Documentation

Getting Started with aXes Mobile for Developers

GETTING STARTED WITH AXES MOBILE FOR DEVELOPERS 3

GETTING STARTED 3
THINGS YOU CAN IMMEDIATELY DO WITH AXES MOBILE 6
ADDING A DOCUMENT LIBRARY 7
ADDING A WEBSITE 8
ADDING A SUBMENU 9
MULTI-DELETE, MOVE AND COPY OF MENU ITEMS 9
REARRANGING MENU ITEMS BY DRAGGING 11

DEVELOPMENT 12

MODIFYING AXES PROJECT TO USE FEATURES ON THE DEVICE 12
TAKING A PICTURE AND UPLOADING IT 15
READING A BAR CODE INTO AN IBM I FIELD 18
SENDING AN EMAIL WHEN A BUTTON IS TOUCHED 20
SENDING A TEXT MESSAGE (SMS) WHEN A BUTTON IS TOUCHED 20
PUTTING SCREEN DATA ON THE CLIPBOARD WHEN THE USER TOUCHES A BUTTON 21
SHARING DATA ACROSS ALL SESSIONS 22
PROGRAMMATIC SWITCHING TO ANOTHER AXES TERMINAL SESSION 23
GETTING YOUR CURRENT LOCATION 24
OTHER THINGS YOU CAN DO TO MODERNIZE YOUR EXISTING 5250 APPLICATION 25

APPENDICES 27

APPENDIX 1- EXPORTING MENUS AND MAKING A STANDARD MENU 27
APPENDIX 1A – EXPORT CONSIDERATIONS IN DETAIL 29
APPENDIX 1B – IMPORT CONSIDERATIONS IN DETAIL 29
APPENDIX 2 - SSL AND SELF SIGNED CERTIFICATES 30
APPENDIX 3 – CLEARING THE CACHE DURING DEVELOPMENT 33
APPENDIX 4 - DOCUMENT LIBRARIES – EXTRA DETAILS 34
APPENDIX 5 – AXES DATA EXPLORER 36

Content © LANSA Group
All Rights Reserved

Getting Started with aXes Mobile for Developers

Getting Started

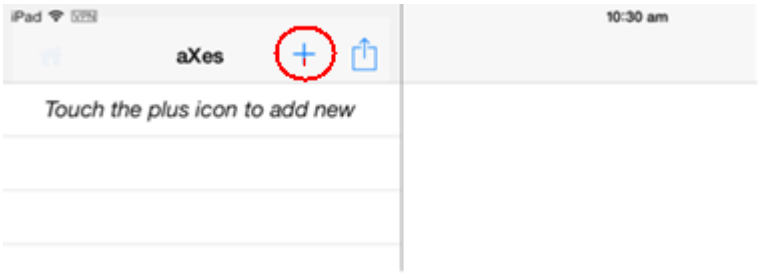
On your device (smartphone, tablet etc.) download and install the aXes Mobile App from the App Store or Play Store. (To find the application, search for 'aXes Mobile'). Then:

- Start the App.

Then create an aXes terminal session:

iOS

- Touch Plus.

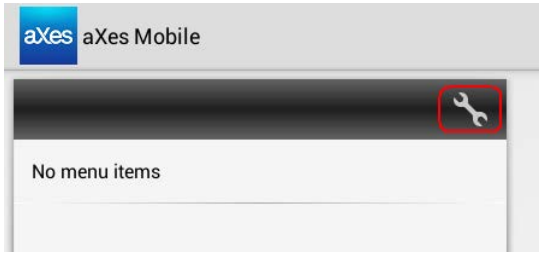


- Select Add aXes Terminal to add an aXes terminal session.

Android

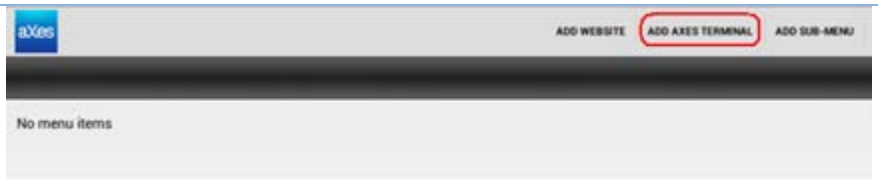
- Touch Overflow:

- Touch Edit Mode:

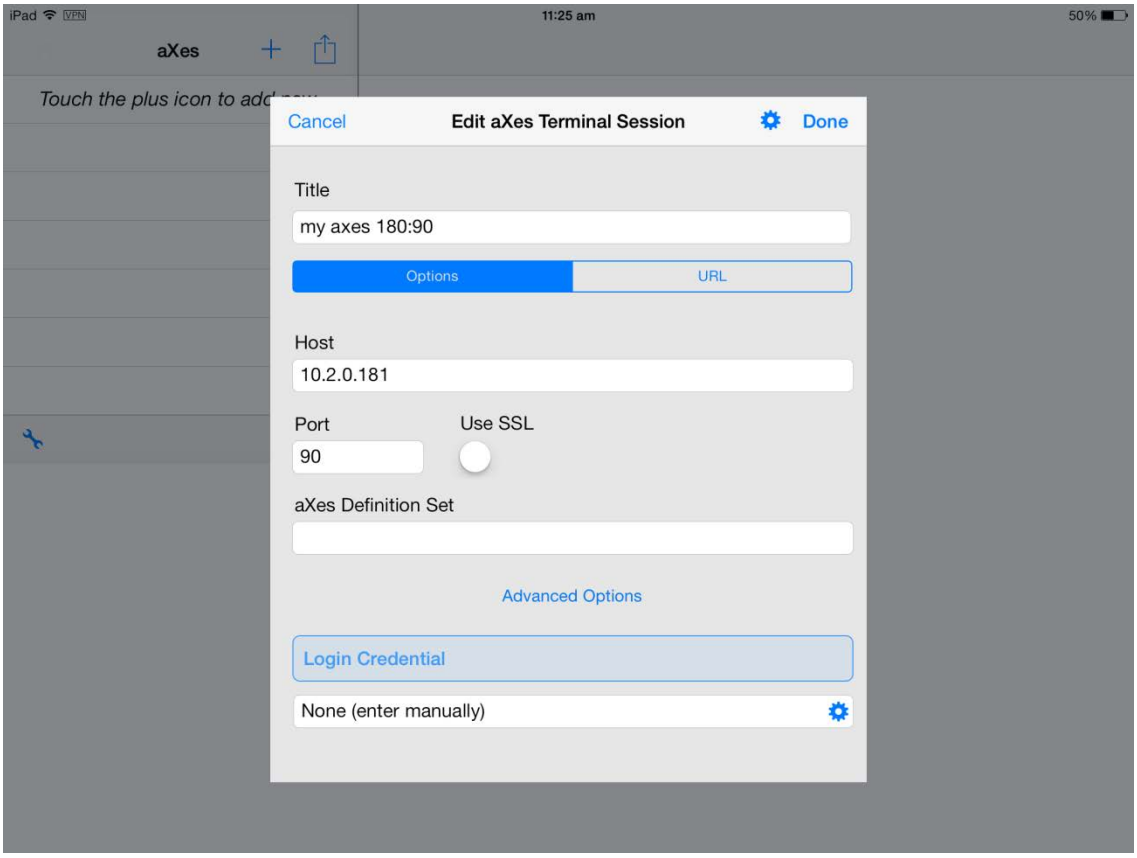


- Add aXes Terminal:

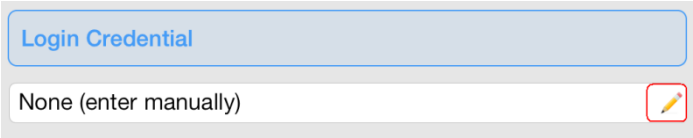
Content © LANSA Group
All Rights Reserved



In Edit aXes Terminal Session:

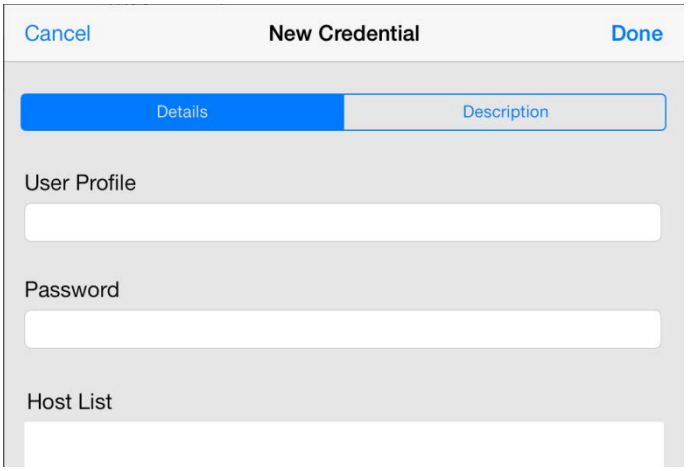


- Fill in the Title, Host, and Port. The host must be the ip address, or the fully qualified host name.
- If applicable, also fill in the aXes definition set (your aXes project) and, if applicable, select SSL.
- Below Login Credential, touch Settings:



- The New Credential screen should be displayed. If it is not, touch Plus to display it.
- Add an IBM i user profile and password:

Content © LANSA Group
All Rights Reserved



- On the Details tab, enter the user profile and password.

The Host List entry is automatically filled in for you.

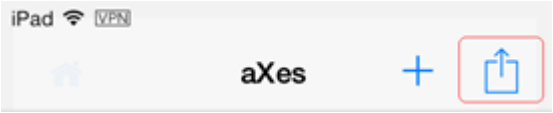
- Touch the Description tab and add a description and an identifier. (Make a note of the identifier – you will use it later as the "credential identifier").
- Touch Done twice to get back out.
- Select the aXes Terminal Session you have created. It will automatically sign on.

Note

If you want to check or edit the details of your aXes Terminal Session (or delete it):

iOS

- Touch the Overflow menu at the top:



- Choose Edit

Android

- Touch Edit Mode:

Then select your session to display or edit the details of the Terminal Session definition.

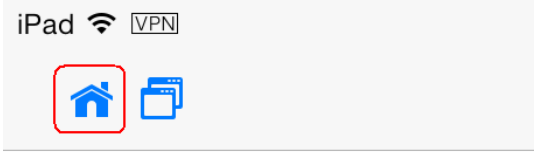
Content © LANSA Group
All Rights Reserved

Note

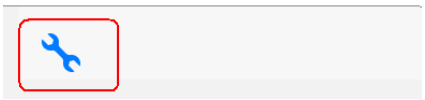
If you want to check or edit the user profile information:

iOS

- Touch Home:

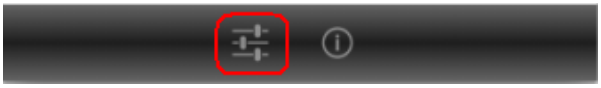


- And then Settings:



Android

- Touch Settings:



And then Manage Credentials, and select the credential you want to edit.

Things You Can Immediately Do with aXes Mobile

- You can leave multiple sessions open, and switch between them.

  You can switch sessions using the main menu, or the live sessions menu which is accessed from this icon:

  The advantage of having several live sessions open is that you don't lose your place and have to re-navigate. Switching is quick and easy.

- If you close a session, aXes Mobile will automatically sign you back on the next time you want to use it.

Content © LANSA Group
All Rights Reserved

- You can touch the function and Enter keys at any time – whether or not you are currently editing an input field.
- You can set up your aXes sessions, with appropriate user profiles, the way you want, and have them remembered.
- Alternatively, the administrator can set up a standard combination of aXes session definitions, websites, and document libraries and make it available to all users as a url on an email. In this way all users can start off with a standard aXes Mobile menu without any configuring.

For more details, see Appendix 1- Exporting Menus and Making a Standard Menu.

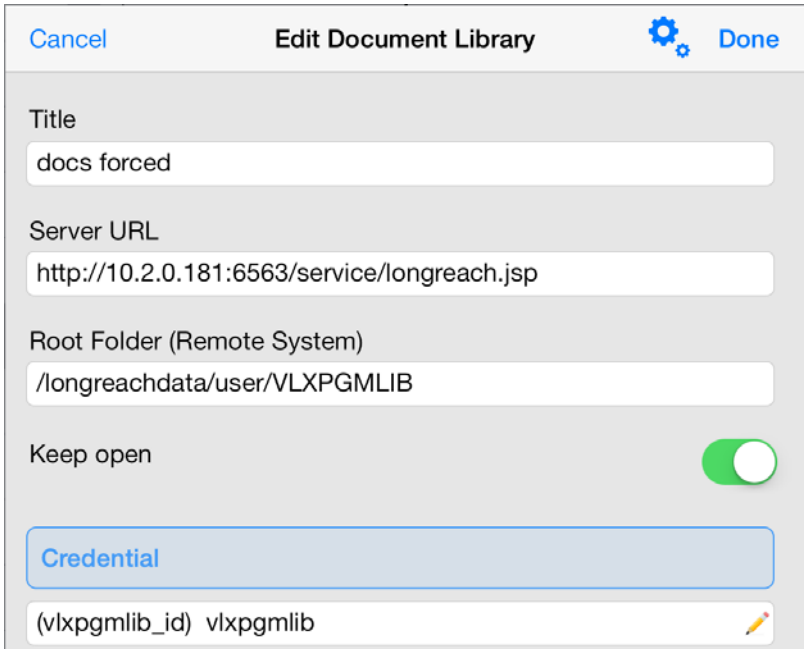
Adding a Document Library

A document library is a link to a directory on the IFS.

You can create and store notes, images etc. locally and then, when a connection is available to the IBM i, you can upload the local files. You can also download from the IFS, and synchronize your local folder with the IFS folder.

(This feature uses the Longreach product which must be installed and operational.)

- Touch Plus in the main menu (or Edit in Android), and select Add Document Library.



Server URL	This property has to be of the form: http://<<host>>:<<port>>/service/longreach.jsp where

Content © LANSA Group
All Rights Reserved

	<<host>> is the ip address of the host <<port>> is the port, and is usually 6563
Root Folder (Remote System)	This property needs to point to a location the user is authorized to, both on the IBM i and by Longreach authorization configuration. The default Longreach user profile is authorized to this directory on the IFS: /longreachdata/user/<<USER PROFILE IN CAPITALS>> The name is case-sensitive, so check that the name in the aXes Mobile Document Library setting matches what you see on the IFS on the IBM i.

Note: If you need to check the LongReach configuration settings, they are usually on the IFS in: \longreach\jsm\instance\system\httpd.xml.

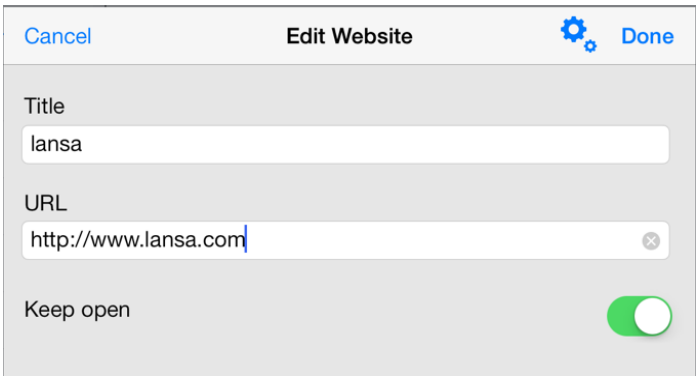
More details about LongReach configuration are available in the LongReach_Administrator_Guide_V1.2.pdf.

You can now refer to a library on the IFS, and download and upload files to it from the device.

Adding a Website

If the user commonly visits a website, you can add a menu item for it.

- Touch Plus (or Edit in Android), and select Add Website.



If the Keep open setting is on, the website remembers where the user was when the user switches between sessions.

If the Keep open setting is off, all websites which have this setting off will share one session. So the website *may still* remember where it was, unless the user switches to another website where the Keep open setting is off.

Content © LANSA Group
All Rights Reserved

Adding a Submenu

You can organize the terminal sessions, websites, and document libraries by adding a submenu.

- Touch Plus (or Edit in Android), and select Add Sub-menu.

Once the submenu is created, you can add items to it by touching it and then adding terminal sessions, websites, and document libraries.

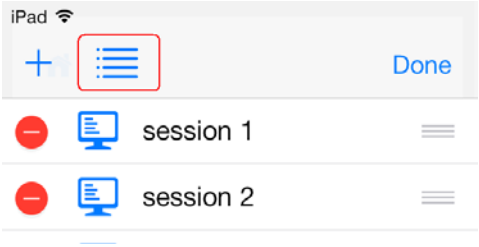
Or you can edit the menu where you want to move items from, use the multi-select option, and then the Move option to move items into your submenu. See the next section.

Multi-Delete, Move and Copy of Menu Items

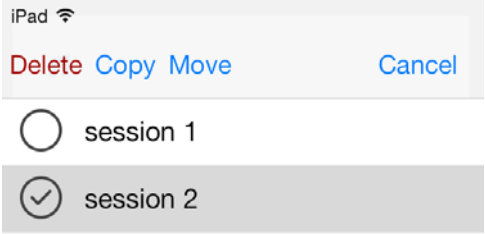
You can work with multiple menu items at once, and delete, move or copy them:

iOS

- Touch the Overflow menu and touch Edit.
- Then select this option:



- Then select the menu items you want to work with, and use Delete, Copy or Move:



Content © LANSA Group
All Rights Reserved

Android

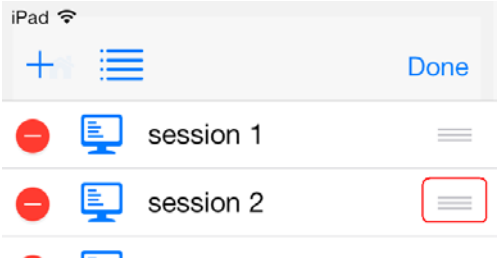
- Touch Edit Mode:

- Then select the menu items you want to work with, and use Delete, Copy or Move.

Content © LANSA Group
All Rights Reserved

Rearranging Menu Items by Dragging

(iOS only) In edit mode you can move menu items up and down by dragging the icon below:



Content © LANSA Group
All Rights Reserved

Development

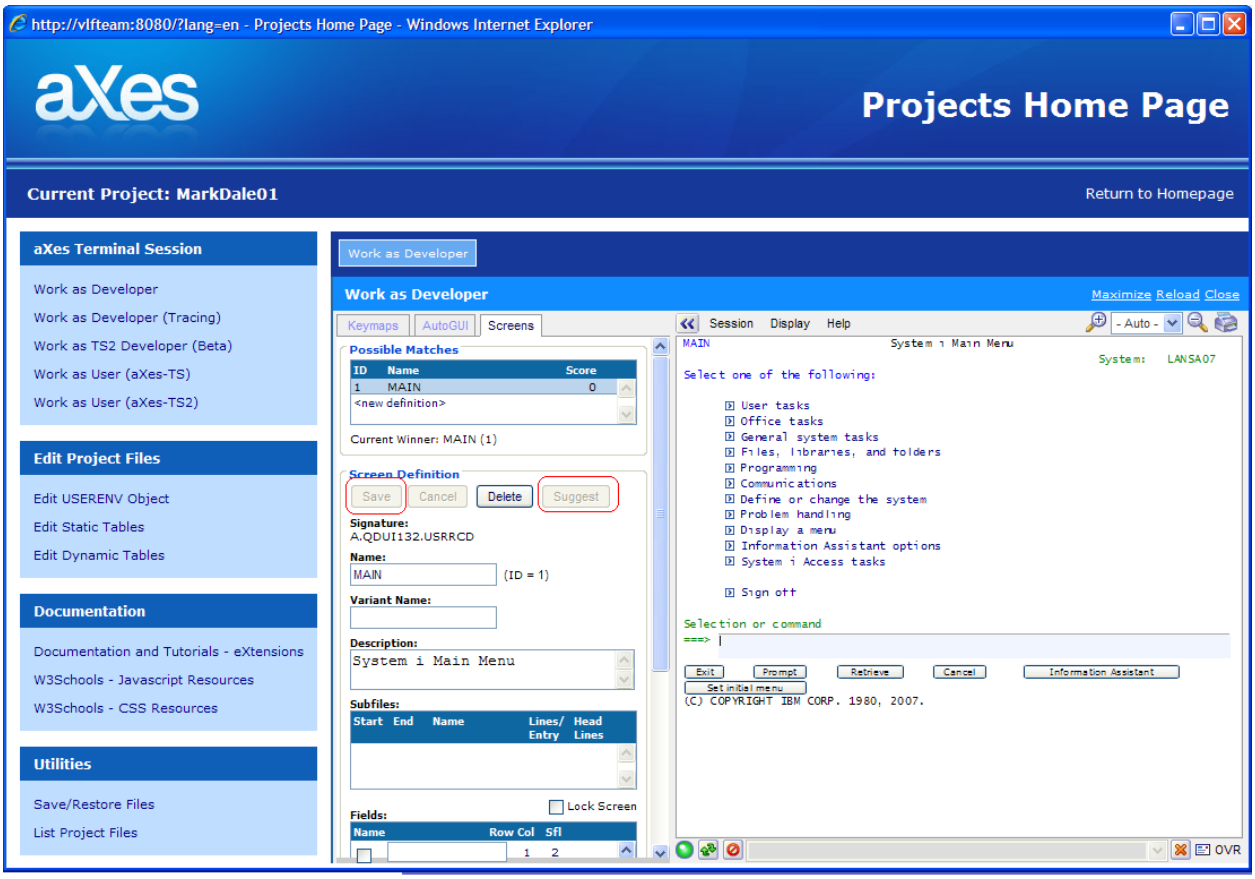
You can easily modify your aXes application to make use of the features on the device such as camera, bar code scanning, geolocation etc. You can also make the device upload files to the IFS, and store information locally.

Modifying aXes Project to Use Features on the Device

To modify your aXes project, you need a basic knowledge of aXes screen customization and development with extensions.

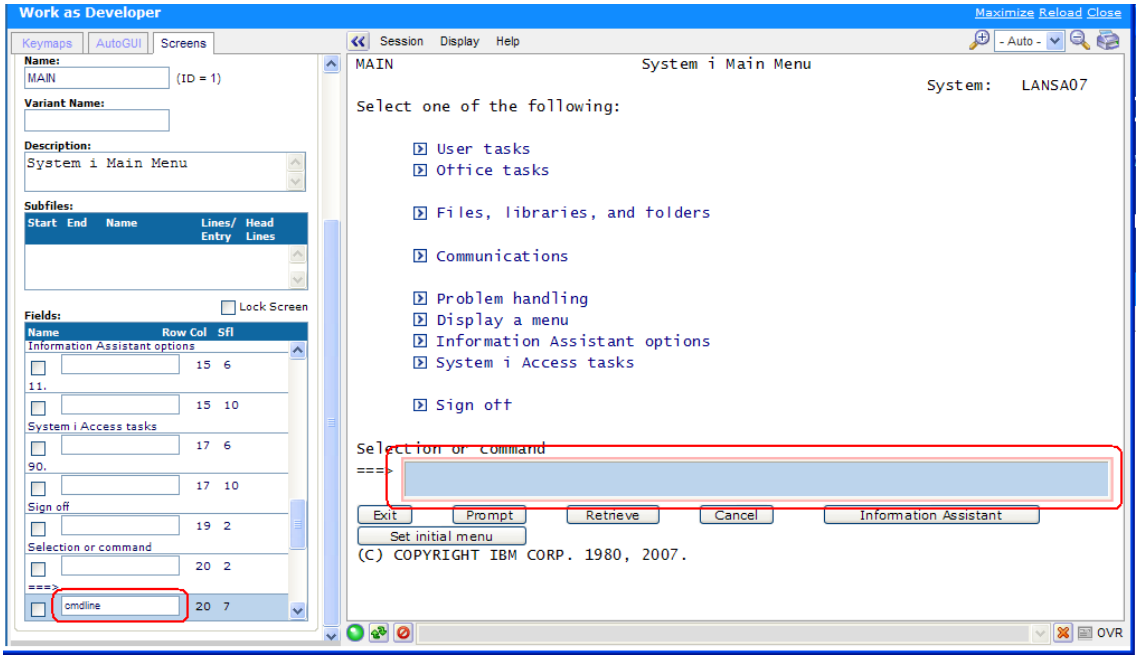
If you do not have this basic knowledge, do the aXes tutorials 0, 1, 2 and 5 in the aXes Extensions Guide.

- To start, create an aXes project for testing on your PC.
- Start the project in Developer mode and display a screen that you can add buttons to and that has at least one input field. You could, for example, use the main menu if it has a command line.
- Use the suggest button to identify the screen:



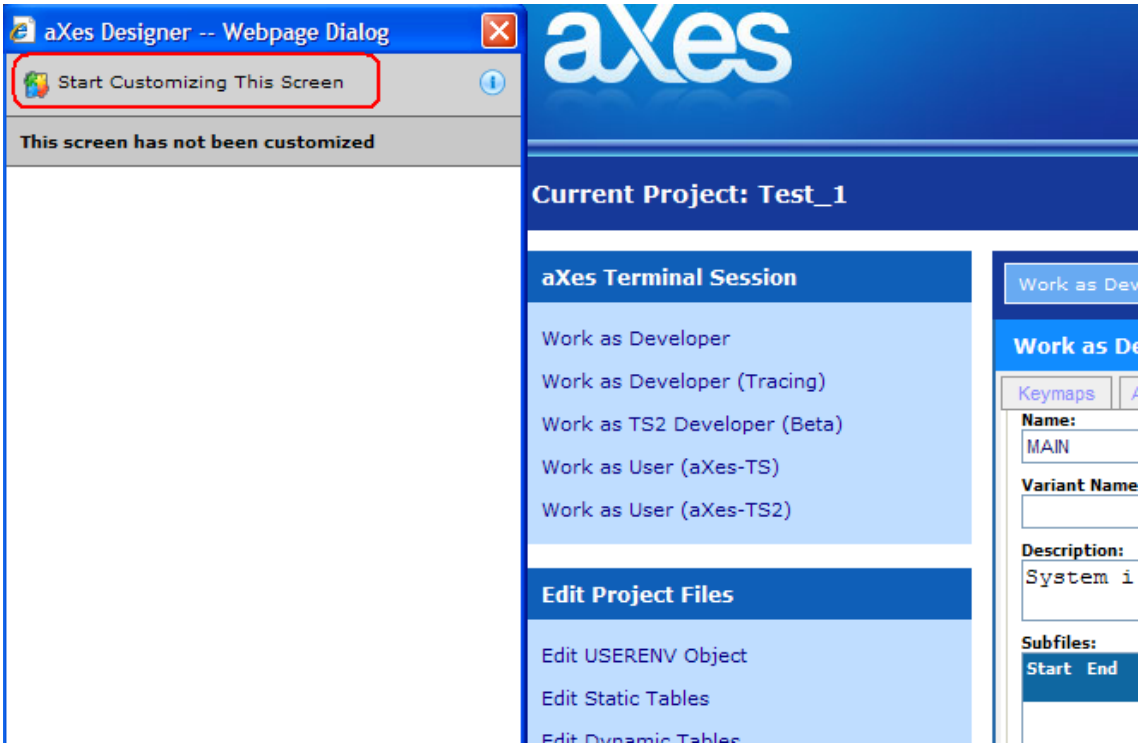
- And name an input field on it as "cmdline".

Content © LANSA Group
All Rights Reserved



- Save your changes.

You can now start customizing the screen.



Now, in the aXes app on your device (smart phone, iPad etc.):

- Touch Plus to add an aXes Terminal that points to your project. (Create the aXes Terminal the same way as you did before. Your project name goes in the "The aXes Definition Set" property.)

Content © LANSA Group
All Rights Reserved

- Check that you can start the session and navigate to the screen you want to use.

Note: When making changes to the aXes project it is usually sufficient to close the aXes Mobile Terminal session and restart it to pick up the project changes. But at some point you may need to clear the cache on the device – See Appendix 3 – Clearing the Cache during Development.

Content © LANSA Group
All Rights Reserved

Taking a Picture and Uploading It

You can modify your aXes project so that an aXes Mobile user can touch a button on an aXes screen, and make the device take a photo.

You could use this feature to, for example, allow employees that have a smart phone to update their own photos on the IBM i.

Taking a Picture

Start with a simple button that takes a picture and puts it in the Photos gallery on your device.

On your PC, in aXes developer mode:

- Go to your screen, edit the screen, and add a new element.
- Make the new element a push button element (extension), with an onclick script like this:

```
var options =
{
        imageName : "Photo_abc.jpg",
        annotate : true,
        width: "300" ,
        height : "300",
        saveToGallery : true
};


try
{
        LRSV.imageBasic(options);
}
catch (Error) { alert("problem setting photo name")};
```

Note how the parameters for the imageBasic service are passed as properties of an object named options.

You could also write the above in this way, if you prefer:

```
LRSV.imageBasic({ imageName : "Photo_abc.jpg",  annotate : true, width: "300" ,
height : "300",  saveToGallery : true });
```

LRSV stands for LongRange Services. It is the short way of writing LONGRANGE.Services.

So you could also write the statement this way:

```
LONGRANGE.Services.imageBasic({ imageName : "Photo_abc.jpg",  annotate : true,
width: "300" ,  height : "300",  saveToGallery : true });
```

Content © LANSA Group
All Rights Reserved

Next try out the button in the aXes App on your device:

- Run your aXes project and navigate to your chosen screen.
- Locate the button you added. If you touch it you should be able to take, and annotate, a picture.
- Minimize the aXes App, and check that the photo has been added to the device's gallery of photos.

Note: If you are not sure that you are seeing the latest version of your button script on your mobile device, you can check it by adding a line like this at the start of the script:

```
alert("version 1");
```

Then when you run your aXes project in your aXes Mobile app and touch the button, you will see the alert message.

Uploading the Picture

Extend the example to make the device take the photo AND upload the photo to the IFS on the IBM i.

You must have LongReach installed and working to be able to do this. The credential being used for the terminal session must have the LongReach URL in its Host List to do the upload.

On your PC, in aXes developer mode:

- Add another push button to your aXes screen with an onclick script like this:

```
var config =
{
   serverUrl: "http://<<ip address>>:<<port>>/service/longreach.jsp",
   credential: "<<credential identifier>>
};
LONGRANGE.FileTransfer.setDefaultConfig(config);
```

Where:

<<ip address>>	The ip address of the IBM i you want to upload to
<<port>>	The port for the LongReach server (usually 6563)
<<credential identifier>>	The identifier of the user profile you have defined earlier. If you want to look up the credential identifier, it is under Settings → Manage Credentials → Description tab.

Content © LANSA Group
All Rights Reserved

This tells LONGRANGE which server to upload to and what user profile to use.

- Then add the script to take the picture and do the upload using imageUpload:

```
var options =
{
   imageName : "Photo_abc.jpg ",
   annotate : true,
   toIFSfolder : <<IFS location>>
};


try {
     LRSV.imageUpload(options);
}
catch (Error) { alert("problem uploading photo"); }
```

where <<IFS location>> is the directory on the IFS where you want to upload to.

The location that you upload to has to be in a location that the LongReach configuration file authorises you to access. Usually you are authorised to:

"/longreachdata/user/<<your user profile>>/"

- So change the IFS location to, for example, "/longreachdata/user/MYPROFILE/"

Note: If you need to check the LongReach configuration settings, they are usually on the IFS in: \longreach\jsm\instance\system\httpd.xml.

More details about LongReach configuration are available in the LongReach_Administrator_Guide_V1.2.pdf

- Make the button caption "Take and Upload a Photo".
- Save your changes.

Now, in aXes Mobile on the device:

- Sign off in the session you were in, and then start the session again.
- Navigate to your screen, and touch the button. You can now take and annotate a picture.

When you have finished, the picture will be automatically uploaded to the IFS in the /longreachdata/user/<<your user profile>>/ directory.

- Check on the IFS that your image has been uploaded.

Content © LANSA Group
All Rights Reserved

Reading a Bar Code into an IBM i Field

Reading a Bar Code

Suppose your company has barcoded products, and you want your employees to be able to look up on the IBM i the details of a product (such as identifier, model number, dimensions or stock levels).

You can set up the aXes screen so that the camera of the device reads the bar code and feeds the number into an IBM i field.

In this example we will use the field that you previously identified as cmdline.

On your PC, in aXes Developer mode:

- Add a button to your aXes screen with an onclick script like this:

```
try
{
    LRSV.scanPaste( { pasteTo : "cmdline" } );
}
catch (Error) { alert("Error using scanPaste service: ", Error.message); }
```

- Make the button caption "Scan a barcode".
- Save your changes.

On your smartphone:

- Run your aXes project in your aXes Mobile app and touch the button.
- Point the camera of the device at the bar code. When the camera recognises it, the bar code number is loaded into the cmdline field.

Entering the Number into an IBM i Field

You might want to modify this example so that the number would be sent to a product identifier field in your own application, and the Enter key would be pressed. However, you cannot do this because *LongRange Services (LRSV) are asynchronous*, which means you can't make use of the barcode value by directly scripting anything immediately following this line:

```
LRSV.scanPaste( { pasteTo : "cmdline" } );
```

In order to code something to happen after the bar code value is returned, your script needs to wait for the scanPaste to finish, and then use the value if the scan was successful.

The way to do this is to use a callback function. A callback function is a property of the options object. It executes when the scanPaste finishes:

```
if (window.LRSV)
{
```

Content © LANSA Group
All Rights Reserved

```
/* Using a callback function to alert and press Enter */


var callBackFunction = function (result)
{

  if ( result.success )
  {
    alert("callback function running. userInfo was:" + result.userInfo);
    SENDKEY(ENV.defaultKey);
  }
  else
  {
    alert("there was an error:" + result.message);
  }

};


var options =
{
  pasteTo : "cmdline",
  callBack : callBackFunction
};

try {
    LRSV.scanPaste( options );
}
catch (Error) { alert("Error using scanPaste service: ", Error.message); }


}
```

The part in blue is the callback function. In this example of a callback function, the user touches the button, the barcode scanning and pasting occurs at some point, *and then when it is finished*, the Enter key on the aXes screen is pressed.

Note that the callback function can also detect if an error occurred, and the result.message value will indicate what the error was.

Note also that it is a good idea for scripts to check whether LongRange Services are available before they start using:

```
if (window.LRSV)
```

Content © LANSA Group
All Rights Reserved

This is because the aXes script may be running in an environment like a PC where LongRange Services are not available, or even if it is running in aXes Mobile, the LongRange services may not yet be ready at the time the button is pressed.

Sending an Email when a Button Is Touched

On your PC, in aXes Developer mode:

- Add a button to your aXes screen with an onclick script like this:

```
if (window.LRSV)
{

var options =
{
  text      : ( "the command line contains:" + FIELDS("cmdline").getValue() ),
  subject   : "This is my email subject",
  recipient : "myEmailAddress@myCompany.com"
};

LRSV.sendEmail( options );

}
```

(Replace myEmailAddress@myCompany.com with your own email address)

- Make the button caption "Send an Email".
- Save your changes.

On your device:

- Run your aXes project, add some text to the cmdline field, and touch the button.

An email with the text from the command line is sent.

Sending a Text Message (SMS) When a Button Is Touched

On your PC, in aXes Developer mode:

- Add a button to your aXes screen with an onclick script like this:

```
if (window.LRSV)
{
```

Content © LANSA Group
All Rights Reserved

```
var options =
{
  messageText : ( "the command line contains:" + FIELDS("cmdline").getValue() ),
  recipient   : "123456789"
};

LRSV.sendTextMessage( options );

}
```

(Replace 123456789 with your own mobile number)

- Make the button caption "Send an SMS".
- Save your changes.

On your device (smart phone or tablet if it can send text messages):

- Run your aXes project in aXes Mobile, add some text to the cmdline field, and touch the button.

An email with the text from the command line is sent.

Putting Screen Data on the Clipboard When the User Touches a Button

If the user wants to put data from a screen field onto the clipboard and allow it to be pasted onto other screens or other applications, they can always use the standard Copy and Paste.

However, sometimes you might want to make it easier for the user to copy data from a group of fields to the clipboard.

On your PC, in aXes Developer mode:

- Add a button to your aXes screen with an onclick script like this:

```
var clipText = FIELDS("cmdline").getValue();

var items = [ { text : clipText } ];

var options = { arrayItems:items } ;

try { LRSV.copyToClipboard(  options  ); }

catch (Error) { alert("problem in clipboard " + Error.message ); }
```

Content © LANSA Group
All Rights Reserved

- Make the button caption "Copy cmdline to Clipboard".

On your device:

- Run the aXes Mobile App and go to your screen, type a value into the cmdline field and touch the button. aXes will put the data on the clipboard.
- Paste what was in the command line into a field on another aXes screen, or into another application on the device, like notes or the email client.

If you want the user to be able to copy multiple fields to the clipboard you can format the pasted data in various ways.

By concatenating field data with tabs "\t" and newlines "\n", you can output the data in a way that suits the user when it is pasted.

To copy multiple fields at once, you can send multiple field data in "items" like this:

```
var clipText1 = FIELDS("field1").getValue();
var clipText2 = FIELDS("field2").getValue();
var clipText3 = FIELDS("field3").getValue();

var items = [ { text : clipText1}, { text : clipText2}, { text : clipText3} ];
```

Sharing Data across All Sessions

Say the user has located a customer they want to work with on aXes Terminal Session A and they now want to work with purchase orders for that customer.

They have purchase order maintenance open in aXes Terminal Session B. What we need is a way to get the selected customer id over to Session B.

If the customer id is in a field named myCustField in aXes in Session A, the button script to make it available to all sessions is:

```
var options =
{
  name: "Shared_customer",
  value: FIELDS("myCustField").getValue()
};

LONGRANGE.Form.setSharedState(options);
```

In session B, a button script can get that data under the name "Shared_customer", and write it to a field named myCustField_B on the purchase order maintenance screen like this:

```
var options =
{
   name: "Shared_customer",
```

Content © LANSA Group
All Rights Reserved

```
  onCompleted: function(obj) { FIELDS("myCustField_B").setValue(obj.result); }
};

LONGRANGE.Form.getSharedState(options);
```

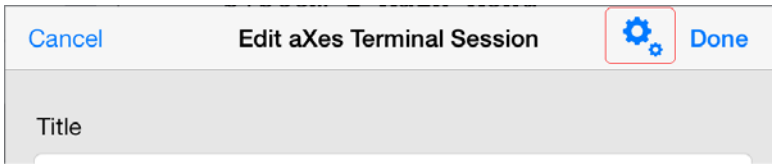
The session B button script could also be coded to press a function key or Enter, and do the purchase order search using the customer number from session A.

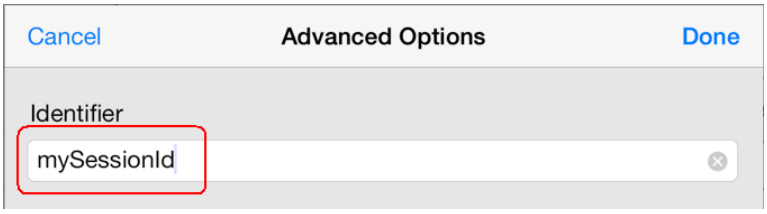
Programmatic Switching to Another aXes Terminal Session

To programmatically switch to another aXes terminal session you need to know the identifier of the aXes session you want the user to be switched to.

On your device:

- Start the aXes-App, and go to Edit Mode.
- Edit the details of the session you want to switch *to*
- Touch Settings for that session:





- Ensure that your Switch-to session has a unique identifier (e.g. mySessionId)
- Save your changes by touching Done twice.

On your PC, in your aXes project:

- Locate the switch-from screen and ensure it is in Developer mode.
- Add a button that does this:

```
var options =
{
    name:"mySessionId",
    replaceSession: false // Specifies whether open session should be closed first
};
LONGRANGE.Form.open(options);
```

Content © LANSA Group
All Rights Reserved

On your device:

- Start the aXes app.
- Go to the switch-from session, and the switch-from screen, and touch the button you created.

You will see the switch-to session start up automatically.

Getting Your Current Location

On your PC, in aXes Developer mode:

- Add a button to your aXes screen, with an onclick script like this:

```
if (window.LRSV)
{

var callBackFunction = function(result)
{
  alert("Longitude: " + result.userInfo.LONGITUDE + " Latitude: " +
result.userInfo.LATITUDE + " Altitude: " + result.userInfo.ALTITUDE  );

};


var options =
{
  callBack   : callBackFunction
};

LRSV.getGeolocation( options );

}
```

- Make the button caption "Get my current location".
- Save your changes.

On your device:

- Run your aXes project in aXes Mobile on your device, and touch the button.

You will see your current location:latitude, longitude, and altitude. (This is the location of the device).

Content © LANSA Group
All Rights Reserved

Other Things You Can Do to Modernize Your Existing 5250 Application

You can:

- Include maps when displaying location information – It is possible to show an html window that displays a google map of a location whenever the IBM i displays a street address. This can give the end-user a better idea of where something is, or even directions on how to get there. If Google street view is available, the user can also get an idea of what the location looks like on the ground.

- Make navigation much easier, especially for mobile phones and iPads, by adding large buttons for common navigation tasks. These buttons can set fields to values, and/or execute commands in the command line, and press any key.

- Allow users to work off-site using a mobile phone, or a phone enabled iPad.

- Allow users to download and store relevant information on the device when they are off-site and cannot get an internet connection.

- Add graphs to show the screen data in a more meaningful way.

- Switch to another session and pass information to it:

  Say the user has located which customer they want to work with on aXes Terminal Session A.

  So if, say, Session B shows the purchase order search screen, the screen can be set up to automatically load the selected customer from Session A into the customer search field, and press Enter, and display the purchase orders for that customer.

  You could use a separate Session and project for each kind of switch – One session for Show_Purchase_Orders_for_a_Customer, one session for Show_Accounts_for_Customer etc.

  Or you could set up one session and aXes project as the common point for all switches, and pass it the required action and identifier data, and make it handle it. Whenever switching to that session, you would tell it to close and restart the session if it was already open. In this way it would always start on the same screen after the switch.

  If your IBM i application has a way of navigating to different parts of the system from the command line, you could put a script on a timer on the screen to write a call to the command line and press Enter to go where the user needs to go.

```
switch( myAction )
{
case "SHOW_ORDER":
```

Content © LANSA Group
All Rights Reserved

```
/* Show the orders for a customer */
FIELDS("commandline").setValue("call pgm(ShowOrder) parms('" + myId + "')");
SENDKEY("Enter")
break;

case "SHOW_ACCOUNT":
/* Show the accounts for a customer */
FIELDS("commandline").setValue("call pgm(ShowAccount) parms('" + myId + "')");
SENDKEY("Enter")
break;
```

Content © LANSA Group
All Rights Reserved

Appendices

Appendix 1- Exporting Menus and Making a Standard Menu

A site administrator can allow all device users to start off with the same aXes Mobile menu. To make a standard menu available to multiple users:

In the aXes Mobile app on your device:

- Set up your own menus the way you want all users to be set up. (This can include definitions of credentials – the passwords will not be exported.)

iOS

- Open the Overflow menu:

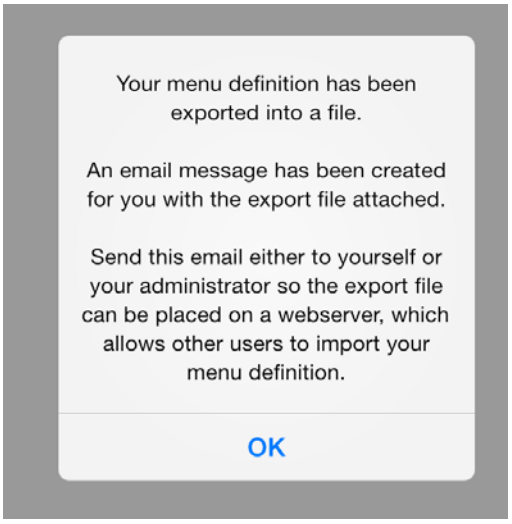
Android

- Go to Edit Mode:

- Select Export.

A new email is created, and over it a message comes up explaining that the aXes Mobile app has created a menu-export as an attachment to an email:

Content © LANSA Group
All Rights Reserved



- Touch OK, and then finish the email and send it to yourself.
- On a PC where you are able to save files to a webserver, receive the email.
- Save the email attachment file (menu-export.dat) to a location all your device users can access with a url, but preferably not somewhere that the general public can access it.

  For example, if you have a VPN, you could save it to a webserver directory that is only visible when connected to the VPN.
- Now create a url that points to the menu-export.dat file:

  http://<<host>>:<<port>>/myDirectory/menu-export.dat

  Email this url to your aXes Mobile app users, with some instructions like this:

  1) Start the aXes Mobile app on your device.

  2) Open the Overflow menu:

  3) Select Import

  4) Import from this url:

  http://<<host>>:<<port>>/myDirectory/menu-export.dat

Optional

You can extend this method to make a url that is even simpler for your users to use.

Say you have a url like this:

http://10.2.0.181:81/IMAGES/Private_ABC/menu-export.dat

If you url encode the url it becomes this:

http%3A%2F%2F10.2.0.181%3A81%2FIMAGES%2FPrivate_ABC%2Fmenu-export.dat

Content © LANSA Group
All Rights Reserved

If you embed the encoded url within this:

axes://importusrmenu/?url=xxxx

you get:

axes://importusrmenu/?url=http%3A%2F%2F10.2.0.181%3A81%2FIMAGES%2FPrivate_ABC%2Fmenu-export.dat

You can now email this url to your users, and provided they have installed the aXes Mobile app on their device, they can touch the url in an email, and the aXes Mobile app will automatically do the import.

Note: There are various websites on the internet you can use to url encode your url. To find them, Google "url encode".

Note: When sending to iOS users, the axes:// prefix will be recognised in the email.

If sending the email to android users you need to wrap the url in:

 axes

This will be recognised by the standard Android email client.

For more details about export and import see Appendix 1a – Export Considerations in Detail and Appendix 1b – Import Considerations in Detail.

Appendix 1a – Export Considerations in Detail

When you choose the Export menu option, the current menu will be packaged and attached to an email.

Remember that you will be exporting the menu (including sub-menus) you currently see (that is, if you are in a sub-menu, you will NOT be exporting the whole menu structure from the root).

As previously mentioned, you can assign an Identifier to a menu item. This Identifier will be used to match the menu items during the import process, which allows menu items to be replaced (as opposed to the menu items being added to the menu).

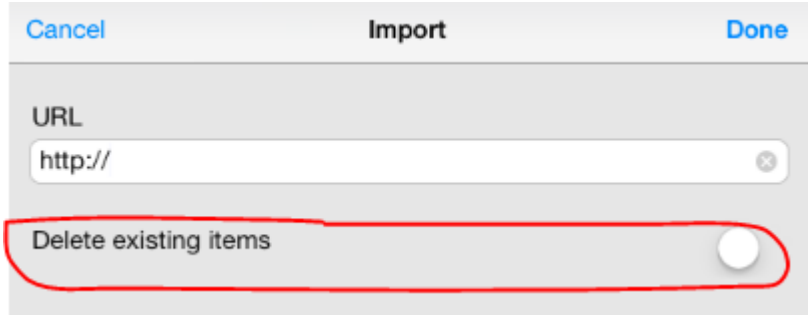
If exported menu items are associated with some credentials, the name and description of the credentials will be exported as well (but not the specific information such as username and password).

In order for a credential to be exportable, an identifier must be assigned to it. If an identifier is not assigned, you won't be able to export a menu that uses that credential, and you will get an error when you try to export

Appendix 1b – Import Considerations in Detail

The user importing the menu structure can also indicate if existing menu should be deleted prior to importing the menu:

Content © LANSA Group
All Rights Reserved



As with the Export, the Import will import the menu structure into the current menu (the menu you currently see).

As previously mentioned, you can assign an Identifier to a menu item. This Identifier will be used to match the menu items during the import process, which allows menu items to be replaced (as opposed to the menu items keep getting added to the menu).

Credentials will be imported as well. However, they will NOT contain any username or password.

NOTE: re-importing menu will overwrite the credentials, and end-users will need to re-enter username/password after import.

After import, a list of incomplete credentials will be presented to users so they can complete the credentials if they wish to do so.

Appendix 2 - SSL and Self Signed Certificates

When creating an aXes terminal session, you have the option of enabling SSL.

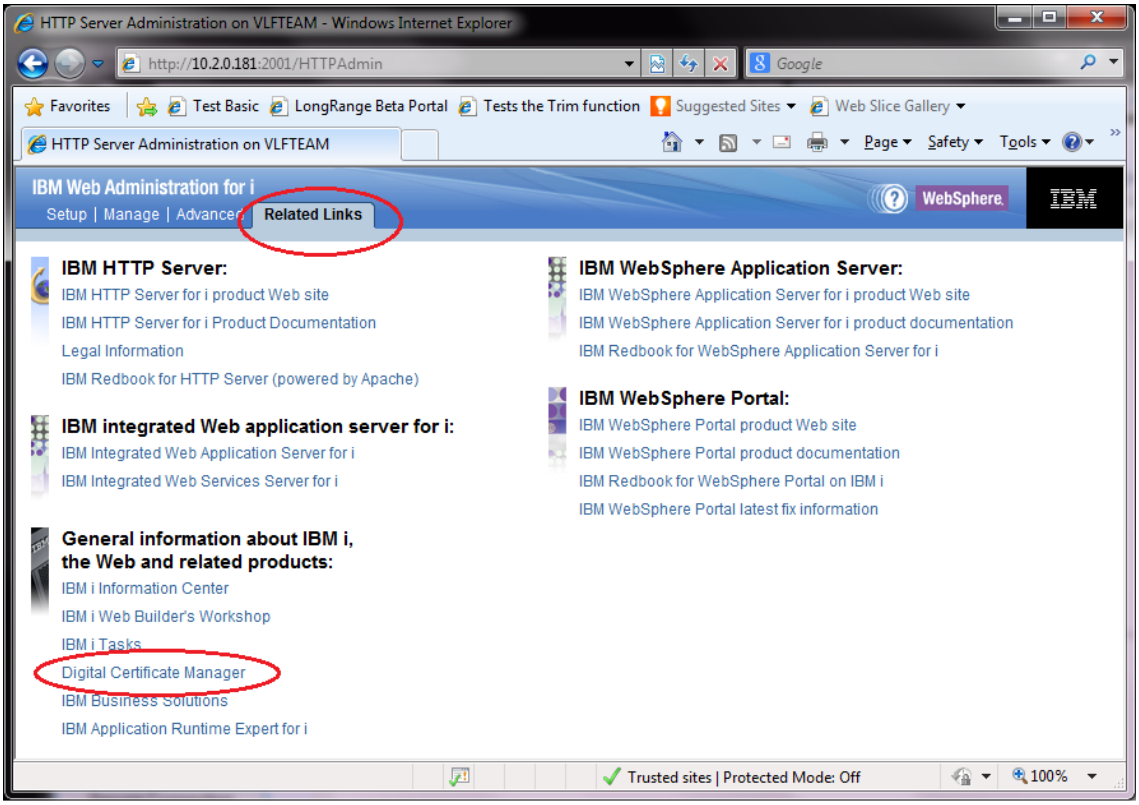
For SSL to work, you either need a valid certificate from a certificate authority on the server, or you need to use a self-signed certificate.

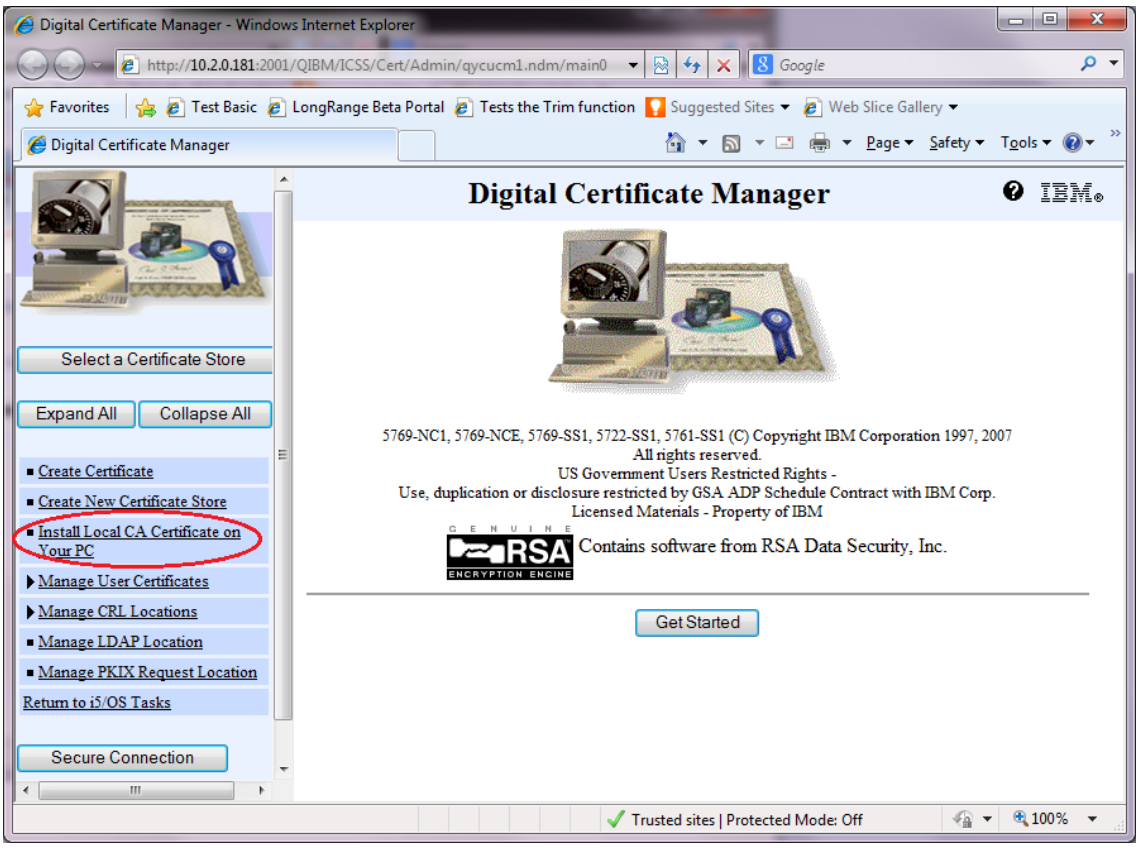
If you want to use a self-signed certificate with iOS7, you also need to install the certificate on each device. This is done by emailing a .cer file as an attachment to each device.

The methods of generating a .cer file for a self-signed certificate vary, but the following screen shots help understand the general process.

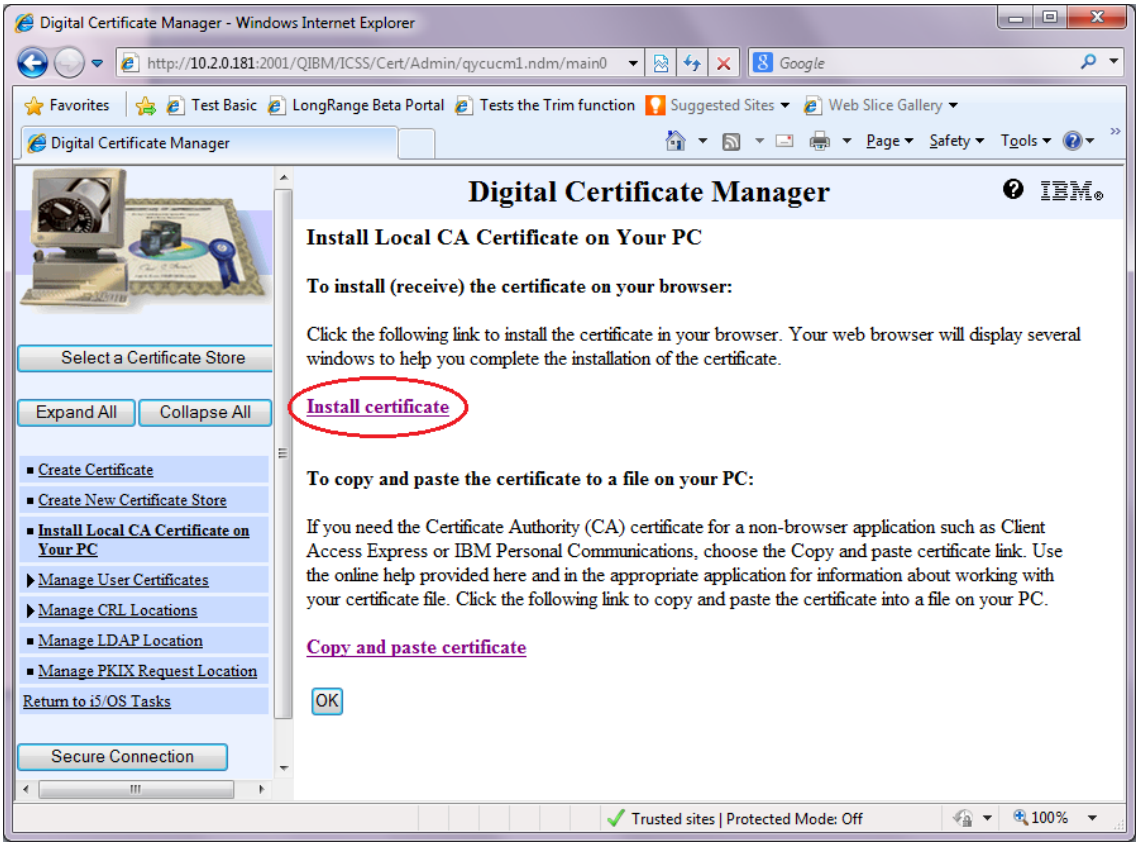
Using the related links tab on the http administrator:

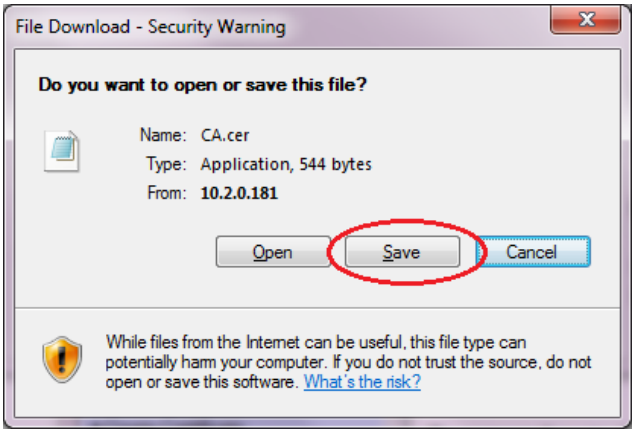
Content © LANSA Group
All Rights Reserved





Content © LANSA Group
All Rights Reserved



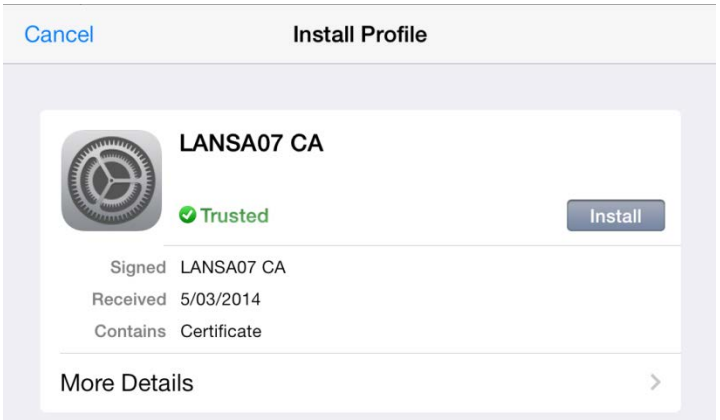


Now you have a file CA.cer you can email to the devices that want to use the SSL connection.

On the device, receive and view the email and touch the attachment on it:

Content © LANSA Group
All Rights Reserved

Then install the certificate:



You can work with and remove certificates in the device settings → General → Profiles.

Note that the host the self-signed certificate is created for must exactly match the host on the aXes terminal session definition.

For aXes terminal sessions, the host must be the fully qualified name, not just a name that gets resolved to the host address.

Note that the correct certificate type is PEM (.cer), not PFX.

Appendix 3 – Clearing the Cache during Development

At some point during your aXes development for aXes Mobile, you may need to clear the cache on the device.

To do this:

- Touch Settings in aXes Mobile:

iOS

Android

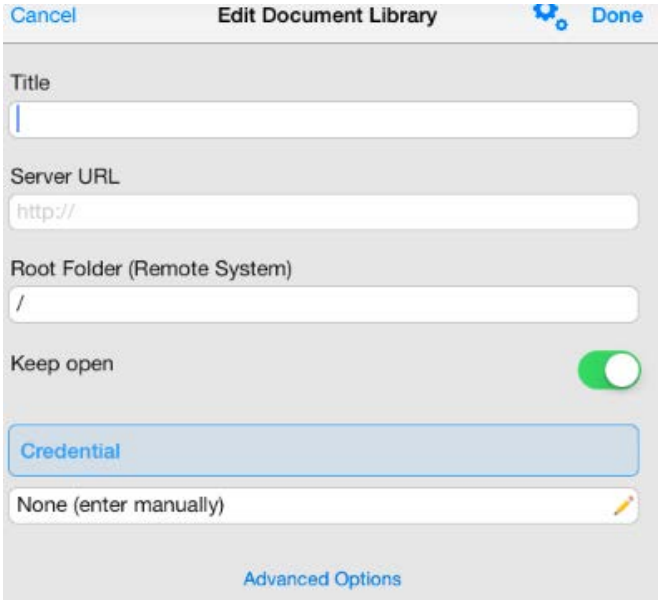
- → Development → Clear Cache.
- Then completely close aXes Mobile.

On an iPad, for example, you need to exit from the aXes Mobile app, and then work with active applications and end the aXes Mobile App.

When you start the aXes Mobile app, the cache will be cleared.

Content © LANSA Group
All Rights Reserved

Appendix 4 - Document Libraries – Extra Details

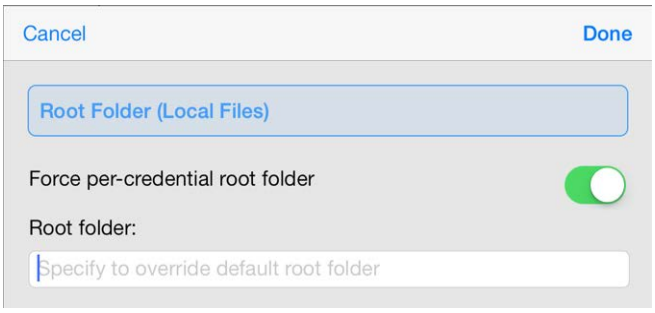


Server URL	Specifies the LongReach service URL (for example, http://10.2.0.181:6563/service/longreach.jsp).
Root Folder	Specifies the initial folder (on the remote system) that this document library should connect to. You can also use two special values when specifying Root Folder: {BASE} → specifies the user's home directory on the server {SHARED} → specifies "shared" or "public" home directory on the server. LongReach calculates the effective value of BASE and SHARED based on settings in the LongReach configuration file. {BASE} and {SHARED} can be followed by path representing sub-directories. For example: {SHARED}/Photos However {BASE} and {SHARED}, if used, must appear at the beginning of the path, and it should not be preceded by '/'. For example, this is NOT valid: "/{BASE}"
User (as specified in the Credential)	The user must have been granted permission (on the server) to access the specified folder.

Content © LANSA Group
All Rights Reserved

	User permissions are specified in the LongReach configuration file. See LongReach documentation for more details.

Advanced Details



Force per-credential root folder	By default (when this option is OFF), multiple users (credentials) will share the same local repository of files. An exception to this is when {BASE} special folder is specified, in which case the behaviour is the same as when this option is turned on. When this option is ON, each user (credential) will have their own local repository of files (ie. separate root directory, one per user). Note that this setting only affects how files are stored locally on the device (this has nothing to do with files on the server).
Root folder	By default, when you define a Document Library menu item, the path to the documents are obscure and hard to deduce. This can become a problem when you are scripting to access files (get/put) on the document library storage folder. The Root folder option allows you to “fix” the root directory of your documents on the device. You can set the “Root folder” setting above to point to your own directory, for example “/MyFiles”. Your script can then refer to the documents in the “Document Library” as “/MyFiles/xxxxxxx”.

Content © LANSA Group
All Rights Reserved

Appendix 5 – aXes Data Explorer

- Add a Website to aXes Mobile and use a URL with the format:
  HTTP://<SERVER>:<DE Port>/axes/database-manager.html?lang=<language code>.

  Default Data Explorer port is 5563 and language code is 2 characters.

  For example:
  http://10.2.0.181:5563/axes/database-manager.html?lang=en

Content © LANSA Group
All Rights Reserved