Longitude and Latitude coordinates. Please be patient.¶¶¶NOTE: You only have to run this script ONCE. If you add new data, then just Search for those records that lack a Longitude and Latitude point and then run it again."; Default Button: "OK", Commit: "Yes", Button 2: "Cancel", Commit: "No" ]

**If** [ Get(LastMessageChoice)=2 ]

**Halt Script**

**End If**

**If** [ IsEmpty(settings::gGoogle\_API\_Key) ]

**Show Custom Dialog** [ Title: "Warning!"; Message: "You will need a Google API Key to run this routine. Or maybe not. Click 'Try' to attempt it without."; Default Button: "Try", Commit: "Yes", Button 2: "Cancel", Commit: "No" ]

**If** [ Get(LastMessageChoice)=2 ]

**Go to Layout** [ "Solution 2 Settings" (settings) ]

**Go to Field** [ settings::gGoogle\_API\_Key ]

**Halt Script**

**End If**

**End If**

**Go to Layout** [ "Solution 2 Data Entry View" (site) ]

**If** [ Get(LastMessageChoice)=1 ]

**Go to Record/Request/Page**  
[ First ]

**Loop** [ Flush: Always ]

**Perform Script** [ "Get Coordinates for this Address" ]

**Go to Record/Request/Page**  
[ Next; Exit after last ]

**End Loop**

**End If**

**Go to Record/Request/Page**  
[ First ]

*#Not currently being used*

**Open URL** [ "https://maps.googleapis.com/maps/api/geocode/json?address=\" & Substitute(address\_site::line1; \",\", \" \") & \",\" & Substitute  
(address\_site::city; \",\", \" \")& \",\" & Substitute(Data::address\_site::county; \",\", \" \")& \",\" & Substitute(address\_site::country; \",\", \" \")  
& \",\" & Substitute(address\_site::postcode; \",\", \" \") & \"&output=csv&key=\" & Settings::gGoogle\_API\_Key&\"&sensor=true" ]  
[ No dialog ]

**If** [ not IsEmpty(address\_site::longitude or address\_site::latitude) ]

**Show Custom Dialog** [ Title: "Coordinates Already Exist"; Message: "Coordinates already exist for this site and will be overwritten.";  
Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 2 ]

**Exit Script** [ ]

**End If**

**End If**

**Go to Layout** [ original layout ]

**Insert from URL** [ site::inserturl; "http://maps.googleapis.com/maps/api/geocode/json?address=" & Substitute(address\_site::line1; ";; " ") &  
" " & Substitute(address\_site::FKtownsandcities\_id; ";; " ")& " " & Substitute(address\_site::FKcounties\_PKid; ";; " ")& " " & Substitute  
(address\_site::country; ";; " ")& " " & Substitute(address\_site::postcode; ";; " ") & "&output=csv&key=" & settings::  
gGoogle\_API\_Key&"&sensor=true" ]  
[ Select; No dialog ]

**Commit Records/Requests**

[ No dialog ]

**Set Variable** [ \$response\_from\_google; Value:site::inserturl ]

**Set Variable** [ \$LongLatChunk; Value:Extract ( \$Response\_From\_Google ; "\"location\" : {\" ; "\"location\_type\""" ) ]

**Set Field** [ address\_site::latitude; Filter(Trim(Extract ( \$LongLatChunk ; "\"lat\" : \" ; \" ; \" )); "0123456789-." ) ]

**Set Field** [ address\_site::longitude; Filter(Trim(Extract ( \$LongLatChunk ; "\"lng\" : \" ; \" ; \" )); "0123456789-." ) ]

**Set Web Viewer** [ Object Name: "GPSWebService"; URL: "" ]

Go to Layout [ "Solution 2 Data Entry View" (site) ]

Adjust Window  
[ Resize to Fit ]

Show All Records



## Mapping Layout: Active Solution: Settings: Check for '.htm' in File Name

---

**If** [ PatternCount(settings::gFile\_Name; ".htm")=0 ]

**Show Custom Dialog** [ Title: "Warning"; Message: "Please add the extension '.htm' or '.html' to your file name. If you do not include this extension, your mapping file may not open correctly."; Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**Go to Layout** [ "Solution 2 Settings" (settings) ]

**Go to Field** [ settings::gFile\_Name ]  
[ Select/perform ]

**Halt Script**

**End If**

Perform Find [ ]

Sort Records by Field [ Ascending; development::projectordevid ]

Go to Layout [ "lay\_site" (site) ]

New Record/Request



Go to Layout [ "lay\_site" (site) ]

Enter Find Mode [ ]

Go to Layout [ "Solution 2 Data Entry View" (site) ]

**Set Error Capture** [ On ]

**Set Variable** [ \$siteid; Value:site::PKid ]

**Enter Find Mode** [ ]

**Set Field** [ development::FKsite\_id; \$siteid ]

**Perform Find** [ ]

**Show Custom Dialog** [ Title: "Developments at this site"; Message: Case ((Get (FoundCount) = 0);"No developments were found";(Get (FoundCount) - 1) > 0;(Get (FoundCount) - 1)&" developments were found at this site. Do you wish to View them?"); Default Button: "OK", Commit: "No"; Button 2: "View", Commit: "No" ]

**If** [ Get(FoundCount) > 0 ]

**Sort Records by Field** [ Ascending; development::FKdevelopmenttype\_id ]

**Else** [ ]

**Show All Records**

**End If**

Go to Layout [ original layout ]

Show All Records

Close Popover



Set Error Capture [ On ]

New Window [ Style: Document; Name: "Developments List Report"; Height: 800; Width: 800; Top: 50; Left: 50; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

Go to Layout [ "developments report" (development) ]

Show All Records

Set Error Capture [ On ]

New Window [ Style: Document; Name: "Planning Schedule"; Height: 800; Width: 800; Top: 50; Left: 50; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

Go to Layout [ "planning schedule" (development) ]

Enter Find Mode [ ]

Set Field [ development::FKdevelopmenttype\_id; 1 ]

Set Field [ development::FKstatus\_ID\_advertapplication; "\*" ]

Perform Find [ ]

Set Error Capture [ On ]

New Window [ Style: Document; Name: "Planning Conditons"; Height: 800; Width: 800; Top: 50; Left: 50; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

Go to Layout [ "planning conditions schedule " (planningconditions) ]

Set Error Capture [ On ]

New Window [ Style: Document; Name: "Planning Schedule"; Height: 800; Width: 800; Top: 50; Left: 50; Close: Yes; Minimize: Yes; Maximize: Yes; Resize: Yes ]

Go to Layout [ "planning schedule" (development) ]

Enter Find Mode [ ]

Set Field [ development::FKdevelopmenttype\_id; 2 ]

Set Field [ development::FKstatus\_ID\_advertapplication; "" ]

Perform Find [ ]

Go to Layout [ "lay\_companiesandcontacts" (organisation) ]

Show All Records

## Development Layout: Development Tab: Development Type Script Trigger

---

**If** [ development::FKdevelopmenttype\_id = 3 ]

**Show Custom Dialog** [ Title: "Existing Advertising"; Message: "When recording existing advertising at a site, a new development should be added for each separate media owner operating panel(s) at the site.¶¶¶If a media owner has multiple panels, these can be recorded in there against their development."; Default Button: "OK", Commit: "Yes"]

**Else** [ ]

**If** [ development::FKdevelopmenttype\_id = 1 ]

**Set Field** [ development::FKorganisation\_id\_developercompany; 779 ]

**End If**

**End If**

**Show Custom Dialog** [ Title: "Projects"; Message: "A development can be related to a specific client or Wildstone 'Project'. For example a project to create a digital network in a region for a client." & ¶&¶ &

"You can select existing projects from the drop down list below." & ¶&¶&

"To create a new client project, go to the 'Companies' layout and add the project on the 'Project' tab."; Default Button: "OK", Commit: "Yes"]

**Show Custom Dialog** [ Title: "Remove development from project"; Message: "Are you sure you want to remove this development from this project?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**  
        [ No dialog ]

**End If**



Refresh Object [ Object Name: "DevelopmentName"; Repetition: 1 ]

```
Show Custom Dialog [ Title: "Delete Panel Format"; Message: "Are you sure you want to delete this panel format at this development?";  
    Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]  
  
If [ Get (LastMessageChoice) = 1 ]  
    Delete Portal Row  
    [ No dialog ]  
  
End If
```

Clear [ development::FKcontact\_id\_nonwildstonedeveloper ]  
[ Select ]

Clear [ development::FKcontact\_id\_lawyer ]  
[ Select ]

Clear [ development::FKcontact\_id\_mediaowner ]  
[ Select ]

Clear [ development::FKcontact\_id\_mediaowneragent ]  
[ Select ]

**If** [ IsEmpty ( development::name ) ]

**Show Custom Dialog** [ Title: "Development Name"; Message: "The Panel Type will be used as the Development Name. This can be modified afterwards."; Default Button: "OK", Commit: "No" ]

**Set Field** [ development::name; address\_site::line1 ]

**Else If** [ development::name ]

**Show Custom Dialog** [ Title: "Modify Development Name"; Message: "Do you want to update the Development Name to show this change?"; Default Button: "Yes", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Set Field** [ development::name; <Table Missing>::<Field Missing> ]

**End If**

**End If**

Go to Field [ [development::<Field Missing>](#) ]

Insert PDF [ ]



## Development Layout: Survey Tab: Open ASGF

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::<Field Missing>));"remote:."; "" ) ]

**Export Field Contents** [ development::<Field Missing>; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Survey Tab: Delete ASGF

---

**Show Custom Dialog** [ Title: "Delete ASGF"; Message: "Are you sure you want to delete this ASGF Report?"; Default Button: "Yes",  
Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::<Field Missing> ]  
[ Select ]

**End If**

Go to Field [ [asgfreports::asgfreport](#) ]

Insert File [   ]

## Development Layout: Survey Tab: Open ASGF Report in Portal

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (asgfreports::asgfreport)),"remote:."; "" ) ]

**Export Field Contents** [ asgfreports::asgfreport; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Survey Tab: Delete ASGF in Portal

---

**Show Custom Dialog** [ Title: "Delete ASGF"; Message: "Are you sure you want to delete this ASGF Report?"; Default Button: "Yes",  
Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ asgfreports::asgfreport ]  
[ Select ]

**End If**

Go to Field [ [development::currentlease](#) ]

Insert PDF [ ]

## Development Layout: Commercial Agreement Tab: Open Current Lease

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:"; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::currentlease));"remote:"; "" ) ]

**Export Field Contents** [ development::currentlease; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Commercial Agreement Tab: Delete Current Lease

---

**Show Custom Dialog** [ Title: "Delete Current Lease"; Message: "Are you sure you want to delete this current lease?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::currentlease ]

        [ Select ]

**End If**



Go to Field [ [contractdocuments::document](#) ]

Insert File [ ]

## Development Layout: Commercial Agreement Tab: Open Contract Document

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (contractdocuments::document));"remote:."; "" ) ]

**Export Field Contents** [ contractdocuments::document; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Commercial Agreement Tab: Delete Contract Document

---

**Show Custom Dialog** [ Title: "Delete Current Lease"; Message: "Are you sure you want to delete this current lease?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ contractdocuments::document ]

    [ Select ]

**End If**

Delete Portal Row

Go to Field [ [development::currentlease](#) ]

Insert PDF [ ]

## Development Layout: Commercial Agreement Tab: Delete Contract

---

**Show Custom Dialog** [ Title: "Delete Contract"; Message: "Are you sure you want to delete this Contract?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::currentlease ]  
        [ Select ]

**End If**

Go to Field [ [development::design](#) ]

Insert PDF [ ]

## Development Layout: Design Tab: Delete Design

---

**Show Custom Dialog** [ Title: "Delete Design"; Message: "Are you sure you want to delete this design?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::design ]  
        [ Select ]

**End If**



If [ development::advertconditions = "No" ]

    Clear [ development::advertconditionsnumber ]

        [ Select ]

End If

```
If [ IsEmpty (organisation_local authority::planningportalurl) ]  
    Show Custom Dialog [ Title: "Planning URL"; Message: "Please add the specific URL for the planning applications website for this  
        local authority first."; Default Button: "OK", Commit: "Yes" ]  
Else [ ]  
    Copy [ development::advertapprefla ]  
        [ Select ]  
    Open URL [ organisation_local authority::planningportalurl ]  
        [ No dialog ]  
End If
```

```
If [ IsEmpty (organisation_local authority::planningportalurl) ]  
    Show Custom Dialog [ Title: "Planning URL"; Message: "Please add the specific URL for the planning applications website for this  
        local authority first."; Default Button: "OK", Commit: "Yes" ]  
Else [ ]  
    Copy [ development::planningapprefla ]  
        [ Select ]  
    Open URL [ organisation_local authority::planningportalurl ]  
        [ No dialog ]  
End If
```

```
If [ IsEmpty (organisation_local authority::planningportalurl) ]  
    Show Custom Dialog [ Title: "Planning URL"; Message: "Please add the specific URL for the planning applications website for this  
        local authority first."; Default Button: "OK", Commit: "Yes" ]  
Else [ ]  
    Copy [ planningconditions::conditionapprefla ]  
        [ Select ]  
    Open URL [ organisation_local authority::planningportalurl ]  
        [ No dialog ]  
End If
```

Copy [ development::planningappealrefla ]  
[ Select ]

Open URL [ "http://www.pcs.planningportal.gov.uk/pcsportal/casesearch.asp"]  
[ No dialog ]

**Copy** [ development::advertappealrefla ]

[ Select ]

**Open URL** [ "http://www.pcs.planningportal.gov.uk/pcsportal/casesearch.asp"]

[ No dialog ]

**Allow User Abort** [ Off ]

**New Window** [ Style: Dialog; Name: "Planning Documents " & development::name & ", " & site::name; Top: 0; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Show/Hide Toolbars**  
[ Hide ]

**Go to Layout** [ "lay\_planningpdfs" (development) ]

**Adjust Window**  
[ Resize to Fit ]

**Sort Records by Field** [ Associated value list; planningdocs::relatedplanningapplication ]

**Clear** [ development::advertappealrepresentations ]  
[ Select ]

**Clear** [ development::advertappealhearing ]  
[ Select ]

**Clear** [ development::advertappealforecastdetermination ]  
[ Select ]



**Clear** [ development::planningappealrepresentations ]  
[ Select ]

**Clear** [ development::planningappealhearing ]  
[ Select ]

**Clear** [ development::planningappealforecastdetermination ]  
[ Select ]

If [ IsEmpty ( development::FKcontact\_id\_advertisementplanningofficer ) ]

Exit Script [ ]

End If

Set Variable [ \$officer; Value:development::FKcontact\_id\_advertisementplanningofficer ]

New Window [ Style: Floating Document; Name: "Contacts Details"; Top: 250; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Go to Layout [ "lay\_contact" (contact) ]

Adjust Window

[ Resize to Fit ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Enter Find Mode [ ]

Set Field [ contact::PKid; \$officer ]

Perform Find [ ]

If [ IsEmpty ( development::FKcontact\_id\_planningplanningofficer ) ]

Exit Script [ ]

End If

Set Variable [ \$officer; Value:development::FKcontact\_id\_planningplanningofficer ]

New Window [ Style: Floating Document; Name: "Contacts Details"; Top: 250; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Go to Layout [ "lay\_contact" (contact) ]

Adjust Window

[ Resize to Fit ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Enter Find Mode [ ]

Set Field [ contact::PKid; \$officer ]

Perform Find [ ]

If [ IsEmpty ( development::FKcontact\_id\_planningplanningofficer ) ]

Exit Script [ ]

End If

Set Variable [ \$officer; Value:planningconditions::FKcontact\_id\_conditionofficer ]

New Window [ Style: Floating Document; Name: "Contacts Details"; Top: 250; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Go to Layout [ "lay\_contact" (contact) ]

Adjust Window

[ Resize to Fit ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Enter Find Mode [ ]

Set Field [ contact::PKid; \$officer ]

Perform Find [ ]

If [ IsEmpty ( development::FKcontact\_id\_preapplaningofficer ) ]

Exit Script [ ]

End If

Set Variable [ \$officer; Value:development::FKcontact\_id\_preapplaningofficer ]

New Window [ Style: Floating Document; Name: "Contacts Details - " & contact\_localauthority::combinedname; Top: 250; Left: 100; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Go to Layout [ "lay\_contact" (contact) ]

Adjust Window

[ Resize to Fit ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Enter Find Mode [ ]

Set Field [ contact::PKid; \$officer ]

Perform Find [ ]

If [ IsEmpty ( development::dialogplanningofficer ) ]

Exit Script [ ]

Else [ ]

Set Variable [ \$lpa; Value:site::FKorganisation\_id\_localauthority ]

Commit Records/Requests

New Window [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Go to Layout [ "lay\_contact" (contact) ]

New Record/Request

Adjust Window

[ Resize to Fit ]

Set Field [ contact::FKorganisation\_id; \$lpa ]

Go to Field [ contact::salutation ]

End If

Exit Script [ ]

**If** [ IsEmpty ( development::dialogplanninginspector ) ]

**Exit Script** [ ]

**Else** [ ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Show/Hide Menubar**

        [ Hide ]

**Show/Hide Toolbars**

        [ Hide ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

        [ Resize to Fit ]

**New Record/Request**

**Set Field** [ contact::FKorganisation\_id; 486 ]

**Go to Field** [ contact::salutation ]

**End If**

**Exit Script** [ ]

If [ IsEmpty ( development::FKcontact\_id\_planningappealinspector ) ]

Exit Script [ ]

End If

Set Variable [ \$inspector; Value:development::FKcontact\_id\_planningappealinspector ]

New Window [ Style: Floating Document; Name: "Contacts Details - " & contact\_localauthority::combinedname; Top: 250; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

Go to Layout [ "lay\_contact" (contact) ]

Adjust Window

[ Resize to Fit ]

Show/Hide Menubar

[ Hide ]

Show/Hide Toolbars

[ Hide ]

Enter Find Mode [ ]

Set Field [ contact::PKid; \$inspector ]

Perform Find [ ]



**If** [ IsEmpty ( development::FKcontact\_id\_caseofficer ) ]

**Exit Script** [ ]

**End If**

**Set Variable** [ \$offapp; Value:development::FKcontact\_id\_caseofficer]

**New Window** [ Style: Floating Document; Name: "Contacts Details - " & contact\_localauthority::combinedname; Top: 250; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

    [ Resize to Fit ]

**Show/Hide Menubar**

    [ Hide ]

**Show/Hide Toolbars**

    [ Hide ]

**Enter Find Mode** [ ]

**Set Field** [ contact::PKid; \$offapp ]

**Perform Find** [ ]

## Development Layout: Planning Tab: Add Planning Officer Contacts

---

**Set Variable** [ \$officer; Value:site::FKorganisation\_id\_localauthority ]

### Commit Records/Requests

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

### Adjust Window

[ Resize to Fit ]

### Show/Hide Menubar

[ Hide ]

### Show/Hide Toolbars

[ Hide ]

### New Record/Request

**Set Field** [ contact::FKorganisation\_id; \$officer ]

**Go to Field** [ contact::fname ]

**Exit Script** [ ]

**Commit Records/Requests**

**New Window** [ Style: Dialog; Name: "Add Contact"; Top: 100; Left: 200; Close: Yes; Minimize: No; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_contact" (contact) ]

**Adjust Window**

[ Resize to Fit ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Hide ]

**New Record/Request**

**Set Field** [ contact::FKorganisation\_id; 486 ]

**Go to Field** [ contact::fname ]

**Exit Script** [ ]

Go to Field [ [development::preappdoc](#) ]

Insert File [   ]

## Development Layout: Planning Tab: Open PreApp Doc

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::preappdoc));"remote:."; "" ) ]

**Export Field Contents** [ development::preappdoc; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Planning Tab: Delete PreApp Doc

---

**Show Custom Dialog** [ Title: "Delete Related File"; Message: "Are you sure you want to delete this file?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::preappdoc ]  
        [ Select ]

**End If**

**Development Layout: Planning Tab: Delete Planning Condition**

---

**Show Custom Dialog** [ Title: "Delete Planning Condition"; Message: "Are you sure you want to delete this planning condition?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

**Development Layout: Marketing Tab: Delete Tender Response Portal Row**

---

**Show Custom Dialog** [ Title: "Delete Tender Response"; Message: "Are you sure you want to delete this tender response?"; Default Button: "Yes", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**



## Development Layout: Marketing Tab: Open Tender Response Document

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (tenderoffers::response)),"remote:."; "" ) ]

**Export Field Contents** [ tenderoffers::response; "\$temp"; Automatically open; Create folders:No ]

Go to Field [ [tenderoffers::response](#) ]

Insert File [   ]

## Development Layout: Marketing Tab: Delete Tender Response Document

---

**Show Custom Dialog** [ Title: "Delete Related File"; Message: "Are you sure you want to delete this file?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ tenderoffers::response ]  
        [ Select ]

**End If**

**Allow User Abort** [ Off ]

**New Window** [ Style: Dialog; Name: "Tender Documents " & development::name & ", " & site::name; Top: 0; Left: 100; Close:Yes; Minimize: No; Maximize: No; Resize: No ]

**Show/Hide Toolbars**  
[ Hide ]

**Go to Layout** [ “lay\_tenderpdfs” (development) ]

**Adjust Window**  
[ Resize to Fit ]

Go to Field [ [development::tendermarketingvideo](#) ]

Insert Audio/Video [   ]

## Development Layout: Marketing Tab: Delete Marketing Video

---

**Show Custom Dialog** [ Title: "Delete Title Plan"; Message: "Are you sure you want to delete this video?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::tendermarketingvideo ]  
[ Select ]

**End If**

Set Field [ development::selectedbuildparttender; jndevelopmentbuildpart::PKid ]

**Commit Records/Requests**

[ No dialog ]

**Refresh Window**

[ Flush cached join results; Flush cached external data ]

**Set Field** [ jndevelopmentbuildpart::FKtenderresponse\_id\_acceptedbid; Let (

id = buildtenderresponses\_filtered::PKid ;

Case ( id = jndevelopmentbuildpart::FKtenderresponse\_id\_acceptedbid ; "" ; id )

) ]

**Commit Records/Requests**

[ Skip data entry validation; No dialog ]

**Refresh Window**

[ Flush cached join results; Flush cached external data ]



```
Show Custom Dialog [ Title: "Delete Build Part"; Message: "Are you sure you want to remove this section of the build?"; Default Button:
    "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]
If [ Get (LastMessageChoice) = 1 ]
    Delete Portal Row
    [ No dialog ]
End If
```

## Development Layout: Build Tab: Delete Build Tender Response

---

**Show Custom Dialog** [ Title: "Delete Build Part"; Message: "Are you sure you want to delete this Tender Response?"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Set Field [ development::selectedbuildpartsnagging; jndevelopmentbuildpart::PKid ]

Commit Records/Requests

[ No dialog ]

Refresh Window

[ Flush cached join results; Flush cached external data ]

## Development Layout: Build Tab: Delete Snagging Item

---

**Show Custom Dialog** [ Title: "Delete Snagging"; Message: "Are you sure you want to delete this item?"; Default Button: "OK", Commit: "No";  
Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

**Development Layout: Consultancy Tab: Rent Review: Delete Contact Portal Row**

---

**Show Custom Dialog** [ Title: "Delete Developer"; Message: "Are you sure you want to delete this contact?"; Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

**Show Custom Dialog** [ Title: "Delete Rent Review"; Message: "Are you sure you want to delete this rent review? All associated detail will also be deleted"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Go to Field [ [development::currentlease](#) ]

Insert File [ ]

## Development Layout: Consultancy Tab: Rent Review: Open Current Lease

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::currentlease));"remote:."; "" ) ]

**Export Field Contents** [ development::currentlease; "\$temp"; Automatically open; Create folders:No ]



## Development Layout: Consultancy Tab: Rent Review: Delete Current Lease

---

**Show Custom Dialog** [ Title: "Delete Current Lease"; Message: "Are you sure you want to delete the current lease?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::currentlease ]

[ Select ]

**End If**

Go to Field [ [consultancyrentreview::triggernotice](#) ]

Insert File [   ]

## Development Layout: Consultancy Tab: Rent Review: Open Trigger Notice

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (consultancyrentreview::triggernotice)),"remote:."; "" ) ]

**Export Field Contents** [ consultancyrentreview::triggernotice; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Consultancy Tab: Rent Review: Delete Trigger Notice

---

**Show Custom Dialog** [ Title: "Delete Trigger Notice"; Message: "Are you sure you want to delete the trigger notice?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ consultancyrentreview::triggernotice ]

[ Select ]

**End If**

Go to Field [ [consultancyrentreview::memoranda](#) ]

Insert File [   ]

## Development Layout: Consultancy Tab: Rent Review: Open Memoranda

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (consultancyrentreview::memoranda));"remote:."; "" ) ]

**Export Field Contents** [ consultancyrentreview::memoranda; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Consultancy Tab: Rent Review: Delete Memoranda

---

**Show Custom Dialog** [ Title: "Delete Memoranda"; Message: "Are you sure you want to delete this memoranda?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ consultancyrentreview::memoranda ]

[ Select ]

**End If**

Go to Field [ [development::renewalnotice](#) ]

Insert File [ ]



## Development Layout: Consultancy Tab: Lease Renewal: Open Renewal Notice

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::renewalnotice));"remote:."; "" ) ]

**Export Field Contents** [ development::renewalnotice; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Consultancy Tab: Lease Renewal: Delete Renewal Noice

---

**Show Custom Dialog** [ Title: "Delete Renewal Notice"; Message: "Are you sure you want to delete this Renewal Notice?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::renewalnotice ]

        [ Select ]

**End If**

**Development Layout: Consultancy Tab: Valuation: Delete Valuation Portal Row**

---

**Show Custom Dialog** [ Title: "Delete Valuation"; Message: "Are you sure you want to delete this valuation? All associated detail will also be deleted"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Go to Field [ [consultancyvaluation::valuation](#) ]

Insert File [ ]

## Development Layout: Consultancy Tab: Valuation: Open Valuation

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (consultancyvaluation::valuation));"remote:."; "" ) ]

**Export Field Contents** [ consultancyvaluation::valuation; "\$temp"; Automatically open; Create folders:No ]

## Development Layout: Consultancy Tab: Valuation: Delete Valuation

---

**Show Custom Dialog** [ Title: "Delete Valuation"; Message: "Are you sure you want to delete the valuation document?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ consultancyvaluation::valuation ]  
        [ Select ]

**End If**

**Show Custom Dialog** [ Title: "Delete Audit"; Message: "Are you sure you want to delete this audit? All associated detail will also be deleted";

Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Go to Field [ [consultancyrevenue::audit](#) ]

Insert File [   ]



## Development Layout: Consultancy Tab: Revenue Share Audit: Open Audit

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (consultancyrevenue::audit));"remote:."; "" ) ]

**Export Field Contents** [ consultancyrevenue::audit; "\$temp"; Automatically open; Create folders:No ]

**Show Custom Dialog** [ Title: "Delete Audit"; Message: "Are you sure you want to delete this Audit document?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ consultancyrevenue::audit ]  
        [ Select ]

**End If**

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac: "; "filewin: " ) & Get ( TemporaryPath ) & Substitute((GetAsText (development::currentlease));"remote: "; "" ) ]

**Export Field Contents** [ development::currentlease; "\$temp"; Automatically open; Create folders:No ]

**Show Custom Dialog** [ Title: "Delete Current Lease"; Message: "Are you sure you want to delete the current lease?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ development::currentlease ]

        [ Select ]

**Go to Object** [ Object Name: "rentCurrentLease"; Repetition: 1 ]

**End If**

Go to Object [ Object Name: "rentCurrentLease"; Repetition: 1 ]

Insert File [ ]

**Development Layout: Consultancy Tab: Other: Delete Other Consultancy Portal Row**

---

**Show Custom Dialog** [ Title: "Delete 'Other' consultancy instruction?"; Message: "Are you sure you want to delete this instruction? All associated detail will also be deleted"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Go to Field [ [consultancyother::document](#) ]

Insert File [ ]

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:."; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (consultancyother::document));"remote:."; "" ) ]

**Export Field Contents** [ consultancyother::document; "\$temp"; Automatically open; Create folders:No ]



## Development Layout: Consultancy Tab: Other: Delete Other Doc

---

**Show Custom Dialog** [ Title: "Delete Document"; Message: "Are you sure you want to delete this document?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ consultancyother::document ]  
    [ Select ]

**End If**

Commit Records/Requests

Refresh Window

[ Flush cached join results ]

## Development Layout: Tasks & Notes Tab: Send Task Email Assigned

---

**Show Custom Dialog** [ Title: "Send Task Notification"; Message: "Do you want to to notify the assignee of this task by email that it has been created?"; Default Button: "Yes", Commit: "No"; Button 2: "No", Commit: "No" ]

**If** [ Get(LastMessageChoice)=1 ]

**Send Mail** [ Send via E-mail Client; To: useraccounts\_developmenttasksassignee::email; Subject: "New Task Assigned to you for the development " & development::name & " at " & site::name; Message: "You've been assigned a new task for the development " & development::name & " at " & site::name & "." & ¶ & ¶ &

"Task Type: " & tasktype::action & ¶ & ¶ &

"Task: " & jndevelopmenttasks::notes & ¶ & ¶ &

"Due Date: " & jndevelopmenttasks::duedate & ¶ & ¶ &

"By: " & jndevelopmenttasks::createdby ]

**End If**

If [ jndevelopmenttasks::complete = "Complete" ]

**Show Custom Dialog** [ Title: "Send Task Complete Notification"; Message: "Do you want to notify the assignor of this task that it has been completed?"; Default Button: "Yes", Commit: "No"; Button 2: "No", Commit: "No" ]

If [ Get ( LastMessageChoice ) = 1 ]

**Send Mail** [ Send via E-mail Client; To: useraccounts\_developmenttaskcreator::email; Subject: "Task Complete - " & development::name & " at " & site::name; Message: "The following task has been completed" & ¶ & ¶ &

"Task Type:" & tasktype::action & ¶ & ¶ &

"Task:" & jndevelopmenttasks::notes & ¶ & ¶ &

"Completed:"& jndevelopmenttasks::completed & ¶ & ¶ &

"By:" & jndevelopmenttasks::assignedto ]

End If

End If

**Commit Records/Requests**

**Refresh Window**

[ Flush cached join results ]

```
Show Custom Dialog [ Title: "Delete Task"; Message: "Are you sure you want to delete this task?"; Default Button: "OK", Commit: "No";  
    Button 2: "Cancel", Commit: "No" ]  
If [ Get (LastMessageChoice) = 1 ]  
    Delete Portal Row  
    [ No dialog ]  
End If
```

Set Field [ development::gtasksbystatus; "incomplete" ]

Go to Field [ [jndevelopmentnotes::relatedfile](#) ]

Insert File [   ]

## Development Layout: Tasks & Notes Tab: Open Notes File

---

**Show Custom Dialog** [ Title: "Viewing File"; Message: "The file will open in it's native application. If you edit the document you'll need to upload the ammended version."; Default Button: "OK", Commit: "Yes" ]

**Set Variable** [ \$temp; Value:Case ( Abs ( Get (SystemPlatform)) = 1; "filemac:"; "filewin:" ) & Get ( TemporaryPath ) & Substitute((GetAsText (jndevelopmentnotes::relatedfile));"remote:"; "" ) ]

**Export Field Contents** [ jndevelopmentnotes::relatedfile; "\$temp"; Automatically open; Create folders:No ]



## Development Layout: Tasks & Notes Tab: Delete Note File

---

**Show Custom Dialog** [ Title: "Delete Related File"; Message: "Are you sure you want to delete this file?"; Default Button: "Yes", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Clear** [ indevelopmentnotes::relatedfile ]  
        [ Select ]

**End If**

**Development Layout: Tasks & Notes Tab: Delete Note Portal Row**

---

**Show Custom Dialog** [ Title: "Delete ASGF"; Message: "Are you sure you want to delete this ASGF?"; Default Button: "OK", Commit: "No";  
Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Go to Layout [ "lay\_site" (site) ]

Show All Records

Enter Find Mode [ ]

Go to Layout [ "lay\_development" (development) ]

Show All Records

Enter Find Mode [ ]

## Organisation Layout: Add Organisation

---

**Show Custom Dialog** [ Title: "Duplicate Organisations"; Message: "It's important that duplicate organisations are not added to the database." ]

& ¶ & ¶ & "Prior to entering a new organisation, perform a search on this form to see if it already exists. Otherwise, continue by clicking 'Add'"; Default Button: "Search", Commit: "No"; Button 2: "Add", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Enter Find Mode** [ ]

**Exit Script** [ ]

**Else If** [ Get ( LastMessageChoice ) = 2 ]

**New Record/Request**

**End If**

**Set Error Capture** [ On ]

**Freeze Window**

*#Copy the Account Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents ) ]

**Go to Layout** [ "tbl\_townsandcities" (townsandcities) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ townsandcities::Name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Account record*

**Set Variable** [ \$id; Value:townsandcities::PKid ]

**Go to Layout** [ original layout ]

*#Link the Invoice record to the Account record*

**Set Field** [ address\_organisation::FKtownsandcities id; \$id ]

*#Normally you would navigate to the next desired field, say the invoice date or the first line item.*

// **Go to Object** [ ]

**Set Error Capture** [ On ]

**Freeze Window**

*#Copy the Account Name contents*

**Set Variable** [ \$name; Value:Get ( ActiveFieldContents )]

**Go to Layout** [ "tbl\_counties" (counties) ]

**Enter Find Mode** [ ]

*#Set the find criteria - use the Quote function to return the entire name is quotes, which prevents quotes in the name from causing find problems. Two equal signs (==) ensure we find an exact match.*

**Set Field** [ counties::name; "==" & Quote(\$name) ]

**Perform Find** [ ]

*#Grab the Unique ID of the Account record*

**Set Variable** [ \$id; Value:counties::PKid ]

**Go to Layout** [ original layout ]

*#Link the Invoice record to the Account record*

**Set Field** [ address\_organisation::FKcounties PKid; \$id ]

*#Normally you would navigate to the next desired field, say the invoice date or the first line item.*

// **Go to Object** [ ]

## Organisation Layout: Check for Duplicate Company Entries

---

**Set Error Capture** [ On ]

**Commit Records/Requests**

**If** [ Get ( LastError ) ≠ 0 ]

**Show Custom Dialog** [ Title: "Duplicate Company Record"; Message: "This company already exists.¶¶ - 'Delete' the record if this is a duplicate.¶¶ - If the new company you are creating is a subsidiary of the existing company, click 'New' and re-enter it's unique name.¶¶ - If you wish to view/edit existing companies matching this name click 'Search' "; Default Button: "Delete", Commit: "No"; Button 2: "New", Commit: "No"; Button 3: "Search", Commit: "No" ]

**If** [ Get ( LastMessageChoice ) = 1 ]

**Revert Record/Request**

[ No dialog ]

**Else If** [ Get ( LastMessageChoice ) = 2 ]

**Go to Field** [ organisation::name ]

**Exit Script** [ Result: False ]

**Else If** [ Get ( LastMessageChoice ) = 3 ]

**Set Variable** [ \$company; Value:organisation::name ]

**Commit Records/Requests**

**Delete Record/Request**

[ No dialog ]

**Enter Find Mode** [ ]

**Set Field** [ organisation::name; \$company ]

**Perform Find** [ ]

**End If**

**End If**



## Organisation Layout: Contacts: Delete Contact Portal Row Orgs

---

*#Not using as deleting the contact creates orphan records across the database*

**Show Custom Dialog** [ Title: "Delete Contact"; Message: "Are you sure you want to delete this contact?"; Default Button: "OK", Commit: "Yes"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Send Mail [ Send via E-mail Client; To: contact::email]

## Tender PDF Layout: Insert PDF Tender

---

If [ IsEmpty(tenderdocumantstest::description)=1 ]

Set Variable [ \$idKey; Value:development::PKid ]

Set Selection [ development::activepdftender ]

Insert PDF [ ]

Freeze Window

Go to Layout [ "lay\_pdfstender" (tenderdocumantstest) ]

New Record/Request

Set Field [ tenderdocumantstest::FKdevelopment\_id; \$idKey ]

Set Field [ tenderdocumantstest::pdf; development::activepdftender ]

Else [ ]

Set Selection [ development::activepdftender ]

Insert PDF [ ]

Freeze Window

Go to Related Record [ From table: "tenderdocumantstest"; Using layout: "lay\_pdfstender" (tenderdocumantstest) ]

Set Field [ tenderdocumantstest::pdf; development::activepdftender ]

End If

Go to Layout [ "lay\_tenderpdfs" (development) ]

**Set Variable** [ \$thepdf; Value:tenderdocumantstest::pdf ]

**Set Field** [ development::activepdf tender; \$thePDF ]

## Tender PDF Layout: Delete Tender Document

---

**Show Custom Dialog** [ Title: "Delete Task"; Message: "Are you sure you want to delete this tender file?"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**



## Planning PDF Layout: Insert Planning Document

---

**Set Variable** [ \$portalRow; Value:Get(ActivePortalRowNumber) ]

**Set Variable** [ \$idKey; Value:development::PKid ]

**Set Selection** [ development::activepdf ]

**Insert PDF** [ ]

**Freeze Window**

**Go to Layout** [ "lay\_pdfs" (planningdocs) ]

**New Record/Request**

**Set Field** [ planningdocs::FKdevelopment\_id; \$idKey ]

**Set Field** [ planningdocs::pdf; development::activepdf ]

**Commit Records/Requests**

**Insert Calculated Result** [ planningdocs::setactivatefield; planningdocs::PKid ]  
[ Select ]

**Go to Layout** [ "lay\_planningpdfs" (development) ]

**Go to Portal Row** [ \$portalRow ]

[ No dialog ]

**Go to Object** [ Object Name: "descrip"; Repetition: 1 ]

**Set Variable** [ \$thepdf; Value:planningdocs::pdf ]

**Set Field** [ development::activepdf; \$thePDF ]



## Planning PDF Layout: Delete Planning Document

---

**Show Custom Dialog** [ Title: "Delete Task"; Message: "Are you sure you want to delete this planning document?"; Default Button: "OK", Commit: "No"; Button 2: "Cancel", Commit: "No" ]

**If** [ Get (LastMessageChoice) = 1 ]

**Delete Portal Row**

[ No dialog ]

**End If**

Set Selection [ [planningdocs::pdf](#) ]

Enter Find Mode [ ]

Set Field [ development::wildstonedeveloperorcontact; "External Developer" ]

Perform Find [ ]

**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: development::name

address site::line1  
address site::line2  
townsandcities addresssite::Name  
address site::postcode  
organisation localauthority::name  
development::projectordevid  
developmenttype::type  
developmentstatus::status  
applicationstatusadvert::status  
organisation siteowner::name  
contact siteowner::combinedname  
contact siteowner::email  
contact siteowner::tel  
contact siteowner::extension  
contact siteowner::mob  
organisation siteagent::name  
contact siteagent::combinedname  
contact siteagent::email  
contact siteagent::tel  
contact siteagent::extension  
contact siteagent::mob  
organisation mediaowner::name  
development::purchaseprice  
development::rentalvalue  
development::profityear1  
development::wildstonedeveloperorcontact ]  
[ Format output using current layout ]

**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: development::name

address site::line1  
address site::line2  
townsandcities addresssite::Name  
address site::postcode  
organisation localauthority::name  
development::projectordevid  
developmenttype::type  
developmentstatus::status  
applicationstatusadvert::status  
organisation siteowner::name  
contact siteowner::combinedname  
contact siteowner::email  
contact siteowner::tel  
contact siteowner::extension  
contact siteowner::mob  
organisation siteagent::name  
contact siteagent::combinedname  
contact siteagent::email  
contact siteagent::tel  
contact siteagent::extension  
contact siteagent::mob  
organisation mediaowner::name  
development::purchaseprice  
development::totalcurrentrent  
development::rentalvalue  
development::rentuplift  
development::profityear1  
development::wildstonedeveloperorcontact ]  
[ Format output using current layout ]

**Print Setup** [ Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore ]

**Print** [ Records being browsed; All Pages; Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore: HP LaserJet 2200 Dom iMac @ Dominic's iMac ]

**Print Setup** [ Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore ]

**Print** [ Records being browsed; All Pages; Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore: HP LaserJet 2200 Dom iMac @ Dominic's iMac ]

**Print Setup** [ Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore ]

**Print** [ Records being browsed; All Pages; Orientation: Landscape; Paper size: 11.69" x 16.54"; Scaling: 35% ]  
[ Restore: HP LaserJet 2200 Dom iMac @ Dominic's iMac ]



**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: development::projectno

development::name  
site::name  
address site::townname  
organisation siteowner::name  
organisation local authority::name  
applicationstatusadvert::status  
development::advertappstatusmodified  
organisation developer::name  
development::wildstonedeveloperorcontact  
development::wildstoneplanner  
development::advertapplicationsubmitted  
development::advertapplicationregistration  
development::advertapplicationconsultationend  
development::advertapprefla  
development::planningapprefla  
development::advertapptargetdate  
development::advertappdeterminationdate  
development::advertappealdeadline  
development::advertappealsubmitted  
development::advertappealstart  
development::advertappealrefla  
development::advertappealprocedure  
development::advertappealrepresentations  
development::advertappealforecastdetermination  
development::advertappealhearing ]  
[ Format output using current layout ]

**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: developmenttype::type  
development::wildstonedeveloperorcontact  
site::name  
panelsize developmentdetail::size  
development::projectno  
developmentstatus::status  
development::leaseperannum  
development::purchaseprice  
development::feeproposal  
development::c\_profityear1  
development::developmentstatusmodified ]  
[ Format output using current layout ]

**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: development::name  
site::name  
development::projectno  
development::tenderstrategy  
development::tenderstatus  
development::tendermaterialsrequired  
development::tendermaterialscomment  
development::tenderrelease  
development::tenderdeadline ]  
[ Format output using current layout ]

**Export Records** [ Create folders:No; Character Set: "Unicode (UTF-16)"; Field Order: development::projectno

development::name  
site::name  
address site::townname  
organisation siteowner::name  
organisation local authority::name  
planningconditions::conditionnumber  
applicationstatusconditions::status  
planningconditions::conditionstatusmodified  
organisation developer::name  
development::wildstonedeveloperorcontact  
development::wildstoneplanner  
planningconditions::conditionapplicationsubmitted  
planningconditions::conditionappregistration  
planningconditions::condiitonapplicationconsultationend  
planningconditions::conditionapprefla  
planningconditions::conditionapptargetdate  
planningconditions::conditionappdetermindate ]  
[ Format output using current layout ]

*#Might not be in use*

**Enter Browse Mode**

**Go to Layout** [ "planning schedule" (development) ]

**Sort Records** [ Keep records in sorted order; Specified Sort Order: applicationstatusadvert::status; ascending ]  
[ Restore; No dialog ]

*#Might not be in use*

Enter Find Mode [ ]

Set Field [ development::FKdevelopmenttype\_id; 1 ]

Set Field [ development::FKstatus\_ID\_advertapplication; "" ]

Perform Find [ ]

**Sort Records** [ Keep records in sorted order; Specified Sort Order: organisation local authority::name; ascending  
address site::line1; ascending ]  
[ Restore ]

**Sort Records** [ Keep records in sorted order; Specified Sort Order: developmentstatus::statuscursustomorder; ascending  
address\_site::line1; ascending ]  
[ Restore ]



Commit Records/Requests

If [ ]

End If

## System Wide: Set Custom Menu

---

**If** [ Get ( AccountName ) = "Elizabeth Roy" ]

**Install Menu Set** [ "[Standard FileMaker Menus]" ]

**Else** [ ]

**Install Menu Set** [ "Wildstone Development Tracker" ]

**End If**



## System Wide: Sort List

---

**Set Variable** [ \$fieldname; Value:Get ( ScriptParameter ) ]

**Go to Object** [ Object Name: \$fieldname; Repetition: 1 ]

**If** [ \$\$Sort=\$fieldname & "A" //ASCENDING SORT ]

**Sort Records by Field** [ Descending ]

**Set Variable** [ \$\$Sort; Value:\$fieldname & "D" ]

**Else** [ ]

**Sort Records by Field** [ Ascending ]

**Set Variable** [ \$\$Sort; Value:\$fieldname & "A" ]

**End If**

**Go to Record/Request/Page**

[ First ]

**System Wide: Start Layout Script**

---

**Freeze Window**

**Go to Layout** [ “user\_log” (userlog) ]

**New Record/Request**

**Go to Layout** [ “lay\_home” (site) ]

**Set Window Title** [ Current Window; New Title: "Wildstone Development Tracker" ]

**Show/Hide Menubar**

[ Hide ]

**Show/Hide Toolbars**

[ Hide ]

**Show/Hide Text Ruler**

[ Hide ]

**Move/Resize Window** [ Name: "Wildstone Development Tracker"; Current file; Height: 665; Width: 1280 ]

**Adjust Window**

[ Resize to Fit ]

## System Wide: Window Close

---

**If** [ Get ( WindowName ) = "Wildstone Development Tracker" ]

**Loop** [ Flush: Always ]

**Set Variable** [ \$count; Value:ValueCount ( WindowNames ( Get ( FileName ) ) )]

**Close Window** [ Name: GetValue ( WindowNames ( Get ( FileName ) ) ; 1 ); Current file ]

**Exit Loop If** [ \$count = ValueCount ( WindowNames ( Get ( FileName ) ) )]

**End Loop**

**End If**

## System Wide: Open Dashboard if Closed

---

**If** [ IsEmpty(FilterValues(WindowNames;"My Dashboard")) ]

### Commit Records/Requests

**New Window** [ Style: Floating Document; Name: "My Dashboard"; Height: 500; Width: 895; Top: 50; Left: 200; Close: Yes; Minimize: Yes; Maximize: No; Resize: No ]

**Go to Layout** [ "lay\_taskdashboard" (jndevelopmenttasks) ]

### Show/Hide Menubar

[ Hide ]

### Show/Hide Toolbars

[ Hide ]

**Else** [ ]

**Select Window** [ Name: "My Dashboard"; Current file ]

**End If**

Portal Scrolling

---

Go to Object [ Object Name: "PhotoPortal"; Repetition: 1 ]

Go to Portal Row  
[ Select; Last ]

Commit Records/Requests





**Development Update Filter**

---

**Enter Find Mode** [   ]

**Set Error Capture** [ On ]

**Set Field** [ developmentstatus::status; "Contact made with land owner / agent" ]

**New Record/Request**

**Set Field** [ developmentstatus::status; "Head of Terms Agreed"]

**New Record/Request**

**Set Field** [ developmentstatus::status; "Contracts Exchanged" ]

**New Record/Request**

**Perform Find** [   ]

**Enter Find Mode** [   ]

[ Pause ]

**Set Field** [ development::developmentstatusmodified; Get ( CurrentDate )-14 & "..." & Get ( CurrentDate ) ]

**Constrain Found Set** [   ]

## Development Update Filter V2

---

**Enter Find Mode** [ ]

**Set Error Capture** [ Off ]

**Set Field** [ development::developmentstatusmodified; Get ( CurrentDate )-14 & "... " & Get ( CurrentDate ) ]

**Perform Find** [ ]

Enter Find Mode [ ]

Set Field [ development::tenderstatus; "Ongoing" ]

New Record/Request

Set Field [ development::tenderstatus; "Pre-tender" ]

New Record/Request

Set Field [ development::tenderstatus; "Complete" ]

Sort Records [ Keep records in sorted order; Specified Sort Order: development::tenderstatus; ascending ]  
[ Restore ]

Perform Find [ ]