finPOWER Connect 3
Account Applications

Version 3.08

10th February 2021

Table of Contents

Disclaimer 5
Version History 6
Introduction 7
  Important 7
Account Applications Overview 8
  User & Responsibilities 10
  User Preferences 11
  Global Settings 12
  Task Manager 13
Concepts 14
  No Page Set or other Custom User Interface 14
  Page Sets 14
  Web Pages 14
  Key Locking 14
  Temporary Account Object 15
  Temporary Client Object 15
Account Application Types Form 16
  General 16
  Options 17
  Quotes 18
  Applicants 19
  Collateral 21
  Data Schema 22
  Data Capture 23
  Script Code 24
  Constants 25
  Documents 26
  Permission 27
Account Applications Form 28
  General 30
  Quotes 32
  Applicants 34
  Collateral 36
  Data Capture 37
  Workflows 38
  Logs 39
  Files 40
  Audit 41
Account Application Sub-Forms 42
  Quote Properties 42
  Edit Financial Wizard 44

Applicant Properties 45
Applicant Wizard 47
Collateral Item Properties 48
Collateral Item Wizard 50
Script Objects 51
finAccountApp 51
finAccountAppApplicant 51
finAccountAppCalculation 51
finAccountAppCollateralItem 51
Account Application Type Script 52
Overview 52
Creating an Account Object 52
Script Events 53
Initialise 54
InitialiseAccountApp 55
SavePre 58
SaveAudit 59
SavePost 60
GetAccountType 61
UpdateAccount 63
UpdateAccountCalc 64
AfterCalculationUpdateFromAccount 66
UpdateAccountAfterSave 67
ValidateAccount 68
UpdateClientFromApplicant 69
UpdateClientAfterSave 71
UpdateApplicantFromClient 72
UpdateSecurityStmt 74
FinaliseAccount 75
GetAccountWarnings 76
BeforeStatusChange 78
AfterStatusChange 79
AcceptDeclineAcceptStatusSet 80
BeforeAccept 81
MonitorCategoryChange_ExecuteCommitPre 83
MonitorCategoryChange_ExecuteCommitPost 84
Summary Pages 85
Account, Client and Security Statement Summary Pages 85
Page Sets 86
Special Page Set Configuration 87
Loading and Validating an Existing Account Application and Initialising a New Record 88
Respecting the Page Set's Read-Only State 90
Showing Special Forms 91

Account Application Applicant 91
Account Financial 92
Collateral Item Wizard 93
HTML Widgets 94
Special HTML Widget Configuration 95
Appendix A – Guidelines 96
Script Responsibilities 96
Appendix B – Collateral Items vs Security Items 97

Disclaimer

This document contains information that may be subject to change at any stage.

All code examples are provided "as is".

Version History

Date	Version	Name	Changes
23/05/2014	1.00	PH	Created.
25/06/2014	1.01	PH	Updated to reflect changes and to reference Page Sets document.
10/07/2014	1.02	PH	Updated as per review by JR.
05/03/2015	1.03	PH	Updated to add details of User Settings and Task Manger integration plus Account Application template code details.
05/06/2015	1.04	PH	Account Application Type Data Schema changes.
12/08/2015	1.05	PH	SavePre, SavePost and SaveAudit events added.
25/11/2015	1.06	PH	Updated title etc for publication on Intersoft Website.
08/01/2016	1.07	PH	Additional buttons below Collateral grid.
20/01/2016	1.08	PH	Insurance details added for Collateral Items.
16/02/2016	3.00	PH	Updated for version 3 functionality and changes.
01/04/2016	3.01	PH	New Monitor Category relatedScript events.
22/04/2016	3.02	PH	New 'AfterSave' events for Clients and Accounts.
07/06/2016	3.03	PH	New 'BeforeAccept' event documented and First/ Last Names now updated on Client.
28/11/2016	3.04	JR	Updated "BeforeAccept" event documentation
31/01/2017	3.05	PH	HTML Widget support and Applicant Image File Name.
10/02/2021	**3.08**	**PH**	**SaveAudit event.**

Introduction

This document discusses finPOWER Connect Account Applications and is focused mainly on the creation and configuration of Account Application Types.

Account Applications require that finPOWER Connect is licensed for the Account Applications Add-On and, if providing a User Interface from within finPOWER Connect, the Page Sets Add-On.

If Page Sets are being used, this document should be used in conjunction with the **finPOWER Connect 2 Page Sets** document.

Workflows can also be used with Account Applications. These are limited to 'Version 2' type Workflows and are detailed in the **finPOWER Connect 2 Workflows (Version 2)** document.

Important

It is very important prior to creating a new Account Application Type that you understand the separation between Account Application Type and Page Set Scripts.

Recommended steps for creating and configuring new Account Applications and Page Sets is detailed Appendix A – Guidelines.

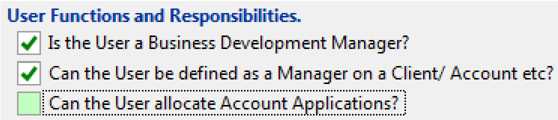
Account Applications Overview

- Account Applications in finPOWER Connect are separate from Accounts.
  - Only when an Account Application is completed is a saved Account record created.
  - This prevents Accounts being overrun by partial Applications and Applications that are declined.
  - The Account is linked back to the Account Application from which it was created.
- The Account Application Type controls most aspects of an Account Application, e.g.:
  - Data entry.
  - Scripting.
- Account Applications can have multiple Applicants.
  - These are analogous to Account Clients.
    - In fact, an Applicant can be pre-loaded from, and link, to a Client.
  - Clients will be created or, optionally, updated from an Applicant upon completing an Application.
    - Or at any point during the Application process if required.
- Account Applications can have multiple Collateral items.
  - These are analogous to Security Statement Items.
  - Only available if licensed for the Security Register Add-On.
  - **NOTE:** Appendix B – Collateral Items vs Security Items details the Scripting and property differences between Collateral Items and Security Items.
- Account Applications can have multiple Quotes (or Calculations).
  - These contain financial details in a simplified format, i.e., simplified compared to an Account's financial details.
    - The entire Account Calculation's financial details are however available.
  - A typical Account Application would only maintain a single Quote but supporting multiple Quotes allows greater functionality.
  - If an Account Application has multiple Quotes, one of them will always be flagged as the Active or Default Quote.
    - This contains the financial information that will be used when creating an Account.
- Account Applications have the following statuses:
  - Status:
    - Incomplete
      - Data entry is still in progress.
    - Reviewing
      - Data entry is complete.
      - Can switch back to 'Incomplete' at any point to update data.
        - Or, if so configured, edit the data without changing the Status back to 'Incomplete'.
    - Completed
      - Closed, i.e., Accepted
    - Declined
      - Closed by being Declined
    - Unwanted
      - Closed by being Unwanted
  - Sub-status:

        - A scriptable sub-status is also available.
- Account Applications rely heavily on Scripting.
    - The Account Application Type defines the Script code.

User & Responsibilities

- Users & Responsibilities form, Responsibilities page:

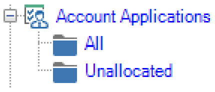


  - Can the User allocate Account Applications?
    - This is used in the Task Manager to decide whether to show the User the "Unallocated" Account Applications system folder and the ability to allocate to a User.
    - Allocation relates to setting the Account Application's Account Manager.

NOTE: The New User wizard also allows responsibilities to be configured.

User Preferences

- User Preferences form, Account Applications section.
  - General page:
    - Show the External Id field?
      - Used by the Account Applications form and the Account Application Summary page.
- User Preferences form, Tasks & Workflows section.
  - General page:
    - Show System Folders for Account Applications?
      - This is used in the Task Manager, i.e.:



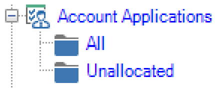
  - Scripts page:
    - Summary Page, AccountApps.
      - Summarises an Account Application in the Task Manager.
      - Can be defined for all Users under Global Settings.

Global Settings

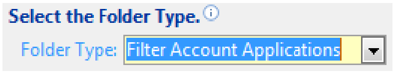
- Global Settings form, Account Applications section.
    - General page:
        - An auto-sequence code can be defined.
            - This can be overridden on each Account Application Type.
    - Defaults page:
        - Allows defaults to be defined (which can be overridden on each Account Application Type), e.g.:
            - The Contact Method to use for the Phone number when creating a new Client from an Applicant.
    - Scripts page:
        - Scripts to summarise objects used by Account Applications.
- Tasks & Workflows section.
    - Scripts page:
        - Summary Page, AccountApps.
            - Summarises an Account Application in the Task Manager.
            - Can be overridden on User Preferences.

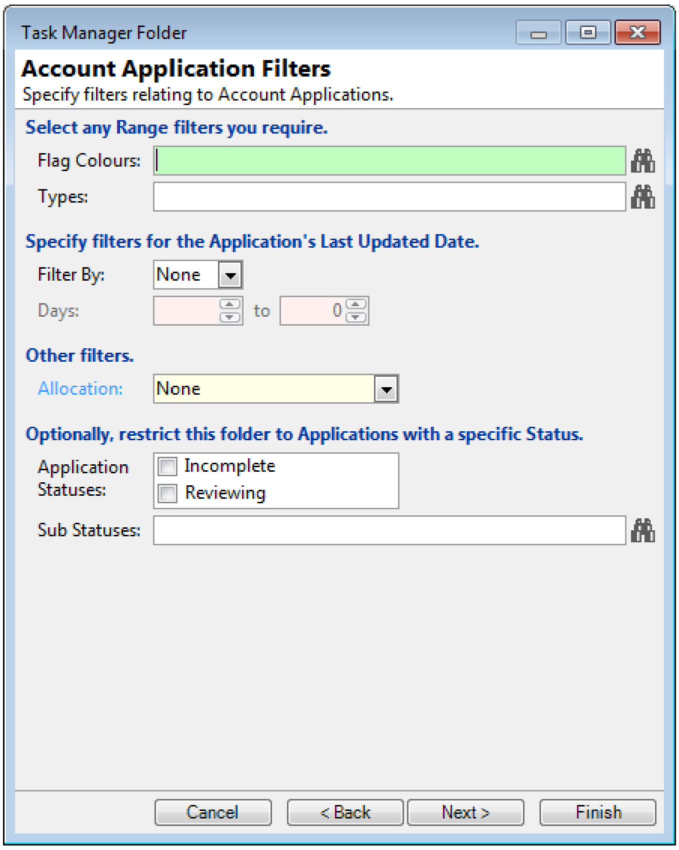
Task Manager

- System folders are added for Account Applications:

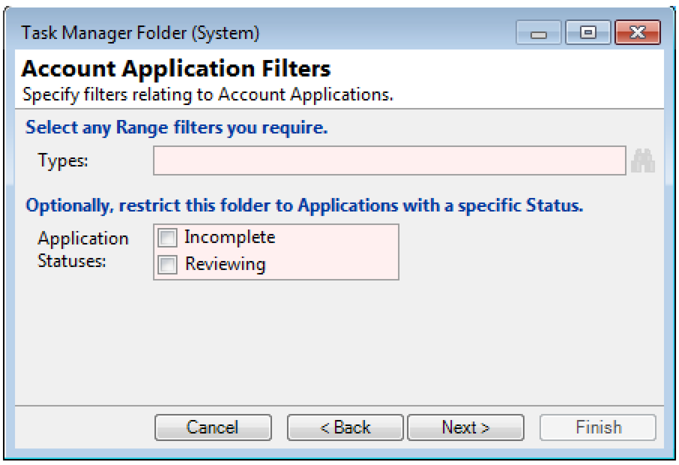


  - Providing not disabled under User Preferences.
  - An "Unallocated" folder is added if the User is allowed to allocate Account Applications.
- Folders can be "Filter Account Applications" type folders:





- Folders that are "Filter Workflows" now have the ability to apply Account Application filters as well as Account filters.



Concepts

No Page Set or other Custom User Interface

- It is possible (although unlikely) to use Account Applications without any customised method of data-entry, i.e., no Page Set from which Users can enter information and no other custom User Interface such as Web Pages.
- An Account Application can be entered directly via the Account Applications form.
  - Applicants can be added.
  - Collateral items can be added.
  - **NOTE:** Certain Application properties may not be available to enter via this form.
- The Application can be Accepted or Declined and an Account generated.
  - This is possible even if the Account Application Type does not contain any Script code.

Page Sets

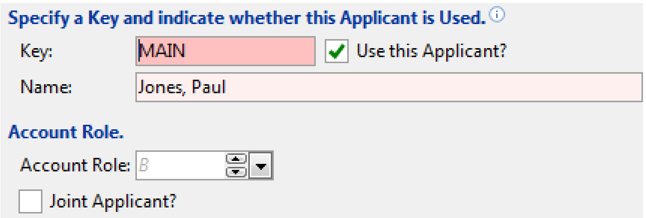
- Page Sets can be used to provide a custom User Interface from which to enter Account Applications from within the finPOWER Connect Windows Interface.

Web Pages

- Web pages can be written to provide a Web-based User Interface for entering Account Applications.
  - Web Pages would typically use the finPOWER Connect Web Services to support this functionality.

Key Locking

- Applicants, Collateral Items and Calculations (Quotes) can specify a key to identify them in their collection.
- These items can be 'Key Locked' meaning that the key cannot be changed from the User Interface:



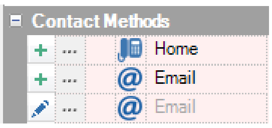
- This would typically be done by the Account Application Type Script.
  - Generally in the Initialise event, e.g.:
    - Create an Applicant with a key of 'MAIN' and set it to Key Locked.
    - Create a Collateral Item with a key of 'MV' and set it to Key Locked.
- Key Locked records cannot be deleted by Users via the Account Applications form and their Key cannot be changed.
  - Setting on the Account Application Type can also prevent Applicant and Collateral Item details being updated by a User if the items are Key Locked.
- Why Key Lock records?
  - Ensures that a data-entry mechanism such as a Page Set is guaranteed that certain sub-items of the Application (e.g., an Applicant with a key of 'MAIN') exist.
  - Makes coding of the Page Set Script (or other custom User Interface) much simpler.

Temporary Account Object

- At any point, an Account Application can be asked to produce a temporary (or provisional) Account object.
- This is fundamental to Account Applications and allows the following:
  - The full finPOWER Connect Account Calculation engine to be used within Account Applications.
    - This includes any Account Calculation Scripts.
    - **WARNING:** Any Account Types defining a Calculation Script should be reviewed before being used for Account Application to ensure they function as expected.
  - Account-based summary pages to be used.
  - The Accounts form can be used to display the 'provisional' Account.

Temporary Client Object

- At any point, an Account Application can be asked to produce a temporary (or provisional) Client object.
- This is fundamental to Account Applications and allows the following:
  - The Clients form can be used to display the 'provisional' Client.
    - This may be a brand new Client record or an existing Client updated from the Applicant details.
      - For existing, updated Clients, grids such as Contact Methods will show an icon in their row selector to show new and updated records, e.g.:



Account Application Types Form

A finPOWER Connect database can contain any number of Account Application Types.

The various pages on the Account Application Types form are described in this section.

General

General details.

- **Code and Description**
  - Code:
    - A unique code up to 10 characters long.
  - Active:
    - New Account Applications can only be added for 'Active' Account Application Types.
  - Description:
    - A description of the Account Application, up to 50 characters long.
- **Version**
  - Version:
    - You can record a version number against an Account Application.
      - Any Account Application created for this Account Application Type will also record the version number at that point in time.
      - Scripts and other functionality can use this version number, e.g., to display different information in a Summary Page for different versions.
      - **NOTE:** If you wish to make major changes to an Account Application Type for which Account Applications have been created, it may be advisable to create an entirely new Account Application Type rather than using the version number.
- **Icon**
  - Icon:
    - An icon to represent this Account Application Type.
    - This may be used in various places, e.g., as a form icon when entering data for the Account Application.
- **Allow User to add new Applications**
  - Show in Account Menu:
    - Indicates whether to display an item in the **Account** menu for adding a new Account Application of this type.
    - Caption:
      - The caption to display in the menu.
  - Show from 'Add New Record' button on Account Applications form:
    - Indicates whether to display an entry in the list that is displayed upon clicking the 'Add New Record' button on the Account Applications form.
  - **NOTE:** If neither of these options are checked, Account Applications of this type cannot be created by Users within finPOWER Connect. However, they can still be created via a Script or an external application.

Options

Miscellaneous options for the Account Application Type.

- **Options for new Account Applications**
  - Next Code:
    - The next code to be allocated to new Account Applications of this type.
    - **NOTE:** This can be used to override the setting on the Global Settings, Account Applications, General page.
- **Script Language and Timeout**
  - Language:
    - The Script Language.
  - Timeout (secs):
    - The Script Timeout in seconds or blank (zero) for no timeout.
- **Other Scripts**
  - Summary Page:
    - Each Account Application Type can define a Summary Page.
    - The built-in Account Application Key Details page uses this and, if undefined, falls back to use the built-in Account Application summary.
- **Defaults for the Accept/ Decline Account Application wizard**
  - Create Clients when not Accepting the Application?
    - By default, when a User chooses 'Decline' or 'Unwanted', new Client records will not be added for the Applicants.
  - Update existing Clients when not Accepting the Application?
    - By default, when a User chooses 'Decline' or 'Unwanted', existing Client records will not be updated from the Applicant details.
  - **NOTE:** These settings can be tweaked on-the-fly using the 'AcceptDeclineAcceptStatusSet' Script event.
- **Decision Cards to run during the Application**
  - NOTE: Not yet implemented
- **Workflows to begin during the Application**
  - Created:
    - The 'Account Application' Workflow Type to start when this Account Application is created and has first been saved.
    - **NOTE:** The Workflow will be added within the same database transaction in which the Account Application is saved.
  - Reviewing:
    - The 'Account Application' Workflow Type to start when this Account Application is status is changed to 'Reviewing'.
    - **NOTE:** The Workflow will be added within the same database transaction in which the Account Application's status has been updated.
  - Accepted:
    - The 'Account' Workflow Type to start on the new Account that is created when this Account Application is accepted.
    - **NOTE:** The Workflow will be added within the same database transaction in which the Account has been created and saved.

Quotes

An Account Application can have one of more Quotes (financial calculations).

- **Options**
  - Allow additional Quotes to be added to the Application:
    - If checked, any number of additional Quotes can be added to an Account Application in addition to the Quote (or Quotes) defined in the User Interface (e.g., the Page Set) from which the Application details are captured.
  - Always auto-calculate when editing Financial details:
    - An Account Application's financial details can be edited from a special form which displays the same information as the Financial page of the New Account Wizard.
      - Checking this option means that financial details on this form will be auto-calculated (i.e., refreshed without having to click the 'Calculate' button).
        - This overrides the setting defined under User Preferences, Accounts, General.
- Default Account Type:
  - The Account Type to use for the default Quote or any new Quotes.

Applicants

An Account Application can have one of more Applicants. These are similar to Clients on an Account.

When an Account Application is completed (or, on demand), the Applicants will be used to create new finPOWER Connect Clients or to update existing Clients.

- **Options**
  - Allow additional Applicants to be added to the Application:
    - If checked, any number of additional Applicants can be added to an Account Application in addition to the Applicant (or Applicants) defined in the User Interface (e.g., the Page Set) from which the Application details are captured.
      - This allows a simple User Interface to be configured, e.g., to capture a single Applicant's details but additional Applicants to be added outside of the data capture User Interface (e.g., Page Set).
  - Allow Script maintained Applicants to be edited independently:
    - When an Account Application is first created, its Account Application Type Script will create one or more Applicants. It is these Applicants that the data capture User Interface (e.g., Page Set) will be designed to allow entry of.
    - Checking this option means that the details of these Applicants can be updated outside of the data capture User Interface, i.e., from a special wizard on the Applicants page of the Account Applications form.
      - This allows the User Interface to record simple details such as a physical address and phone number but for additional details such as a postal address and email address to be entered outside of the data capture User Interface.
- **Defaults for Account Clients and new Clients created from Applicants**
  - Account Role:
    - The Account Role to assign to Account Clients.
      - This is just a default and can be updated from a Script or from the Applicants page of the Account Applications form.
  - Client Type:
    - The Client Type to assign to new Clients.
      - This is just a default and can be updated from a Script or from the Applicants page of the Account Applications form.
  - Client Group:
    - The Client Group to assign to new Clients.
      - This is just a default and can be updated from a Script or from the Applicants page of the Account Applications form.
- **Options for updating existing Clients**
  - Contacts:
    - Specifies how contact records (e.g., address or phone number) will be updated when updating an existing Client.
      - None
        - The built-in Account Application functionality will not attempt to update existing Contact Methods for the Client.
      - Create a New Record
        - The built-in Account Application functionality will always create a new Contact Method record and make the existing Contact Method not current (i.e., set the 'Valid To' date and flag as 'Historic').
      - Update Existing Record

            - The built-in Account Application functionality will update the existing Contact Method record.
    - Identification:
        - Specifies how identification records (e.g., drivers licence) will be updated when updating an existing Client.
            - None
                - The built-in Account Application functionality will not attempt to update existing Identification items for the Client.
            - Create a New Record
                - The built-in Account Application functionality will always create a new Identification item record and make the existing Identification item not current (i.e., flag as 'Historic').
            - Update Existing Record
                - The built-in Account Application functionality will update the existing Identification item record.
- **Contact Methods for new Client Contacts**
    - **NOTE:** These can be defined under Global Settings, Account Applications, Defaults.

Collateral

An Account Application can have one of more Collateral Items. These are similar to Security Statement Items.

When an Account Application is completed, the Collateral Items will be used to create a new Security Statement.

NOTE: This page is only visible if licensed for the Security Register Add-On.

- **Options**
    - Allow additional Collateral to be added to the Application:
        - If checked, any number of additional Collateral Items can be added to an Account Application in addition to the Collateral Item (or Collateral Items) defined in the User Interface (e.g., the Page Set) from which the Collateral details are captured.
            - This allows a simple User Interface to be configured, e.g., to capture a single Collateral Item's details (or maybe no details at all) but allow additional Collateral to be added outside of the data capture User Interface (e.g., Page Set).

Data Schema

Although Account Applications, Applicants and Collateral Items have a large number of built-in properties, any additional data is stored in User Data on the respective object.

The Data Schema page shows a grid of available properties (fields) for the various objects (selectable from the Object dropdown). Built-in properties are shown in blue.

NOTE: Although you need not define any additional data you will be recording, doing so has the advantage of validating User Data access by Scripts, Page Sets etc.

- **Objects and Custom Properties**
  - Object:
    - Selecting the object (Application, Applicant or Collateral) will display the respective properties in the grid.
    - Clicking the 'Business Layer Help' button will display the business layer help for the selected object, e.g., `finAccountApp` for Application.
  - Validate Data Schema access:
    - When using any form of Script to access an object's User Data, a trace warning will be given if the item being access is not defined in the table of fields.
  - Grid:
    - Additional properties (i.e., valid User Data entries) can be entered in the grid.
    - Clicking the drilldown button will show the business layer help for the selected system property or for the selected object's `UserData` property for User Defined properties.

Data Capture

From within finPOWER Connect, an Account Application's data can be entered via one or more Page Sets which are defined on this page.

It is possible to enter an Application from the Account Applications form alone or, from an application external to finPOWER Connect, e.g., a Website. In these cases, it is feasible that no Page Sets are defined on this page.

NOTE: Any Page Set related information on this page is only visible if licensed for the Page Sets Add-On.

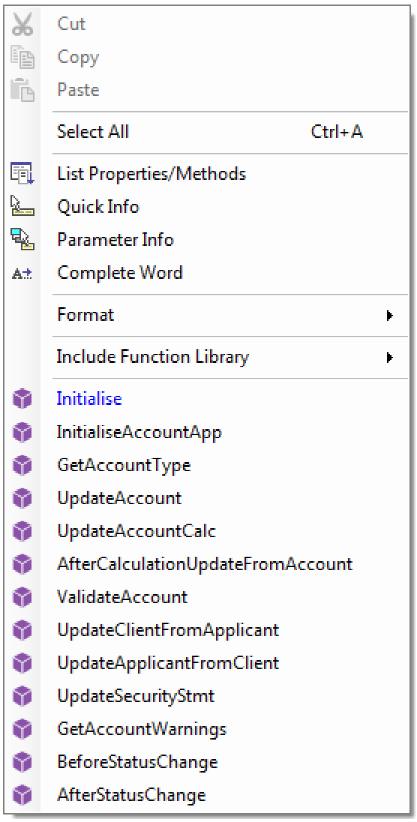
- **Page Sets used for data capture**
    - General:
        - The Page Set to use to add a new or edit an existing Account Application.
    - Applicant:
        - A Page Set into which Applicant details can be entered if not captured from the 'General' Page Set.
            - **NOTE:** This is not yet implemented.
    - Collateral:
        - A Page Set into which Collateral Item details can be entered if not captured from the 'General' Page Set.
        - **NOTE:** This is not yet implemented.
- **View Method for entered details**
    - Method:
        - The Account Applications form has a 'Data Capture' page which displays the information entered.
            - If data capture has been achieved via a Page Set then the Page Set will be used to display a read-only version of the Application data.
                - Each page is represented by a different tab.
            - If data is not captured via a Page Set or if you want to totally customise how the data is presented then a Summary Page Script can be defined.
                - A built-in Summary page will show a basic summary of the Account Application details if none is specified.
- **Editing of data while Application is 'Reviewing'**
    - Allow editing of captured data when the Application is 'Reviewing':
        - By default, once an Application's status has changed from 'Incomplete' to 'Reviewing', data entry (via the data capture Page Set) is no longer allowed. This flag allows this behaviour to be overridden.
            - **NOTE:** An Application's status can generally be changed from 'Reviewing' to 'Incomplete' to allow the data to be edited therefore, unless you specifically want to be able to update the captured data when 'Reviewing', this option can be left unchecked.
    - Page Set:
        - You can override the Page Set used for editing of a 'Reviewing' Account Application.
            - If unspecified, the 'General' Page Set will be used but it will be put into a special 'Tabbed Pages' mode so that it no longer appears as a wizard, thereby allowing faster access to the data.

Script Code

Account Applications use Script code for almost everything. Although it is possible to have an Account Application Type with no Script code, this is unlikely.

Unlike many other Scriptable objects (Scripts, Documents, Queues etc.), an Account Application Type's Script does not have a 'Main' method. Instead, it has event methods that are called at various points in the Application's lifetime or when performing certain tasks.

Right-clicking in the Script editor will list all available events. These will be blue if they already exist in the Script or black if they do not.



Clicking a blue even will go to that block of code; clicking a black event will insert it into the Script, e.g.:

```
Public Overrides Function InitialiseAccountApp(accountApp As finAccountApp) As Boolean

   ' Assume Success
   InitialiseAccountApp = True

   ' Set Defaults or return False to fail initialisation

End Function
```

NOTE: The 'Paste template Script code' button inserts all available events and is therefore less likely to be used than other types of Script.

Constants

Like any other admin library that uses Scripts, Constants can be defined on this page and then used by the Script.

NOTE: Unlike most types of Scripts where Constants are assign in the 'Main' method of the Script, Account Applications have a special 'Initialise' event to do this.

Documents

Lists both Account an Account Application Documents than can be used for Applications of this Type.

Although a Product Type can define whether an Account Document can be used for Account Applications and Account Applications can be configured to use Account Documents, realistically this is limited to where Account Application support has been built in to the Document. The following Documents supplied with finPOWER support Account Applications:

- Account_Quotation.dot
- Account_Quotation_Multi.dot

NOTE: Unlike most VBA documents, the Quotation documents are passed an in-memory `finAccount` object (rather than being published from an Account Log).

Since Account Applications have the ability to create an unsaved `finAccount` object, very little extra code is required to support Account Applications in this type of document.

Permission

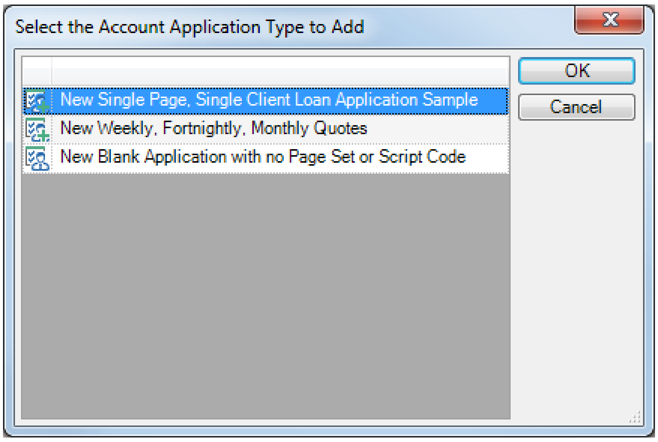
- **Administrator Permission**
  - Run with Administrator permission:
    - If checked, the Script will run with Administrator permission meaning that any permission key checks will always be passed.
- **Optional Permission Key to restrict using this Account Application**
  - Permission Key:
    - Allows a permission key to be defined which, if denied for a User, prevents the User from creating or editing Account Applications of this type.

Account Applications Form

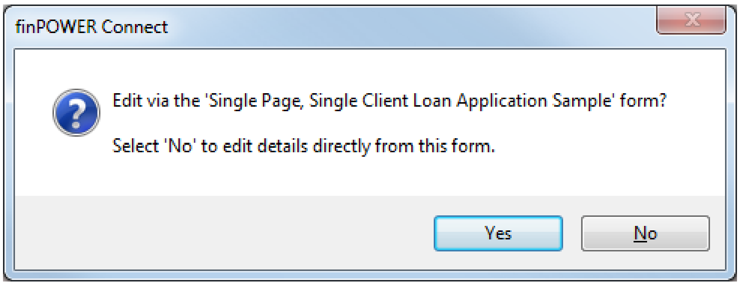
The various pages on the Account Applications form are described in this section.

The Add and Edit buttons for Account Applications work differently from other forms:

- Add
    - Instead of putting the Account Applications form into edit mode or opening a wizard (as per the Accounts and Clients form), this form does the following:
        - Prompts the User to select the Account Application Type to add, e.g.:



        - Upon selecting an item and clicking OK, either of the following will occur:
            - If the Account Application Type defines a Page Set, this will be run.
            - Otherwise, a new Account Application record will be initialised and the Account Applications form put into Edit mode.
- Edit
    - The Edit button works differently depending on the Status of the Account Application:
        - Incomplete
            - The User will be prompted to edit the Account Application via a Page Set (providing the Account Application Type defines one) or directly from this form, e.g.:



                - If the Page Set is chosen, this will be run and will allow editing of the Account Application's data.
                - Editing an Application via the Account Applications form rather than a Page Set allows the following:
                    - Properties such as Flag Colour and Review Date to be updated.
                    - Applicants and Collateral Items can be edited or added independently of the Page Set.
                    - Advanced financial details can be edited for the Account Application's Quotes or additional Quotes added, even if the Page Set does not support this.
        - Reviewing

- As per the 'Incomplete' Status, the User will be prompted to edit the Account Application via a Page Set if applicable.
    - If the Page Set is chosen then the 'Reviewing' Page Set defined on the Account Application Type will be used for editing or, if undefined, the 'General' Page Set will be used but will be forced to a 'Tabbed Pages' navigation method.

¤ Other

- The form will be put into Edit mode but most fields will be read-only and Quotes, Applicants and Collateral cannot be added, edited or deleted.

General

General details.

- **Code and Name**
  - Code:
    - A unique code up to 10 characters long.
    - This is auto-sequenced based on the setting defined under Global Settings, Account Applications, General page or overidden on the Account Application Types form, Options page.
  - Name:
    - The Application's name, up to 100 characters long.
      - Be default, this is auto-generated from the Applicants' names.
        - If these are blank or not available (an Account Application can be saved with minimal details entered), an auto-generated description will be used for the name. This will then be overwritten when Applicant details are entered.
  - External Id:
    - This Application's identifier in an external system.
    - **NOTE:** This will only be displayed if the User has opted to display it under the User Preference, Account Applications, General page.
- **Application Type and Status**
  - Type:
    - The Account Application Type.
  - Status:
    - The Account Application's Status.
      - One of:
        - Incomplete
          - Data entry is still in progress.
        - Reviewing
          - Data entry is complete.
          - Can switch back to 'Incomplete' at any point to update data via the 'Reset to Incomplete' form action.
          - Or, if so configured, edit the data without changing the Status back to 'Incomplete'.
        - Completed
          - Closed, i.e., Accepted
        - Declined
          - Closed by being Declined
        - Unwanted
          - Closed by being Unwanted
  - Sub-status:
    - A scriptable sub-status is also available and will be displayed after the Status if it has a value. If not, it will be hidden.
  - Flag Colour:
    - A flag colour for this Application.
  - Flag Notes:
    - Flag notes for this Application.
- **Application Dates and completion details**

- Quoted:
  - The Quoted date. By default this is the date on which the Application was created.
- Closed:
  - The date on which the application was closed, i.e., Completed, Declined or Unwanted.
- Account:
  - The Account that was created as a result of completing this Account Application.

- **Account Manager**
  - Manager:
    - The Account Manager.
      - This defaults from the setting defined under User Preferences, Clients, General.
  - Review Date:
    - Account Applications on or after a set Review Date will appear in the Task Manager for their Account Manager.
- **Notes**
  - Any notes regarding this Application.

Quotes

An Account Application can store any number of Quotes.

A quote contains the financial details used in an Account Calculation.

If multiple Quotes are defined, this page will contain a tab for each Quote.

- **A preview of the Account that this Application will produce**
  - Top Summary:
    - If this Application is complete, this will be a summary of the actual Account that was created.
    - If the Application is not complete, this will be a summary of the Account that will be created based upon this Account Applications details and the selected Quote.
      - A hyperlink will be displayed in the Summary section (providing the built-in Script is used). Clicking this will open the Accounts form to allow the Account that will be created to be viewed.
        - This is possible since every Account Application Type Script must provide functionality to create an Account, Clients and a Security Statement from the information entered on the Account Application.
  - Bottom Summary:
    - This is a summary of the Account financial details for the selected Quote.
    - If the Application is complete, there will always be at least 2 tabs, one for the Actual Account that was created and one for each of the original Quotes.
      - The 'Starred' Quote is the one that will be used when creating an Account upon completing the Application.
      - The 'Edit Quote Properties' button allows notes to be recorded against a Quote and also allows the Quote to be flagged as invalid, i.e., so the Quote details can be retained but to indicate that this Quote should not be used for some reason.
      - An 'Edit Quote Financial details' button allows the financial details to be edited via a special wizard which has the same functionality as the Financial page on the New Account wizard.
        - **NOTE:** Once edited via this wizard, the details entered in the wizard are stored with the Quote and are used in preference to any of the more basic properties such as PreferredLoanAmount stored on the Account Application. From the form launched from the 'Edit Quote Properties' button, you can use the 'Clear' button to reset this information.

This table describes each of the buttons displayed at the bottom of this page.

	Add New Quote	Add a new Quote. The Account Application Type must allow multiple Quotes (Quotes page, 'Allow additional Quotes to be added to the Application').
	Delete Quote	Delete this Quote. Cannot delete if only one Quote exists or if the Quote is 'Key Locked' by a Script.
	Make Selected	Make this the 'Selected' Quote, i.e., the one that will be used when the final Account is created.
	Duplicate	Duplicate this Quote.

	Move Left	Move this Quote left in the list.
	Move Right	Move this Quote right in the list.
	Properties	Edit Quote properties as described above.
	Edit Financial	Edit Quote's financial details as described above.
	View Schedule	View Account Schedule for this Quote.
	Print Quotation	Print Account Quotation. Using the document specified on this Quote's Account Type.

Applicants

An Account Application can store any number of Applicants, regardless of whether all of these can be entered from the Account Application data capture User Interface (e.g., a Page Set).

- **Applicants**
  - Top Grid:
    - Displays a grid of all Applicants.
    - A drilldown button (ellipsis) allows the following:
      - Details such as Account Role, Client Type and Client Group can be set.
      - Links to an existing Client can be maintained.
        - The existing Client can be updated at any time, or, a new Client created without having to wait until the Application is completed.
      - The Applicant can be flagged as not used, i.e., their details will still be retained but they will not become Clients or be linked to the Account that will be created upon completing the Application.
    - A drilldown button (applicant icon) allows the following:
      - Applicant details to be updated via a built-in wizard (or a Page Set defined on the Account Application Type) which is like a cut-down version of the New Client wizard.
  - Bottom Summary:
    - A summary of the Applicant selected in the above grid. This includes a summary of the Client that will be (or has been) either created from or updated from the Applicant record.
      - A hyperlink will be displayed in the Applicant Summary section (providing the built-in Script is used). Clicking this will open the Clients form to allow the Client that will be created or updated to be viewed.
        - This is possible since every Account Application Type Script must provide functionality to create an Account, Clients and Security Statement from the information entered on the Account Application.

This table describes each of the buttons below the Applicants grid.

	Add New Applicant	Add a new Applicant record. The Account Application Type must allow additional Applications to be added (Applicants page, 'Allow additional Applicants to be added to the Application').
	Delete Applicant	Delete the selected Applicant. Cannot delete if the Applicant is 'Key Locked' by a Script.
	Move Up	Move the current Applicant up in the list.
	Move Down	Move the current Applicant down in the list.
	Add Log	Add a Log for the current Applicant.
	Credit Enquiry	Perform a Credit Enquiry for this Applicant.
	Bank Account Enquiry	Perform a Bank Account Enquiry for this Applicant. **NOTE:** Bank Account Enquiries can currently only be performed for Applicants linked to an existing Client record.

	PPSR G2B Search	Perform a PPSR G2B Search for this Applicant.

Collateral

An Account Application can store any number of Collateral Items. These will be used to create a Security Statement when the Account Application is completed

NOTE: This page will only be displayed if licensed for the Security Register Add-On.

- **Collateral**
  - Top Grid:
    - Displays a grid of all Collateral Items.
    - A drilldown button (ellipsis) allows the following:
      - Details such as the Security Type can be set.
      - The Collateral Item can be flagged as not used, i.e., the details will still be retained but they will not become items on the Security Statement that will be created upon completing the Application.
    - A drilldown button (security icon) allows the following:
      - Collateral details to be updated via a built-in wizard which is like a cut-down version of the Security Item form.
  - Bottom Summary:
    - A summary of the Collateral Item selected in the above grid. This includes a summary of the Security Statement Item that will be created from the Collateral Item record.
      - A hyperlink will be displayed in the Applicant Summary section (providing the built-in Script is used). Clicking this will open the Clients form to allow the Client that will be created or updated to be viewed.
        - This is possible since every Account Application Type Script must provide functionality to create an Account, Clients and a Security Statement (if required and licensed) from the information entered on the Account Application.

This table describes each of the most common buttons below the Collateral grid.

	Add New Collateral	Add a new Collateral item. The Account Application Type must allow additional Applications to be added (Collateral page, 'Allow additional Collatereal to be added to the Application').
	Delete Collateral	Delete the selected Applicant. Cannot delete if the Collateral item is 'Key Locked' by a Script.
	Duplicate	Duplicate the current Collateral item.
	Move Up	Move the current Collateral item up in the list.
	Move Down	Move the current Collateral item down in the list.
	Add Log	Add a Log for the current Collateral Item.
	CentrixNZ	Perform a CentrixNZ Security Enquiry.
	MotorWeb	Perform a MotorWeb Security Enquiry.

Data Capture

This page displays the details entered on the Account Application in a read-only format.

The Account Application Type can define a Page Set by which the Application's data is captured. By default, this page will show a series of tabs, each one representing a page in this Page Set.

The Account Application Type can be configured to use a Summary Page Script to display this information instead (or if a Page Set is not being used) if required.

Workflows

This page displays all Workflows created for this Account Application.

The page functions the same as the Workflows page on the Accounts and Clients forms.

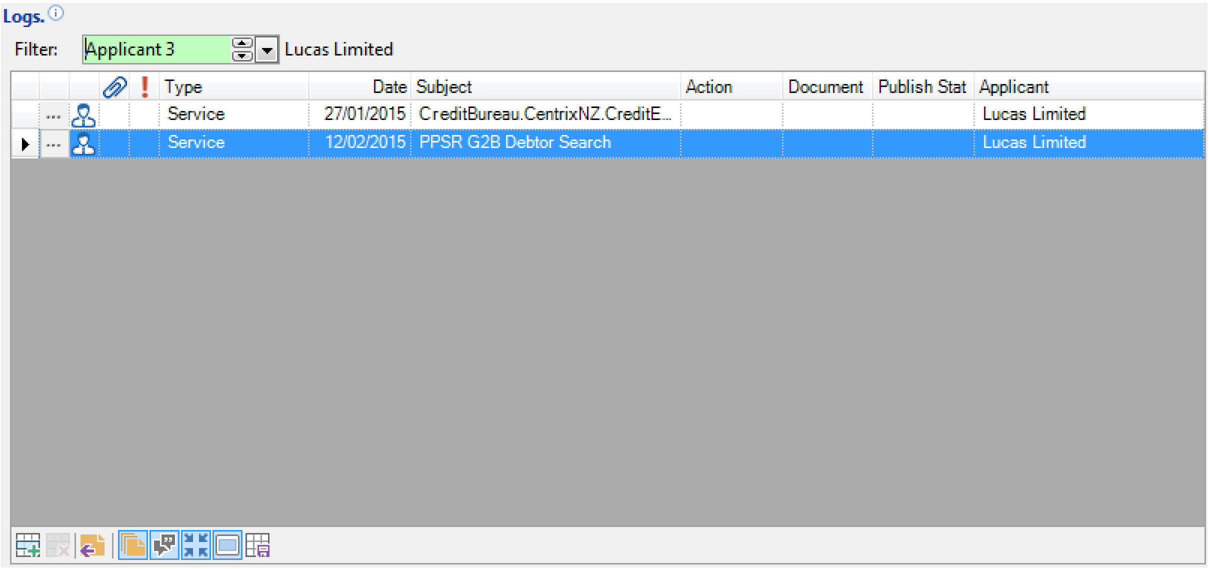
Logs

This page displays all Logs recorded against the Account Application.

Account Application Logs can optionally be linked to a specific Applicant or Collateral Item.

Client Logs dated from the date at which the Account Application was created up until the date at which it was closed are also includes.

The page functions the same as the Logs page on the Accounts and Clients forms but allows various filters to be applied, e.g., to show only Logs related to a particular Applicant.



When performing an action such as a Credit Enquiry against an Applicant that is linked to an existing Client, two Logs are created; one for the Client and one for the Applicant.

When viewing the Logs grid, this may appear confusing hence a 'Show/ Hide duplicate Logs linked to the same Service Log' button is included. This is unchecked by default which means that any Client Logs relating to actions such as Credit Enquires will be hidden if a corresponding Account Application Log exists.

Files

This page displays all Files linked to the Account Application either via Logs or because they existing in the Account Application's Document Manager folder.

Audit

This page displays audit information for this Account Application.

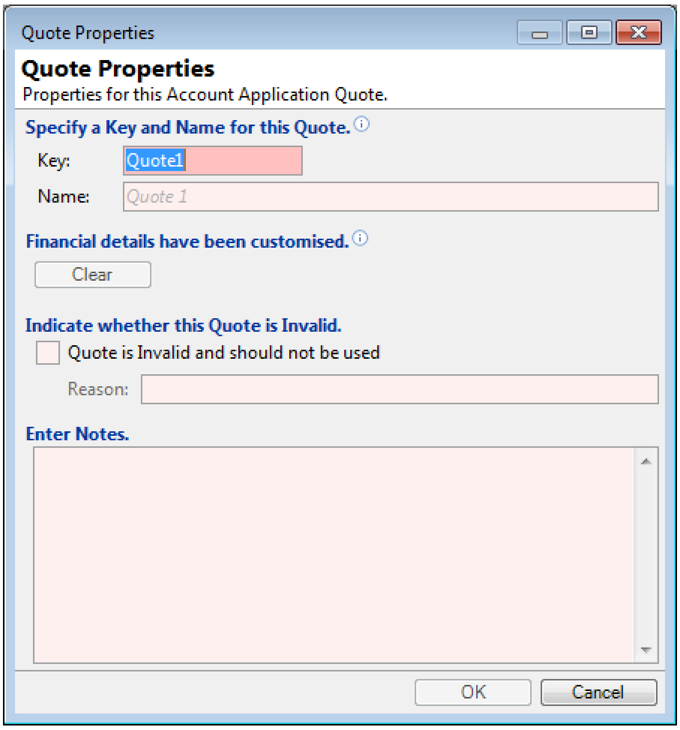
By default, this page is only visible to finPOWER Connect Administrators.

Account Application Sub-Forms

This section describes the sub-forms that are accessible from the Account Applications form to view or edit sub-object details.

Quote Properties

This form allows basic Quote properties to be edited and is displayed upon clicking the 'Edit Quote Properties' button at the bottom of the Quotes page.



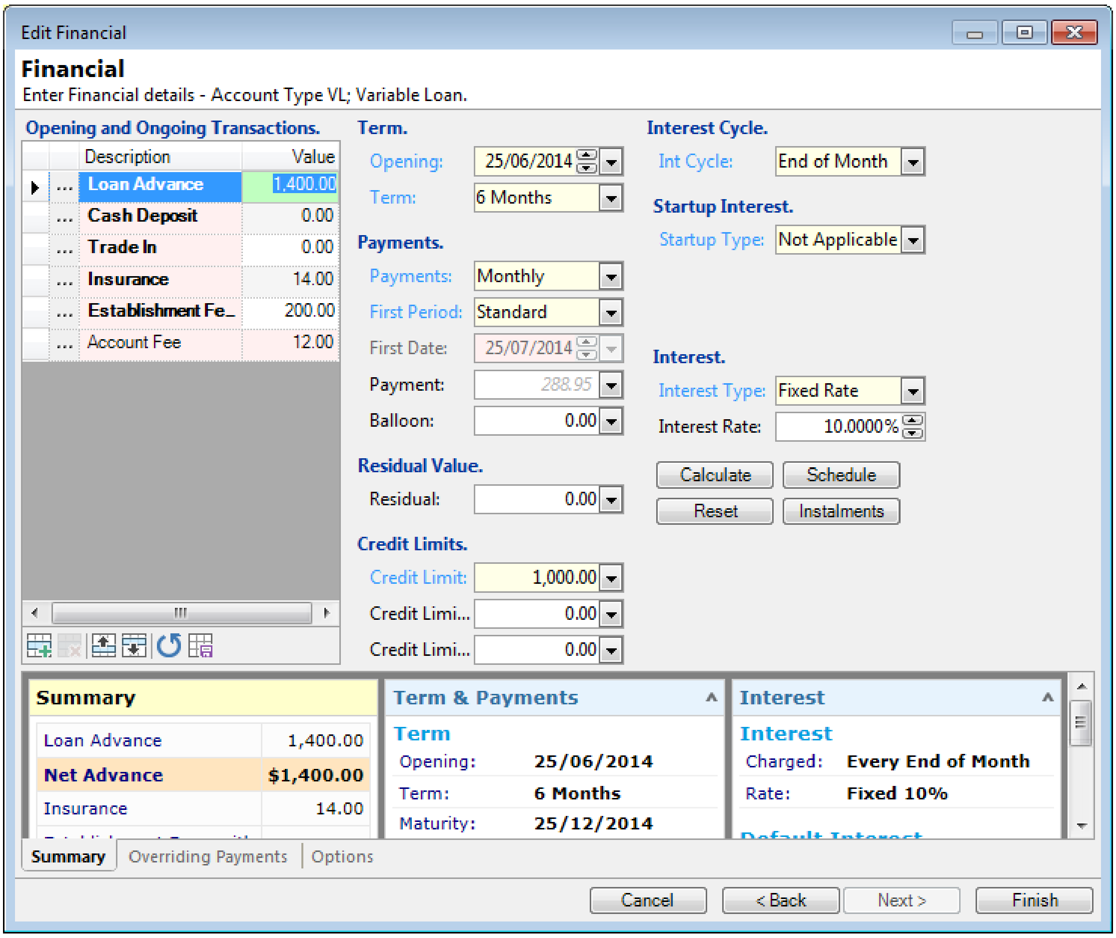
- **Specify a Key and Name for this Quote**
  - Key:
    - A unique code up to 20 characters long.
    - This will default when adding a new Quote, e.g., 'Quote1', 'Quote2' etc.
    - **NOTE:** This will be read-only if this Quote is 'Key Locked', i.e., a Script (Account Application Type of Page Set) has locked it since it relies on a Quote with this Key existing for the Account Application.
  - Name:
    - A friendly Name for this Quote. This will be displayed in the Quote's tab if the Account Application has multiple Quotes.
- **Financial details have been customised**
  - The 'Clear' button will be enabled if this Quote's financial details have been customised via the 'Edit Financial Wizard' or via some mechanism on the Page Set through which the Quote was entered.
    - This clears out the financial details and forces the Account Application to use default financial details.
      - By default, these will be properties such as PreferredAccountTypeId, PreferredTerm etc specified on the Account Application.
        - This functionality will commonly be overridden in the Page Set Script's UpdateAccountCalc event.
- **Indicate whether this Quote is Invalid**
  - Quote is Invalid and should not be used:

        - A Quote can be flagged as invalid and a reason given, e.g., where the repayments are deemed to be too high.
            - This will prevent an Account based upon this Quote from being used when the Account Application is accepted.
            - Invalid Quotes will display red text on the Quote's tab if the Account Application has multiple Quotes and, if using the build-in summary page, the Account preview shown on the Quotes page of the Account Applications form will display an alert message at the top of the 'Summary' block.
        - Reason:
            - A reason describing why this Quote is invalid up to 100 characters.
- **Enter Notes**
    - Any notes you wish to record against this Quote.

Edit Financial Wizard

This form allows the full financial details of a Quote to be edited and is displayed upon clicking the 'Edit Quote Financial details button at the bottom of the Quotes page.

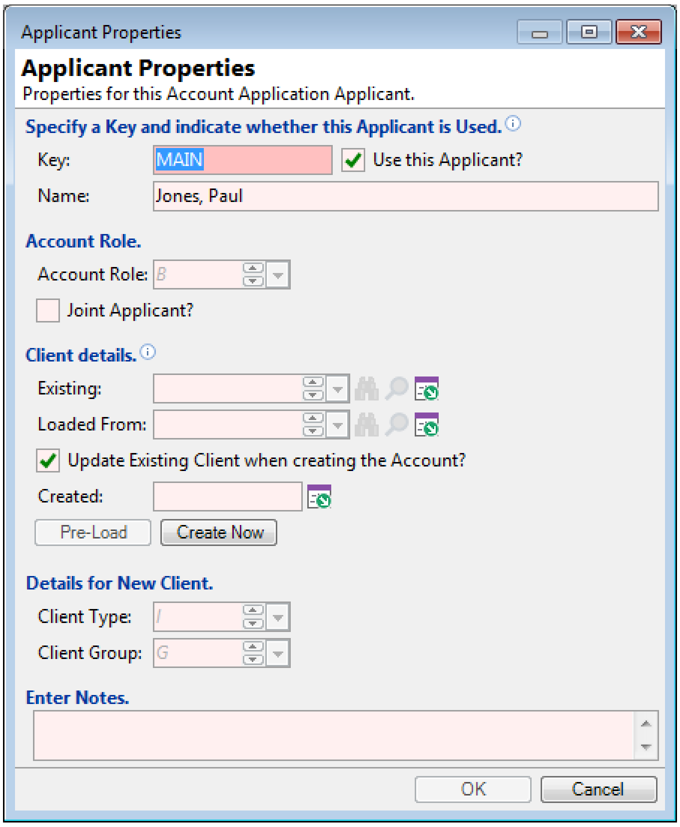
NOTE: This wizard allows the same information to be edited as the New Account and Account Quotation wizards and so is not described in detail in this document.



Although any financial details can be updated from this wizard, the Account Application Type Script's ValidateAccount or GetAccountWarnings events can be used to validate the details and prevent the Account Application from being accepted if required.

Applicant Properties

This form allows basic Applicant properties to be edited and is displayed upon clicking the ellipsis drilldown button in the Applicants grid.



- **Specify a Key and indicate whether this Applicant is Used**
  - Key:
    - A unique code up to 20 characters long.
    - This will only generally have a value if the Applicant has been created from a Script.
    - **NOTE:** This will be read-only if this Applicant is 'Key Locked', i.e., a Script (Account Application Type of Page Set) has locked it since it relies on an Applicant with this Key existing for the Account Application.
  - Use this Applicant:
    - Indicates whether this Applicant is used, i.e., it should create a Client (or update an existing Client) when creating the Account when the Application is accepted.
    - Typically, if an Account Application Type is configured to allow entry (via a Page Set for example) of multiple Applicants, e.g., a 'Main' and a 'Joint', the Account Application Type Script might create multiple Applicant records but flag one of them, e.g., the Joint as not being used.
      - The Page Set could then have a checkbox indicating whether to use the 'Joint' Applicant and then update the 'Used' flag on the Applicant record accordingly.
  - Name:
    - A read-only, auto-generated name for this Applicant.
- **Account Role**
  - Account Role:
    - The Account Role to use when generating an Account Client from this Applicant.
      - If unspecified, this will be the Account Role defined on the Account Application Types form, Applicants page.
  - Joint Applicant:

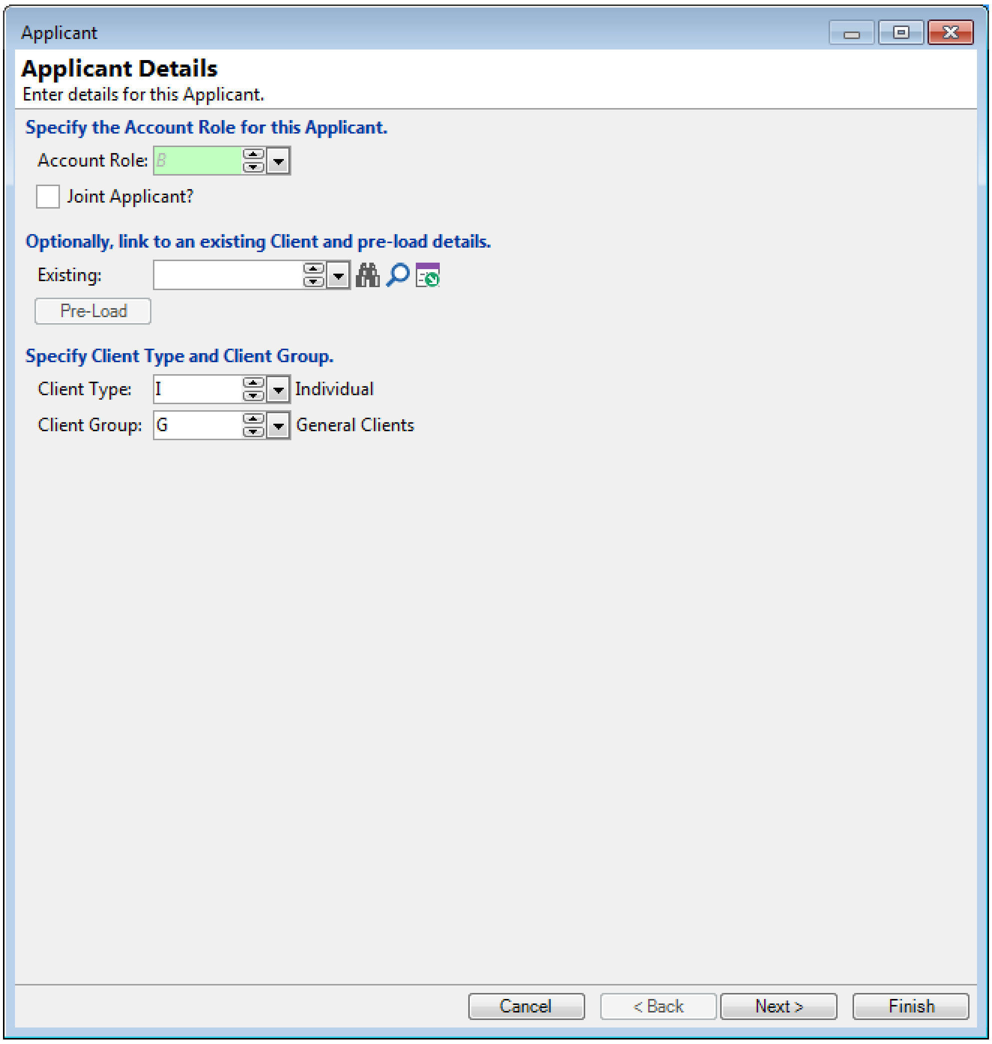
        - This will be used when generating an Account Client from this Applicant.
- **Client Details**
    - Existing:
        - The Client record that will be updated when accepting this Account Application (providing this Applicant is Used and the 'Update Existing Client' box is checked).
    - Loaded From:
        - The Client record that this Applicant's details have been pre-loaded from.
            - **NOTE:** Providing an 'Existing' Client is specified, this should be the same Client. If there is a mismatch, a warning will be given on the Applicant Summary page.
    - Update Existing Client when creating the Account:
        - Indicates whether to update the 'Existing' Client when creating the Account, i.e., accepting the Account Application.
            - **NOTE:** Certain details such as First and Last Name are never updated for existing Clients.
    - Created:
        - The Client record that was created from this Applicant.
            - Typically, this will be blank until the Application is accepted.
    - Pre-Load:
        - This will pre-load the Applicant's details from either the 'Loaded From' or 'Existing' Client.
    - Create Now/ Update Now:
        - If both the 'Existing' and 'Created' Clients are blank, this button will read 'Create Now' and will do the following:
            - Create a new Client record from this Applicant.
            - Set the 'Existing' and 'Created' Clients to the newly created Client.
        - If the 'Existing' Client is not blank, this button will read 'Update Now' and will do the following:
            - Update the 'Existing' Client from the Applicant details.
- **Details for New Client**
    - Client Type:
        - The Client Type to assign when creating a new Client from this Applicant.
            - If unspecified, this will be the Client Type defined on the Account Application Types form, Applicants page.
    - Client Group:
        - The Client Group to assign when creating a new Client from this Applicant.
            - If unspecified, this will be the Client Group defined on the Account Application Types form, Applicants page.
- **Enter Notes**
    - Any notes you wish to record against this Applicant record.

Applicant Wizard

This form is a cut-down version of the New Client wizard and is displayed upon clicking the Applicant drilldown button in the Applicants grid.

The Account Application Type can override this wizard by specifying an 'Applicant' Page Set on the Account Application Types form, Data Capture page.

NOTE: For Applicants that are 'Key Locked', editing of Applicants via this wizard can be disabled by unchecking the 'Allow Script maintained Applicants to be edited independently' option on the Applicants page of the Account Application Types wizard.

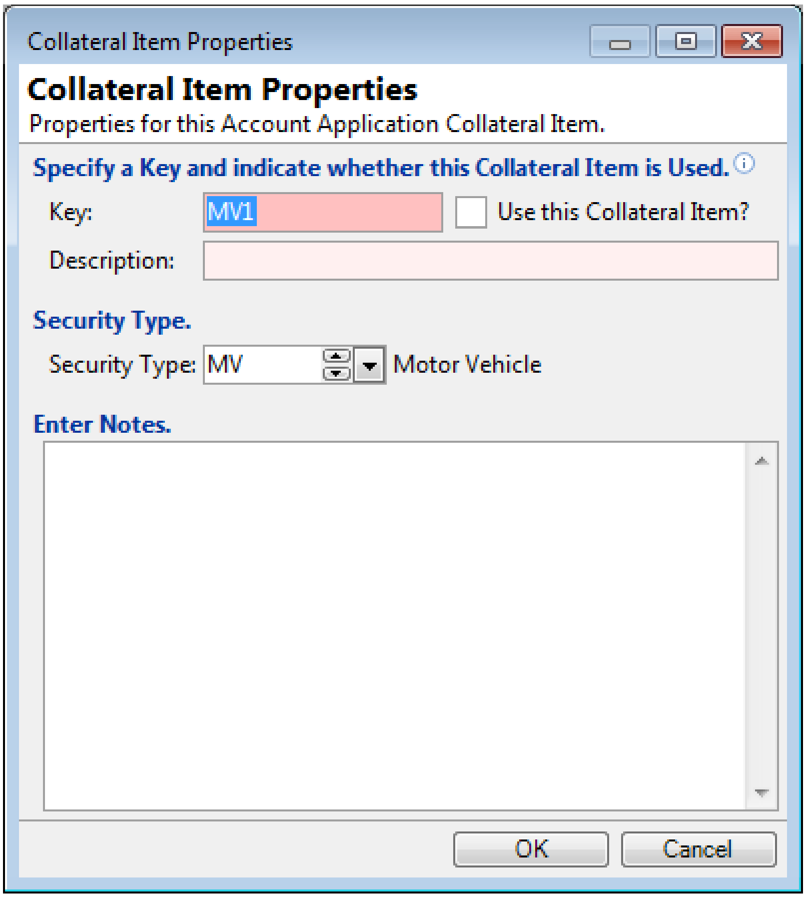


The first page of this wizard (displayed above) contains much of the same information as the Account Application Applicant form detailed in the previous section, including the ability to pre-load an existing Client's details.

The majority of the other wizard pages are self-explanatory and are similar to (although simpler than) the New Client wizard.

Collateral Item Properties

This form allows basic Collateral Item properties to be edited and is displayed upon clicking the ellipsis drilldown button in the Collateral grid.



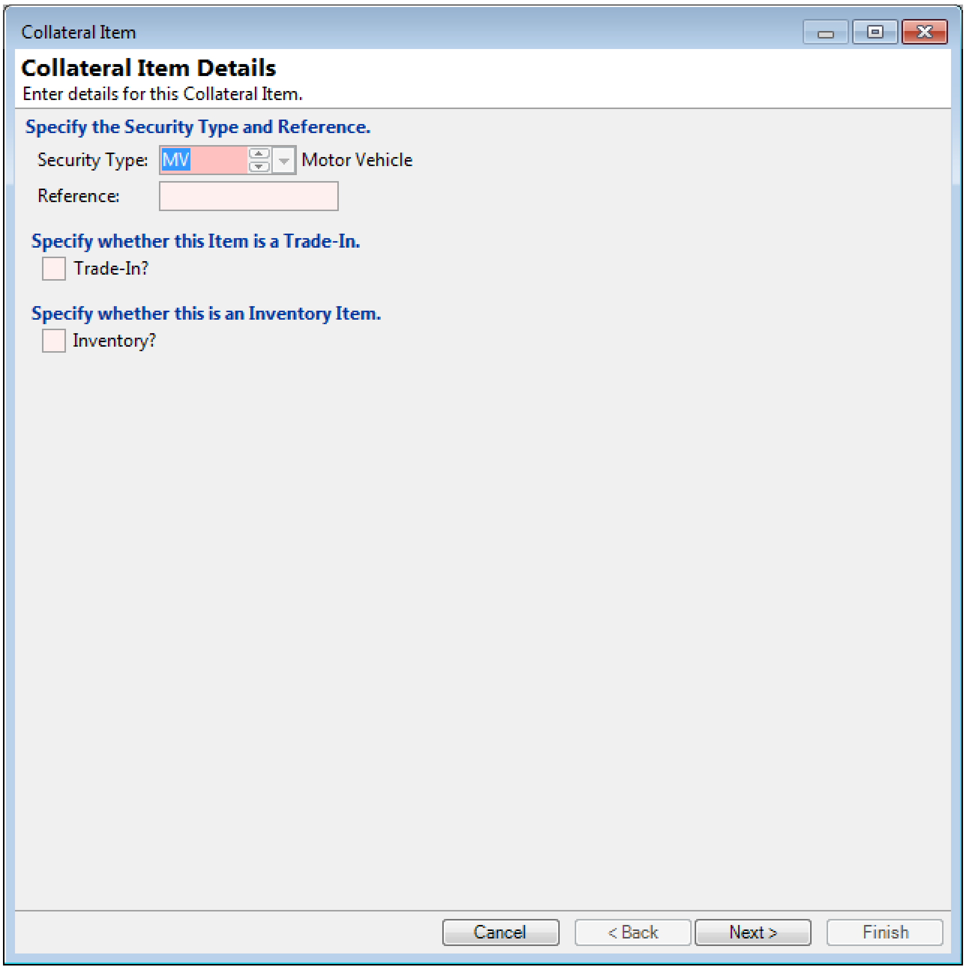
- **Specify a Key and indicate whether this Collateral Item is Used**
  - Key:
    - A unique code up to 20 characters long.
    - This will only generally have a value if the Collateral Item has been created from a Script.
    - **NOTE:** This will be read-only if this Collateral Item is 'Key Locked', i.e., a Script (Account Application Type of Page Set) has locked it since it relies on a Collateral Item with this Key existing for the Account Application.
  - Use this Collateral Item:
    - Indicates whether this Collateral Item is used, i.e., it should be included when creating a Security Statement for the Account when the Application is accepted.
    - Typically, if an Account Application Type is configured to allow entry (via a Page Set for example) of multiple Collateral Items, e.g., 'MV1', 'MV2' and 'TradeIn', the Account Application Type Script might create multiple Collateral Item records but flag one or more of them as not being used.
      - The Page Set could then update the 'Used' flag on the Collateral Item if details have been entered for it.
  - Description:
    - A read-only, auto-generated description for this Collateral Item.
- **Security Type**
  - Security Type:

        - The Security Type to use when generating a Security Statement Item from this Collateral Item.

- **Enter Notes**
    - Any notes you wish to record against this Collateral Item.

Collateral Item Wizard

This form is a cut-down version of the Security Item form but presented in wizard form. It is displayed upon clicking the Collateral Item drilldown button in the Collateral Items grid.



Script Objects

finAccountApp

Represents an Account Application.

- Main properties are similar to those on an Account (`finAccount`) object.
- Has collections:
  - Applicants
  - Calculations
  - CollateralItems
- Can produce an unsaved Account (`finAccount`) object at any time via the `CreateAccount()` method.

finAccountAppApplicant

Represents an Applicant.

- Main properties are similar to those on a Client (`finClient`) object.
  - No sub-collections like Clients though (e.g., no ContactMethods collection).
- Accessed via the `finAccountApp.Applicants` collection.
  - A Page Set would typically retrieve a '[Key Locked]' Applicant, e.g.:

```
ApplicantMain = AccountApp.Applicants.ItemByKey("MAIN")
```

- Can produce an unsaved Client (`finClient`) object at any time via the `CreateClient()` method.

finAccountAppCalculation

Represents a 'Quote'.

-

finAccountAppCollateralItem

Represents a Collateral Item.

- Main properties are similar to those of Security Statement Items but are more generic.
- Accessed via the `finAccountApp.CollateralItems` collection.
  - A Page Set would typically retrieve a '[Key Locked]' Collateral Item, e.g.:

```
MotorVehicle = AccountApp.CollateralItems.ItemByKey("MV")
```

Account Application Type Script

Overview

The Script code defined on an Account Application Type allows almost every aspect of an Account Application to be controlled. This includes:

- Initialisation of a new Account Application, e.g.:
  - Setting defaults.
  - Creating one or more default Applicants.
  - Starting a Workflow.
- Handling a change of status.
- Refining or fully controlling how the Account (or Accounts), Clients and Security Statements are created from the Account Application.
- Controlling and configuring the financial calculation.
- Validation, e.g., ensuring data is valid to create an Account, Client etc.

> **NOTE:** Calculation (financial) details are **ALWAYS** displayed and maintained via a `finAccount` object.
>
> This limits how easily the Account calculation can be customised directly from the Account Application but it does ensure that the Calculation should always be consistent on the final Account, e.g., any calculation Script defined by the Account Type will be used regardless of whether the financial details are being viewed from an Account Application or an Account.

Many of the Script events are concerned with creating an Account object. This is outlined in the next section

Creating an Account Object

An Account Application must be able to do the following:

- Create an Account object.
- Create new or update existing Clients from the Account Application's Applicants.
- Create a new Security Statement and add Security Statement Items from the Account Application's Collateral Items.

Built-in functionality allows all of the above to occur and the following Script events are called at various stages during the creating of a `finAccount` object. These are listed (where possible) in order:

- UpdateAccount
-

Script Events

Unlike most Scripts within finPOWER Connect, an Account Application Type Script does not have a 'Main' method. Instead it has pre-defined methods that can be used as events.

These 'events' often have default functionality, e.g., the `UpdateClientFromApplicant` event updates a `finClient` object's properties from a `finAccountAppApplicant`.

Often, you will want your event to first perform the default functionality, e.g., populate basic Client details such as Name and Date of Birth. Your Script will then update other properties on the `finClient` object, e.g., non-built-in information entered against an Applicant.

This is achieved by calling the 'Base' method before implanting your custom Script code. This is shown in some of the code samples included in this section.

All available events are described in this section. They are listed in a logical order rather than alphabetically.

NOTE: Although the Script events are represented by overridden Script methods, this section refers interchangeably to either 'event' or 'method'

WARNING: Many events can return `False` to indicate that they have failed.

Use this wisely since failing to perform an action such as updating an Account object may affect functionality such as the ability to view details on the Quotes page of the Account Applications form.

Initialise

This is an unusual event in that it is automatically called before every other event.

This gives the Script code a chance to set up variables (usually constants) that can then be used by any other method or event in the Script.

This event has no built-in functionality.

```
'#Const "PreferredLoanAmount",Currency,1500,"Preferred Loan Amount."
'#Const "PreferredTerm",String,"12 Months","Preferred Loan Term."
'#Const "PreferredPaymentFrequency",String,"Monthly","Preferred Payment Frequency."

' Constants
Private PreferredLoanAmount As Decimal
Private PreferredTerm As String
Private PreferredPaymentFrequency As String

Public Overrides Sub Initialise()

  ' Get Constants
  With ScriptInfo.Constants
    PreferredLoanAmount =.GetCurrency("PreferredLoanAmount", 1500)
    PreferredTerm =.GetString("PreferredTerm", "12 Months")
    PreferredPaymentFrequency = .GetString("PreferredPaymentFrequency", "Monthly")
  End With

End Sub
```

InitialiseAccountApp

When a new `finAccountApp` object is created, it must be initialised to use a specified Account Application Type via either its `Initialise` or `InitialisePk` methods.

These methods first set any defaults defined on the Account Type (currently none) and then call the `InitialiseAccountApp` event.

It is the job of this event to initialise the Account Application. This includes:

- Setting any defaults, e.g., the Preferred Term.
- Adding one or more default Applicants.
  - For example, entry of Account Applications of this type may use a Page Set that relies on a 'MAIN' Applicant existing. This Page Set can then refuse to run if this record does not exist.
  - The business layer contains helper functionality to achieve this, as shown in the Script sample below.
- Adding one or more default Collateral Items.
  - For example, entry of Account Applications of this type may use a Page Set that relies on a 'MOTORVEHICLE1' Collateral Item existing. This Page Set can then refuse to run if this record does not exist.
  - The business layer contains helper functionality to achieve this, as shown in the Script sample below.
- Adding additional calculations (Quotes).
  - Although the Account Application will automatically add a default calculation, in certain circumstances, you may wish to give the applicant a choice between multiple quotes, e.g., a 'Gold', 'Silver' or 'Bronze' option.

WARNING: Use this event and NOT the Page Set or external application to configure the default Account Application, e.g., to add 'MAIN' and 'JOINT' Applicants.

The event takes the following parameters:

- accountApp
  - The Account Application.

This event has no built-in default functionality.

```
Public Overrides Function InitialiseAccountApp(accountApp As finAccountApp) As Boolean

  ' Assume Success
  InitialiseAccountApp = True

  ' Set Defaults
  With accountApp
    .PreferredLoanAmount = PreferredLoanAmount
    .PreferredTerm = PreferredTerm
    .PreferredPaymentFrequency = PreferredPaymentFrequency
  End With

  ' Add a default Applicant
  With accountApp.Applicants.AddApplicant("MAIN", True, True)
    ' Only allow entry of one address so flag as Physical as well as Postal
    .AddressPhysicalIsPostal = True
  End With

  ' Add a default Collateral
  With accountApp.CollateralItems
    .AddCollateralItem("MOTORVEHICLE1", True, False, "MV")
    .AddCollateralItem("MOTORVEHICLE2", True, False, "MV")
    With .AddCollateralItem("MOTORVEHICLETR", True, False, "MV")
      .TradeIn = True
    End With
```

```
  End With
End Function
```

NOTE: Even if your `InitialiseAccountApp` method adds multiple Applicants and multiple Collateral Items, there is nothing to stop these as being flagged as not used, for example, you might add a 'MAIN' and a 'JOINT' Applicant but flag the 'JOINT' as being not used by default.

The following helper methods of the `finAccountAppApplicants` object can be used to add Applicants during initialisation:

- `AddApplicant`
  - This adds an Applicant to the collection and returns the new `finAccountAppApplicant` object. It takes the following parameters:
    - key
      - A unique key to identify this Applicant in the collection, e.g., 'MAIN' to represent the Main Applicant.
    - keyLocked (optional)
      - A Boolean value indicating whether this record is key-locked, i.e., the Applicant cannot be deleted or its key changed by the User.
      - The default value is `False`.
    - used (optional)
      - Indicates whether this Applicant is used. Typically this would be set to `True` for the main Applicant and `False` for any optional Applicants such as joint borrowers or guarantors.
      - The default value is `True`.
    - accountRoleId (optional)
      - The Id of the Account Role to use when creating an Account Client from this Applicant.
    - clientGroupId (optional)
      - The Id of the Client Group to use when creating a new Client from this Applicant.
    - clientTypeId (optional)
      - The Id of the Client Type to use when creating a new Client from this Applicant.

The following helper methods of the `finAccountAppCollateralItems` object can be used to add Collateral Items during initialisation:

- `AddACollateralItem`
  - This adds a Collateral Item to the collection and returns the new `finAccountAppCollateralItem` object. It takes the following parameters:
    - key
      - A unique key to identify this Collateral Item in the collection, e.g., 'MV' to represent the main Motor Vehicle.
    - keyLocked (optional)
      - A Boolean value indicating whether this record is key-locked, i.e., the Collateral Item cannot be deleted or its key changed by the User.
      - The default value is `False`.
    - used (optional)

        - Indicates whether this Collateral Item is used. Typically this would be set to `True` for the main Collateral Item (e.g., a Motor Vehicle in a Motor Vehicle loan) and `False` for any optional Collateral such as a 'Trade In' Motor Vehicle.
        - The default value is `True`.
    - securityTypeId (optional)
        - The Id of the Security Type to use when creating a Security Statement Item from this Collateral Item.

The following helper methods of the `finAccountAppCalculations` object can be used to add Calculations (Quotes) during initialisation:

- `AddACalculation`
    - This adds a calculation to the collection and returns the new `finAccountAppCalculation` object. It takes the following parameters:
        - key
            - A unique key to identify this Calculation in the collection, e.g., 'MV' to represent the main Motor Vehicle.

              NOTE: Calculations added automatically or via the User Interface are given an automatically generated key, e.g., 'Quote1', 'Quote2'.
        - keyLocked (optional)
            - A Boolean value indicating whether this record is key-locked, i.e., the Calculation cannot be deleted or its key changed by the User.
            - The default value is `False`.

SavePre

This event occurs when the Account Application is being saved.

If occurs after validation has been performed but before the database has been updated and outside of the database Transaction.

This event can be used to:

- Perform any additional validation.
  - The event can return `False` to cause the Save to fail.
- Tweaking Account Application values.

WARNING: It is not recommended that the Save is failed when the Account Application is still in its data-entry phase since Users expect to be able to save the Application at any time without having to worry about whether the data is valid or not.

The event takes the following parameters:

- accountApp
  - The Account Application.
- auditAccountApp
  - The Account Application prior to any changes being made.

This event has no built-in default functionality.

```
Public Overrides Function SavePre(accountApp As finAccountApp,
                                  auditAccountApp As finAccountApp) As Boolean

  ' Assume Success
  SavePre = True

  ' Ensure Branch specified when reviewing
  If accountApp.Status = isefinAccountAppStatus.Reviewing Then
    If accountApp.BranchPk = 0 Then
      SavePre = False
      finBL.Error.ErrorBegin("You must specify a Branch when reviewing.")
    End If
  End If

End Function
```

SaveAudit

This event occurs when the Account Application is being saved.

If occurs after validation has been performed but and is inside of the database Transaction.

This event can be used to:

- Perform any additional auditing
  - The event can return `False` to cause the Save to fail.

WARNING: Never attempt to save the Account Application from within this event.

The event takes the following parameters:

- accountApp
  - The Account Application.
- auditAccountApp
  - The Account Application prior to any changes being made.

This event has no built-in default functionality.

SavePost

This event occurs when the Account Application is being saved.

If occurs after the database has been updated and outside of the database Transaction.

Returning `False` from this event will not cause the Save to fail. However, it will set the Save Method's `ByRef savePostFailed` parameter to `True`.

This event can be used to:

- Update an external application.
- Send a notification (e.g., an Email) to the Account Manager.

The event takes the following parameters:

- accountApp
  - The Account Application.

This event has no built-in default functionality.

```
Public Overrides Function SavePost(accountApp As finAccountApp,
                                   auditAccountApp As finAccountApp) As Boolean

  Dim Message As String

  ' Assume Success
  SavePost = True

  ' Send Email if Branch changed
  If accountApp.BranchPk <> auditAccountApp.BranchPk Then
    ' Create Message
    Message = String.Format("User '{0}' has changed the Branch from '{1}' to '{2}'.",
                            finBL.Users.ItemByPkSafe(accountApp.UpdatedUserPk).UserId,
                            auditAccountApp.BranchId,
                            accountApp.BranchId)

    ' Send
    SavePost = finBL.SendEmail("ph@intersoft.co.nz", "", "",
                               String.Format("Branch Changed on '{0}'", accountApp.AccountAppId),
                               Message,
                               "",
                               False,
                               iseMessageTarget.Send)

  End If

End Function
```

GetAccountType

This job of this event is to return an Account Type and Promotion to use for the supplied Quote (`finAccountAppCalculation`).

It can optionally return a list of valid Account Types which could then be used by a Page Set or other mechanism to allow a User to select an Account Type from a dropdown list.

This event is called when creating an Account (a `finAccount` object) for an Account Application; this is something that occurs regularly, e.g., when showing a summary on the Quotes page of the Account Applications form.

At the point at which this event is called, the `finAccount` object will have already been created and other properties such as Clients and the Security Statement created.

The event takes the following parameters:

- accountApp
  - The Account Application.
- accountAppCalculation
  - The Calculation (or Quote) to retrieve the Account Type for.
- account
  - The Account object that is being created.

    NOTE: The Script may wish to examine this, e.g., so that it can return a different Account Type based on the number of Account Clients.
- ByRef accountTypeId
  - The Id of the Account Type to use.
- ByRef accountTypePromotionId
  - The Id of the Account Type Promotion to use if licensed for the Promotions Add-On.
- ByRef accountWarnings
  - The warnings collection.
- ByRef validAccountTypeIds
  - A list of valid Account Types. This can be generated by the Script and then used by a Page Set (or other mechanism) to restrict which Account Types the User can select.

This event has the following built-in functionality:

- If the Account Application has a preferred Account Type defined (`finAccountApp.PreferredAccountTypeId`) then:
  - Set the Account Type Id to this and the Promotion Id to the preferred Promotion (`finAccountApp.PreferredAccountTypePromotionId`).
- Otherwise:
  - Set the Account Type Id to the default as defined on the Quotes page of the Account Application Types form.

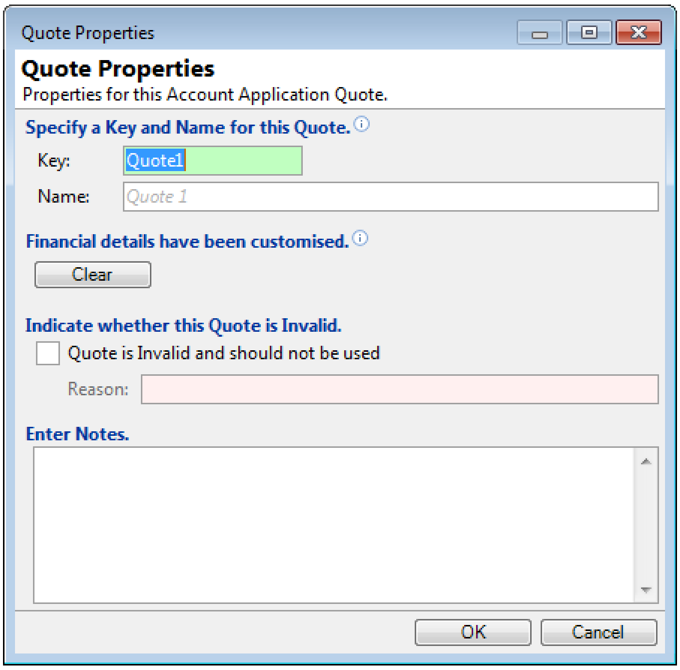
Script code can return whatever custom values, e.g., your Account Application Type may wish to determine an Account Type based on other User input, e.g., whether they have selected the 'Gold', 'Silver' or 'Bronze' plans (in this case, these are represented by three separate calculations) as per the following example:

```
Public Overrides Sub GetAccountType(accountApp As finAccountApp,
                                    accountAppCalculation As finAccountAppCalculation,
                                    account As finAccount,
                                    ByRef accountTypeId As String,
                           Optional ByRef accountTypePromotionId As String = "",
                           Optional ByRef validAccountTypeIds As ISList = Nothing)
```

```
  Select Case accountAppCalculation.Key
    Case "Gold"
      accountTypeId = "VL"
      accountTypePromotionId = "Gold"

    Case "Silver"
      accountTypeId = "VL"
      accountTypePromotionId = "Silver"

    Case "Bronze"
      accountTypeId = "VL"
      accountTypePromotionId = "Bronze"

    Case Else
      ' Call Base Method to decide what Account Type to use
      MyBase.GetAccountType(accountApp,
                            accountAppCalculation,
                            account,
                            accountTypeId,
                            accountTypePromotionId,
                            validAccountTypeIds)
  End Select

End Sub
```

WARNING: If the supplied calculation has saved financial details (i.e., `finAccountAppCalculation.HasSavedCalculationDetails` is `True`), these will always overwrite the Account Type and Promotion returned from this event.

The `finAccountAppCalculation` object has a `CalculationReset` method that allows these overriding details to be cleared. This can be performed from Script code or via the 'Clear' button on the 'Quote Properties' form:



UpdateAccount

This job of this event is to update the supplied `finAccount` object from an Account Application (`finAccountApp`).

WARNING: This event should update only the Account and should not touch the Account Calculation, Clients or Security Items since these are handled by other events.

This event is called when creating an Account (a `finAccount` object) for an Account Application; this is something that occurs regularly, e.g., when showing a summary on the Quotes page of the Account Applications form.

The event takes the following parameters:

- accountApp
  - The Account Application.
- account
  - The Account object that is being created.

This event has the following built-in functionality:

- Set all general Account details from the Account Application, e.g., Date Quoted, Name, Purpose, Branch, Dealer etc.

NOTE: Your custom Script code will almost always want to call the base functionality and then perform any custom functionality such as updating other `finAccount` properties or User Data based upon values stored in the Account Application.

```
Public Overrides Function UpdateAccount(accountApp As finAccountApp,
                                        account As finAccount) As Boolean

  ' Call Base Method to update the Account
  UpdateAccount = MyBase.UpdateAccount(accountApp, account)

  ' Set other properties, e.g., UserData to record additional information from Application
  account.UserData.SetString("ReferredBy", accountApp.UserData.GetString("ReferredBy"))

End Function
```

WARNING: This event is called when an Account is being created and occurs before other objects such as Clients and Security Statements are created/ updated.

Use the FinaliseAccount event occurs at the end of the Account creation and can therefore be used to tweak these objects.

UpdateAccountCalc

This job of this event is to update a supplied Account Calculation (`finAccountCalc)` object from an Account Application Quote (`finAccountAppCalculation`).

This event is called when creating an Account (a `finAccount` object) for an Account Application; this is something that occurs regularly, e.g., when showing a summary on the Quotes page of the Account Applications form.

This occurs after the calculation has been initialised using the Account Type details returned from the GetAccountType event.

The event takes the following parameters:

- accountApp
  - The Account Application.
- accountAppCalculation
  - The Account Application Calculation from which the Account is being created.
- accountCalc
  - The Account Calculation to update.
- defaultFromAccountApp
  - Indicates whether to default the Account Calculation Details from the Account Application.

    NOTE: This will be False if the Account Application Calculation has overriding financial details, i.e., `finAccountAppCalculation.HasSavedCalculationDetails` is `True`.

This event has the following built-in functionality:

- Update the `finAccountCalc` object as follows:
  - If the Account Application Quote has complex financial details, e.g., as a result of editing the financial details via the special 'Edit Financial' wizard which allows the same functionality as the Financial page of the New Account wizard, the `finAccountAppCalculation.HasSavedCalculationDetails` property will be set to True and the calculation will be updated from these details.
  - Otherwise, the following will occur:
    - Amount financed (Standard Transaction 0) will be set to:
      - `finAccountApp.OfferedLoanAmount` if non-zero, otherwise, `finAccountApp.PreferredLoanAmount`.
  - The Term (or Number of Payments, depending on the Account Type) will be updated from the following properties of the `finAccountApp` object.
    - `PreferredNumberOfPayments`
    - `PreferredTerm`
  - The Payment Cycle will be updated to `finAccountApp.PreferredPaymentFrequency`.

Your custom Script code will almost always want to call the base functionality first and then perform any custom updates to the Calculation, e.g., you might have a Trade In amount stored in the Account Application's User Data for which you want to add an additional Standard Transaction to the calculation.

```
Public Overrides Function UpdateAccountCalc(accountApp As finAccountApp,
                                            accountAppCalculation As finAccountAppCalculation,
                                            accountCalc As finAccountCalc,
                                            defaultFromAccountApp As Boolean) As Boolean

  ' Call Base Method to update the Account Calculation
  UpdateAccountCalc = MyBase.UpdateAccountCalc(accountApp,
                                               accountAppCalculation,
                                               accountCalc,
```

```
                                                defaultFromAccountApp)

  ' Update properties from Account Application
  ' NOTE: This is pretty much what the base functionality does but is repeated here
  '       for clarity.
  If accountAppCalculation.Index = 0 Then
    With accountCalc
      .StandardTransactions(0).ValueCurrency = accountApp.PreferredLoanAmount
      .PaymentCycle = accountApp.PreferredPaymentFrequency
      .TermCycle = accountApp.PreferredTerm
    End With
  End If

End Function
```

AfterCalculationUpdateFromAccount

This event is only called after the Account Application's Quote is updated from an Account, e.g., it has been edited via the special 'Edit Financial' wizard which allows the same functionality as the Financial page of the New Account wizard.

At this point, the `finAccountAppCalculation` object's `HasSavedCalculationDetails` property will always return `True` indicating that this Quote holds complex financial details.

This job of this event is to update the Account Application's properties, if desired, to reflect the Account Calculation values, e.g., if this is the Quote used in a Page Set, it may be intuitive to update the Account Application's Preferred Loan Amount or Offered Loan Amount properties to match the complex financial details now stored on the Quote.

The event takes the following parameters:

- accountApp
  - The Account Application.
- accountAppCalculation
  - The Account Application Calculation to update.
- account
  - The Account object to update the Account Application from.

This event has no built-in default functionality.

```
Public Overrides Function AfterCalculationUpdateFromAccount(accountApp As finAccountApp,
                                                            accountAppCalculation As finAccountAppCalculation,
                                                            account As finAccount) As Boolean

  ' Assume Success
  AfterCalculationUpdateFromAccount = True

  ' Update properties on Account Application (only if Quote used by Page Set)
  ' NOTE: This is not strictly necessary but it keeps things intuitive
  If accountAppCalculation.Index = 0 Then
    With account.Calculation
      accountApp.PreferredAccountTypeId = .AccountTypeId
      accountApp.PreferredLoanAmount = .StandardTransactions(0).ValueCurrency
      accountApp.PreferredPaymentFrequency = .PaymentCycleDescription
      accountApp.PreferredTerm = .TermCycleDescription
    End With
  End If

End Function
```

NOTE: This method can return `False` to indicate that it failed to update the Account Application Calculation.

UpdateAccountAfterSave

This event is used when an Account object has been created from an Account Application (from Accepting the Application).

This event is called directly after the Account has been saved and 'Opened' (and runs within a database transaction) and allows any special tweaks or other operations to be performed that should only be performed when the database is updated.

The event takes the following parameters:

- accountApp
  - The Account Application.
- account
  - The Account that has just been created, saved and 'Opened'.

This event's base functionality does nothing.

```
Public Overrides Function UpdateAccountAfterSave(accountApp As finAccountApp,
                                                  account As finAccount) As Boolean

  ' Assume Success
  UpdateAccountAfterSave = True

  ' Update Account
  account.Notes = "Updated after saving!"

End Function
```

The `account` object will automatically be saved after this event has run if any changes have been made to it.

WARNING: Use this event only when you need to do something to the Account that requires the Account to exist on the database.

This only applies the Account created automatically via the Account Accept Decline wizard.

ValidateAccount

This job of this event is to validate an Account that has been created from an Account Application using a particular Account Application Calculation (Quote).

This event is called from the `finAccountAppTypeRO.ValidateAccount` method which is called from certain places in the User Interface, e.g., when producing a Summary Page for displaying on the Quotes page of the Account Applications form, this method will be called to generate a warning that can be displayed at the top of the Summary Page.

The event takes the following parameters:

- accountApp
  - The Account Application.
- accountAppCalculation
  - The calculation (or 'Quote') used to create the Account.
- account
  - The unsaved Account that has been created.
- ByRef warningSeverity
  - The severity of the warning message.

    NOTE: If the severity is 'Fail' then this will prevent the Account Application from being completed and an Account from being created.
- ByRef warningMessage
  - The warning message to display.

This event's base functionality does the following:

- If the supplied Quote has been flagged as Invalid by the User, the Warning Message returned will reflect this and the Warning Severity will be set to 'Fail' to indicate that this Account is not valid for saving.

```
Public Overrides Sub ValidateAccount(accountApp As finAccountApp,
                                     accountAppCalculation As finAccountAppCalculation,
                                     account As finAccount,
                                     ByRef warningSeverity As isefinAccountWarningSeverity,
                                     ByRef warningMessage As String)

  ' Call Base Method
  MyBase.ValidateAccount(accountApp,
                         accountAppCalculation,
                         account,
                         warningSeverity,
                         warningMessage)

  ' Warn if Quote values do not match the Preferred Values in the wizard
  ' NOTE: Perfectly valid with multiple Quotes but its nice to give a warning
  If Len(warningMessage) = 0 Then
    With account.Calculation
      If account.Calculation.StandardTransactions.Count <> 0 AndAlso
         accountApp.PreferredLoanAmount <>
         account.Calculation.StandardTransactions(0).ValueCurrency Then
        warningSeverity = isefinAccountWarningSeverity.Alert
        warningMessage = String.Format("Preferred Amount ({0}) doesn't match Quote amount ({1}).",
                         finBL.FormatCurrency(accountApp.PreferredLoanAmount, True),
               finBL.FormatCurrency(account.Calculation.StandardTransactions(0).ValueCurrency,
True))
      End If
    End With
  End If

End Sub
```

UpdateClientFromApplicant

This event is used during the Account creation process to update either a new or existing Client record from an Applicant.

This event is called when creating or updating a Client (a `finClient` object) for an Applicant and would generally be used to update the Client with non-standard Applicant properties, e.g., by setting the Client's User Data to record or by creating a new Identification Item based on details recorded against the Applicant.

The event takes the following parameters:

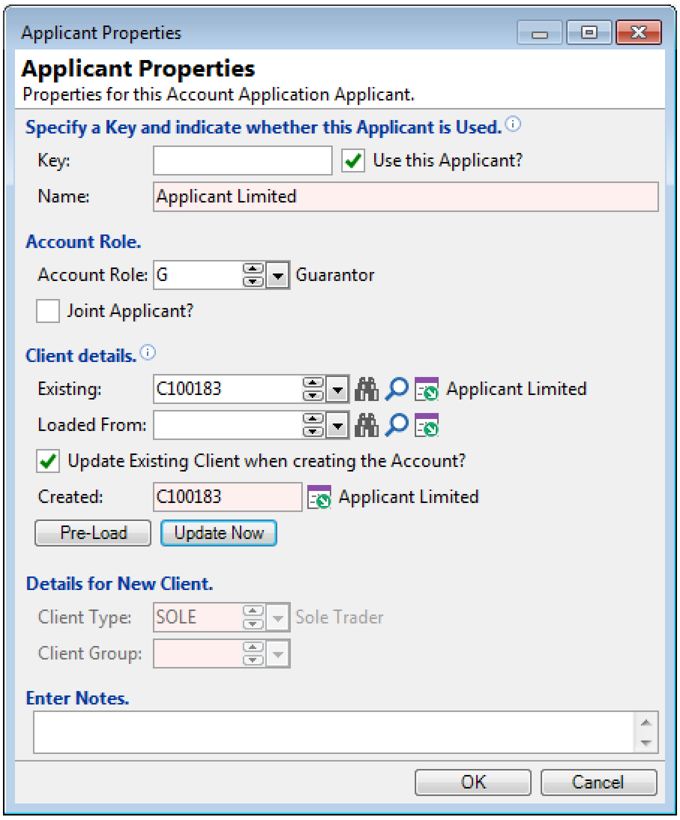
- accountAppApplicant
  - The Applicant.
- client
  - The Client to update.

This event's base functionality does the following:

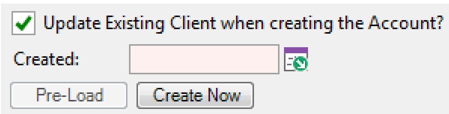
- Updates standard Client properties and collections (such as Contact Methods) from the Applicant

```
Public Overrides Function UpdateClientFromApplicant(accountAppApplicant As finAccountAppApplicant,
                                                    client As finClient) As Boolean

  Dim ClientIdentificationItem As finClientIdentificationItem

  ' Call Base Method to update the Client
  UpdateClientFromApplicant = MyBase.UpdateClientFromApplicant(accountAppApplicant, client)

  ' Add Identification Item for 'Health Card'
  If Len(accountAppApplicant.UserData.GetString("HealthCardNumber")) <> 0 Then
    ClientIdentificationItem = client.IdentificationItems.CreateClientIdentificationItem()
    With ClientIdentificationItem
      .IdentificationTypeId = "EB"
      .Value = accountAppApplicant.UserData.GetString("HealthCardNumber")
    End With
    client.IdentificationItems.Add(ClientIdentificationItem)
  End If

End Function
```

From an Account Application, a Client can be created or updated from the Applicant details at any point from a button on the Applicant Properties form:



Or, if the Applicant does not link to an existing Client, the button will read 'Create Now' instead:



UpdateClientAfterSave

This event is used when a Client object has been created from an Applicant or an existing Client object has been updated from an Applicant.

This event is called directly after the Client has been saved (and runs within a database transaction) and allows any special tweaks or other operations to be performed that should only be performed when the database is updated.

The event takes the following parameters:

- accountAppApplicant
  - The Applicant.
- client
  - The Client that has just been saved after being created or updated.

This event's base functionality handles copying an Applicant's Image File to the Document Manager for the Client, something that can only be done once a new record has been saved.

```
Public Overrides Function UpdateClientAfterSave(accountAppApplicant As finAccountAppApplicant,
                                                client As finClient) As Boolean

  ' Assume Success
  UpdateClientAfterSave = True

  ' Update Client
  client.Notes = "Updated after saving!"

End Function
```

The `client` object will automatically be saved after this event has run if any changes have been made to it.

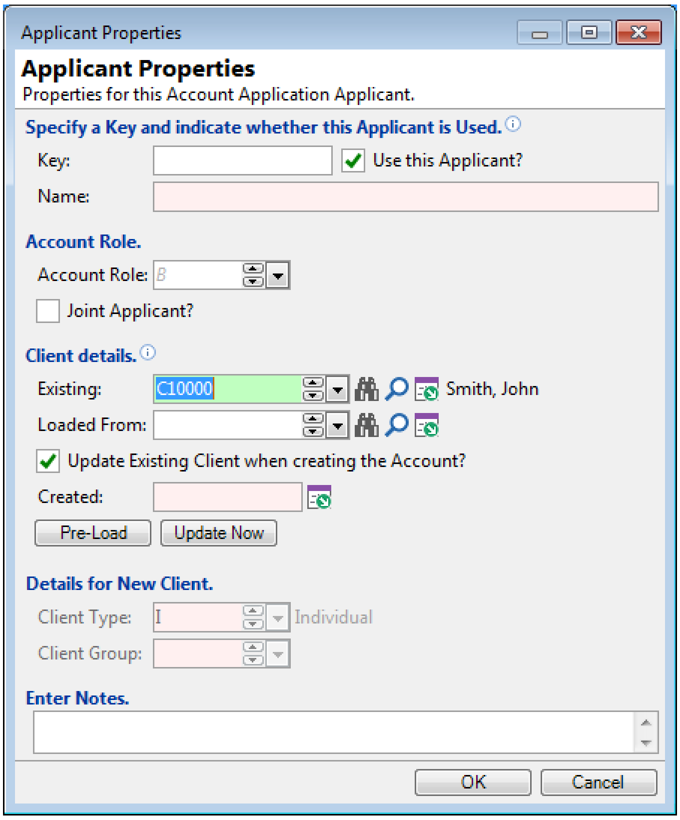
WARNING: Use this event only when you need to do something to the Client that requires the Client to exist on the database or have been updated on the database.

From an Account Application, a Client can be created or updated from the Applicant details at any point from a button on the Applicant Properties form.

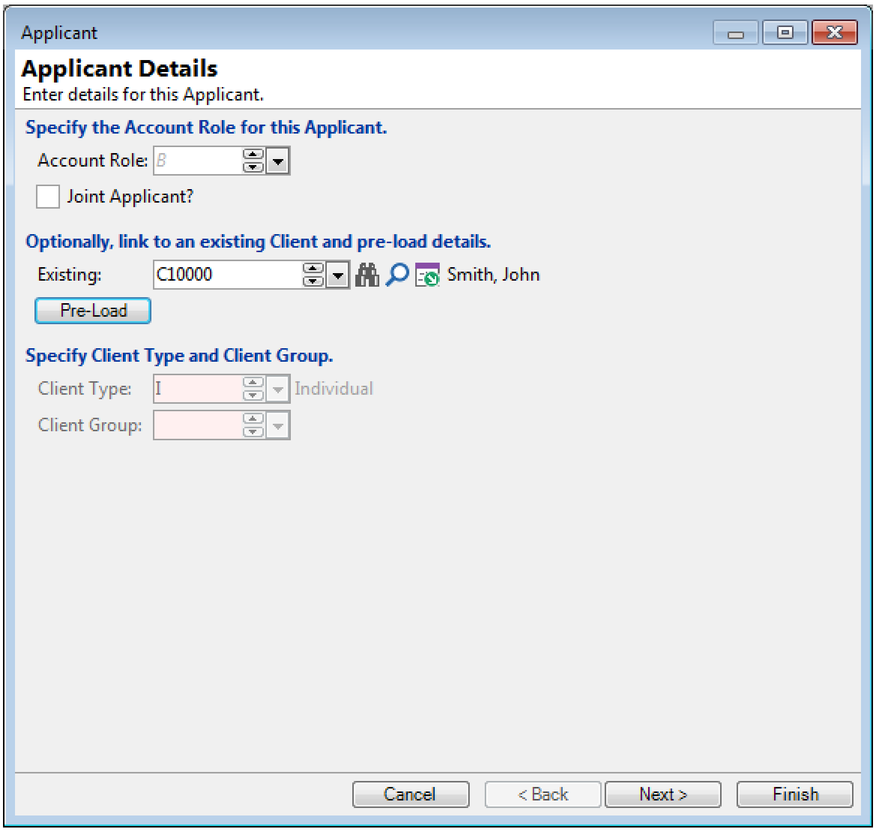
UpdateApplicantFromClient

This event is used when pre-loading the Applicant details from an existing Client record.

Pre-loading would generally be performed via a Page Set but can also be done via the built-in Applicant Properties form by selecting a Client and clicking the 'Pre-Load' button, e.g.:



Or via the first page of the built-in Applicant wizard, e.g.:



This event would generally be used to update the Applicant with non-standard Client properties, e.g., by setting the Applicant's User Data to values held on the Client record.

The event takes the following parameters:

- accountAppApplicant
  - The Applicant to update.
- client
  - The existing Client.

This event's base functionality does the following:

- Updates standard Applicant properties from the Client's properties and collections (such as Contact Methods).

```
Public Overrides Function UpdateApplicantFromClient(accountAppApplicant As finAccountAppApplicant,
                                                    client As finClient) As Boolean

  Dim ClientContactMethod As finClientContactMethod

  ' Call Base Method to update the Applicant
  UpdateApplicantFromClient = MyBase.UpdateApplicantFromClient(accountAppApplicant, client)

  ' Update Skype address from Client Contact Method
  For Each ClientContactMethod In client.ContactMethods
    If ClientContactMethod.IsCurrent() AndAlso ClientContactMethod.ContactMethodId = "SKYPE" Then
      accountAppApplicant.UserData.SetString("Skype", ClientContactMethod.Value)
      Exit For
    End If
  Next

End Function
```

UpdateSecurityStmt

'TODO:

FinaliseAccount

This event can make any last-minute changes to the Account or it's related Client and Security Statement objects.

This event is called when creating an Account (a `finAccount` object) for an Account Application; this is something that occurs regularly, e.g., when showing a summary on the Quotes page of the Account Applications form.

The event takes the following parameters:

- accountApp
  - The Account Application.
- account
  - The Account object that is being created.

This event does not have any built-in functionality.

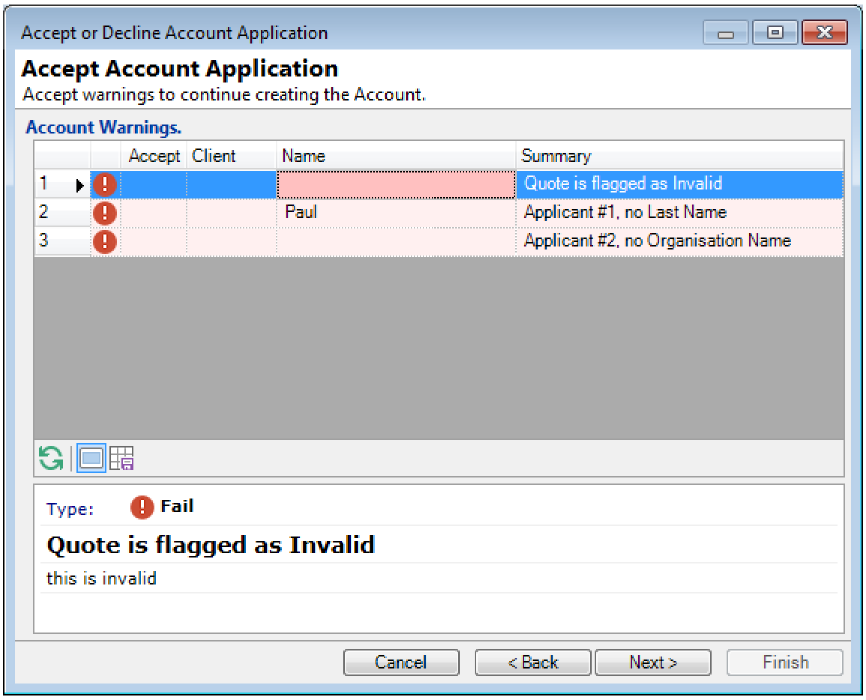
NOTE: Your custom Script code can perform tasks such as searching through the Security Statement Items and setting the Insurance Renewal Date to match the Account's Maturity date.

```
Public Overrides Function FinaliseAccount(accountApp As finAccountApp,
                                          account As finAccount) As Boolean

  ' Assume Success
  FinaliseAccount = True

  ' Make any last-minute changes to the Account or it's related Clients and Security Statements
  ' but do not save the Account

End Function
```

GetAccountWarnings

This event allows the warnings shown in the Accept or Decline Account Application wizard to be modified.

The warnings are shown on the second page of the wizard and are only generated if the User is attempting to Accept the Account Application, e.g.:



The event takes the following parameters:

- accountApp
  - The Account Application which is being accepted.
- account
  - The unsaved Account that has been created.
- includeAccountAppWarnings
  - Indicates whether to include Account Application-related warnings, e.g., you are trying to use a Quote flagged as 'Invalid'.
- includeAccountTypeWarnings
  - Indicates whether to add any Account Type warnings generated from the Script defined on the Account Type.
- ByRef accountWarnings
  - The warnings collection.

This event's base functionality will populate the accountWarnings collection, e.g.:

- Add any Account Application warnings that would prevent the Account from being created such as:
  - You are trying to use a Quote flagged as 'Invalid'.
  - The Applicant has insufficient details to create a Client, e.g., the Last Name has not been entered.
- Add any Account Type warnings generated from the Script defined on the Account Type.

The following sample gets a list of all built-in warnings and then adds a custom warning of each avialable type (Fail, Warning or Information):

```
Public Overrides Function GetAccountWarnings(accountApp As finAccountApp,
                                             account As finAccount,
                                             includeAccountAppWarnings As Boolean,
                                             includeAccountTypeWarnings As Boolean,
                                             ByRef accountWarnings As finAccountWarnings) As
Boolean

  ' Call Base Method to get Account Application Warnings and Account Warnings
  GetAccountWarnings = MyBase.GetAccountWarnings(accountApp,
                                                 account,
                                                 includeAccountAppWarnings,
                                                 includeAccountTypeWarnings,
                                                 accountWarnings)

  ' Add new Warnings
  accountWarnings.AddFail("test fail")
  accountWarnings.AddAlert("test alert", account.Clients(0).Client, "Details")
  accountWarnings.AddInformation("test information")

End Function
```

'Fail' type warnings will prevent the User from moving to the last page of the wizard and there prevent them from Accepting the Account Application.

When adding warnings, the second parameter is the Client to which the warning applies.

NOTE: Although this method can return `False` to indicate that it failed, this would rarely, if ever, be used.

BeforeStatusChange

This event is called before the Account Application Status is updated, e.g., changed from Incomplete to Reviewing.

Common uses of this event are:

- Prevent the status being changed (i.e., set the error message and return `False`), e.g., if insufficient information has been entered.

The event takes the following parameters:

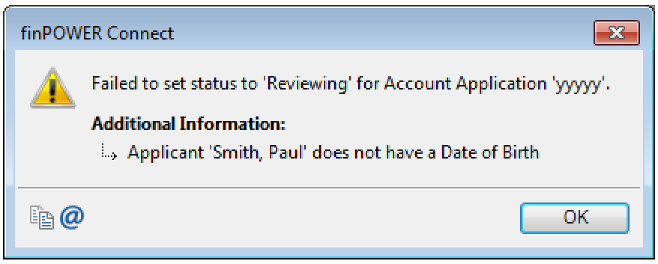
- accountApp
  - The Account Application.
- newStatus
  - The New Status.

This event has no built-in base functionality.

The following example prevents the status from being changed to 'Reviewing' unless all Applicants' dates of birth have been entered.

```
Public Overrides Function BeforeStatusChange(accountApp As finAccountApp,
                                             newStatus As isefinAccountAppStatus) As Boolean

  Dim AccountAppApplicant As finAccountAppApplicant

  ' Assume Success
  BeforeStatusChange = True

  If newStatus = isefinAccountAppStatus.Reviewing Then
    ' Check all 'Used' Applicants have Date of Birth
    For Each AccountAppApplicant In accountapp.Applicants
      With AccountAppApplicant
        If .Used Then
          If .DateOfBirth = Nothing Then
            BeforeStatusChange = False
            finBL.Error.ErrorBeginFormat("Applicant '{0}' does not have a Date of Birth", .Name)
            Exit For
          End If
        End If
      End With
    Next
  End If

End Function
```

Upon clicking the 'Data Entry Complete' action on the Account Applications form, the User will receive an error message, e.g.:



AfterStatusChange

This event is called after the Account Application Status has been updated, e.g., changed from Incomplete to Reviewing.

NOTE: This event occurs outside of the database transaction in which the Account Application's Status is changed. Therefore, returning `False` will not prevent the status change. The 'BeforeStatusChange' event must be used to achieve this.

Common uses of this event are:

- Set a property on the Account Application, e.g., the Review Date so that it appears in the Task Manager.
- Set the Account Application's sub-status (`StatusSub` property).
- Start a Workflow.

The event takes the following parameters:

- accountApp
  - The Account Application.
- oldStatus
  - The Old Status.

This event has no built-in base functionality.

The following example sets the sub-status when the status is changed to 'Reviewing'. It also clears the sub-status if the status is changed from 'Reviewing' back to 'Incomplete'.

```
Public Overrides Function AfterStatusChange(accountApp As finAccountApp,
                                            oldStatus As isefinAccountAppStatus) As Boolean

  ' Assume Success
  AfterStatusChange = True

  If accountApp.Status = isefinAccountAppStatus.Reviewing AndAlso
     oldStatus = isefinAccountAppStatus.Incomplete Then
    ' Set sub-status if 'Reviewing'
    accountApp.StatusSub = "Part 1"
    AfterStatusChange = accountApp.Save()
  ElseIf accountApp.Status = isefinAccountAppStatus.Incomplete AndAlso
     oldStatus = isefinAccountAppStatus.Reviewing Then
    ' Clear sub-status if reset back to 'Incomplete'
    accountApp.StatusSub = ""
    AfterStatusChange = accountApp.Save()
  End If

End Function
```

NOTE: Unlike most other events, the Script must explicitly save any changes made to the Account Application.

The sub-status is displayed on the General page of the Account Application form if it is non blank, e.g.:

Typically, the sub-status would be updated from a Workflow.

AcceptDeclineAcceptStatusSet

This event is called after an Account Application has been loaded into the `finAccountAppAcceptDecline` object or after the `AccountAppAcceptStatusSet` method has been called on this object.

Common uses of this event are:

- Tweak properties on the Account/ Decline object, e.g.:
  - Default the `AccountAppAcceptStatus` property to default to 'Decline' the Application.
  - Set the `CreateClientsWhenNotAccepted` property to True to create Clients, even when not accepting the Application.
    - **NOTE:** The default for this, and other properties, can be defined on the Account Application type.
- Prevent a particular Account Application from being loaded, e.g., due to custom permissions for the current User or the state of the Application (maybe by checking the `StatusSub` property of the Account Application).

The event takes the following parameters:

- accountAppAcceptDecline
  - The `finAccountAppAcceptDecline` object.
- accountAppLoaded
  - Indicates whether this event is being called because an Account Application is being loaded (as opposed to the `AccountAppAcceptStatusSet` method being called).

This event has no built-in base functionality.

The following example tweaks the 'Reason' and, if the Application is 'Unwanted', defaults to NOT create any new Client records.

```
Public Overrides Function AcceptDeclineAcceptStatusSet(accountAppAcceptDecline As
finAccountAppAcceptDecline,
                                                       accountAppLoaded As Boolean) As Boolean

  ' Assume Success
  AcceptDeclineAcceptStatusSet = True

  ' Tweak properties
  With accountAppAcceptDecline
    ' Update reason
    .AuthorisedReason = "Application processed by " & finBL.CurrentUser.UserId

    ' Don't create new Clients if 'Unwanted'
    If .AccountAppAcceptStatus = isefinAccountAcceptStatus.Unwanted Then
      .CreateClientsWhenNotAccepted = False
    End If
  End With

End Function
```

NOTE: This event should not be used for functionality such as preventing a User from declining an Application, e.g., due to insufficient information being entered; Use the GetAccountWarnings event for this.

WARNING: Never call the `AccountAppAcceptStatusSet` method of the **`finAccountAppAcceptDecline`** object from this event.

BeforeAccept

This event is called when the Account Application is being accepted via the Accept/ Decline wizard and occurs directly before the Account Application is accepted.

It may be used to override the Account or Accounts being created.

NOTE: This event occurs inside of the database transaction. Therefore, returning `False` will cause the Accept to fail.

Common uses of this event are:

- Bypass the creation of the normal Account to manually add an Account.
- Create multiple Accounts.
  - An Account Application may create multiple Accounts.
  - An Account may be linked to multiple Applications, e.g. where used to create a "topup".

The event takes the following parameters:

- accountAppAcceptDecline
  - The Account Application Accept Decline object.

This event has no built-in base functionality.

Creating multiple Accounts

It is possible to create multiple Accounts from an Account Application, or conversely link an Account to multiple Account Applications.

To do this you need to create or update Accounts as required and populate the finAccountAppAcceptDecline.AccountAppAcceptAccounts collection with the Accounts you have created/ updated.

WARNING: If any items are added to the AccountAppAcceptAccounts collection the Accept process assumes Accounts have been added/ updated externally and DOES NOT add Accounts as per the normal process.

Each finAccountAppAcceptAccount item you add should have the following properties set so that they are saved against the Application:

- AccountPk
  - Must be set to the Pk of the Account added or updated.
- AccountTypePk
- AccountTypePromotionPk
- For Loan Accounts:
  - TotalAdvances
  - TotalCosts
  - TotalDeposits
  - TotalInterest
  - TotalPayments
- For Deposit Accounts:
  - TotalInvestment
  - TotalInvestmentAdd

- TotalInterest

In addition, finAccountAppAcceptAccount includes the following properties:

- Custom
  - Will always be set to True if adding items to the collection.
- AccountSetCreationMethodToNewExApplication
  - Set to False if the Account's creation method should NOT be updated to "NewExApplication", e.g. if a topup to an existing Account.
- UserData (ISKeyValueList)
  - Use to store additional information. E.g. if a topup to an existing Account you might store the AccountTransactionPk of the Withdrawal transaction.

NOTE: Unlike most other events, the Script must explicitly save any changes made to the Account Application.

MonitorCategoryChange_ExecuteCommitPre

This event is called at the beginning of an Account Application Monitor Category Change being committed.

Common uses of this event are:

- Prevent the change from being committed.

The event takes the following parameters:

- accountAppMonitorCategoryChange
  - The `finAccountAppMonitorCategoryChange` object.

This event has no built-in base functionality.

NOTE: This event takes place outside of the main Database Transaction.

MonitorCategoryChange_ExecuteCommitPost

This event is called at the end of an Account Application Monitor Category Change being committed.

Common uses of this event are:

- Alert a User to the change or write a Log.

The event takes the following parameters:

- accountAppMonitorCategoryChange
  - The `finAccountAppMonitorCategoryChange` object.

This event has no built-in base functionality.

NOTE: This event takes place outside of the main Database Transaction, after the change has been committed.

If this event returns `False`, this will not fail the commit unless an outer Database Transaction is being used.

Summary Pages

Account Applications introduce several new, built-in Summary Pages and also modifications to existing Summary Pages.

Account, Client and Security Statement Summary Pages

Summary Pages targeting Accounts, Clients and Security Statements may wish to display different information when summarising a temporary object created from an Account Application.

The built-in Summary Pages have several tweaks when summarising temporary objects.

Summary Page Scripts can identify whether the object they are summarising is a temporary, Account Application-generated object by checking the following properties:

- `finAccount`
  - `AccountApp`
    - This will be `Nothing` when dealing with a normal Account but is a pointer to the Account Application from which it was generated for temporary Accounts.
  - `AccountAppCalculation`
    - This will be `Nothing` when dealing with a normal Account but is a pointer to the Account Application Calculation from which it was generated for temporary Accounts.
- `finClient`
  - `AccountAppApplicant`
    - This will be `Nothing` when dealing with a normal Client but is a pointer to the Account Application Applicant from which it was generated (or updated) for temporary Clients.
  - `AccountAppApplicantPk`
    - The primary key of the Applicant or zero for a normal Client.
- `finSecurityStmt`
  - `AccountApp`
    - This will be `Nothing` when dealing with a normal Security Statement but is a pointer to the Account Application from which it was generated for temporary Security Statements.
- `finSecurityStmtItemBase`
  - `AccountAppCollateralItem`
    - This will be `Nothing` when dealing with a normal Security Statement Item but is a pointer to the Account Application Collateral Item from which it was generated for temporary Security Statements.

Page Sets

Account Applications have been developed from the ground-up to use Page Sets for their main data-entry mechanism.

The **finPOWER Connect 2 Page Sets** document should be used in conjunction with this document when creating Page Sets for Account Applications.

The Data Capture page on the Account Application Types form allows several Page Sets to be defined:

- General
  - This is the Page Set from which the initial data capture is performed.
  - This Page Set will generally be configured with a Form Type of 'Wizard'.
- Reviewing
  - An alternative Page Set can be specified for when the Account Application's State has been changed from 'Incomplete', e.g., it is now 'Reviewing'.
  - If unspecified, the 'General' Page Set will be used.
    - If the 'General' Page Set is configured with a Form Type of 'Wizard', this will be overridden to use 'Tabbed Pages' when run from an Account Application that is 'Reviewing'.

Special Page Set Configuration

For a Page Set to be used with Account Applications, it is required to do the following:

- Handle the following parameters in its Initialise method:
  - AccountAppPk
    - This is the primary key of the Account Application that the Page Set should display.
    - This parameter will be zero when adding a new Account Application.
  - AccountAppTypeId
    - The Id of Account Application Type when adding a new Account Application.
    - The Page Set should create the Account Application object and initialise it using this parameter.
- Validation of an existing Account Application. Check that, if editing an existing Account Application, that the Page Set can handle this, e.g.,:
  - Check it has the correct Applicants, e.g., if the Page Set requires both a 'Main' and 'Joint' Applicant, ensure that the Account Application record contains these.
- Decide which Pages should be visible if the Page Set Form Type is 'Tabbed Pages'.
  - When an Account Application's State is not 'Incomplete', i.e., data entry has been complete, the Form Type will have been changed to 'Tabbed Pages'.
  - The Initialise method should hide any pages that are not applicable, e.g., those that would be skipped if the Page Set was still acting as a 'Wizard'.
    - A good example is a Page Set that contains a page for Individual type Clients and Organisation type Clients. In 'Wizard' mode, the Individual Page would only be displayed if an Individual type Client (or Applicant) is being entered and vice-versa.
      - When displayed as 'Tabbed Pages', the Page Set Script's Initialise method should hide the page that does not apply, e.g., `psh.Pages("Individual").Visible = False`.
        - A special `psh.SetVisibilyFromWizardMove()` method can achieve this automatically by 'faking' movement through the wizard.
- Respect the Page Set's Read-Only state:
  - If the Page Set is being displayed read-only (`psh.ReadOnly = True`), the Page Set should not attempt to update the Account Application in any way and any Page Objects such as buttons or links to pre-load Client information should be disabled.

NOTE: When adding template Script code for a Page Set, the option is given to add 'Account Application' template code. This template code gives an example of how to handle the special parameters required for Account Applications.

Loading and Validating an Existing Account Application and Initialising a New Record

The following code example demonstrates how the `Initialise()` method of the Page Set should cater for loading and validating an existing Account Application and also initialising a new Account Application:

```
' Objects
Private mAccountApp As finAccountApp
Private mApplicant As finAccountAppApplicant

Public Overrides Function Initialise() As Boolean

  Dim AccountAppPk As Integer
  Dim AccountAppTypeId As String

  ' Assume Success
  Initialise = True

  ' Initialise
  mReports = DirectCast(psh.Reports, ISfinReports)
  mUI = DirectCast(psh.UserInterface, ISUserInterfaceBL)

  ' Create Objects
  mAccountApp = finBL.CreateAccountApp()

  ' Get Parameters (default Account Application Type for new records matches the Page Set Id)
  AccountAppPk = psh.Parameters.GetInteger("accountAppPk")
  AccountAppTypeId = psh.Parameters.GetString("accountAppTypeId", psh.PageSetId)

  ' Load Record
  Initialise = Record_Load(AccountAppPk, AccountAppTypeId)

  ' Validate (i.e., ensure this Application can be edited from this wizard)
  If Initialise Then
    If mAccountApp.Applicants.Exists("MAIN") Then
      mApplicant = mAccountApp.Applicants.ItemByKey("MAIN")
    Else
      Initialise = False
      finBL.Error.ErrorBegin("Application does not have a 'MAIN' Applicant and so cannot be
accessed via this wizard.")
    End If
  End If

  ' Decide which pages to show if not viewing as wizard
  If psh.NavigationMethod <> isefinPageSetNavigationMethod.Wizard Then
    ' Set visibility automatically (could also manually decide)
    psh.Pages.SetVisibilyFromWizardMove()
  End If

  ' Load Fields
  If Initialise Then
    Fields_Load()
  End If

End Function

Private Function Record_Load(pk As Integer,
                             accountAppTypeId As String) As Boolean

  ' Assume Success
  Record_Load = True

  If pk = 0 Then
    ' New Account Application
    mAccountApp.Clear()

    ' Initialise
    Record_Load = mAccountApp.Initialise(accountAppTypeId)
  Else
    ' Load Existing
    Record_Load = mAccountApp.LoadPk(pk)
  End If

End Function

Private Sub Fields_Load()

  ' Update Page Object values from mAccountApp and mApplicant
```

```
End Sub
```

NOTE: The above example assumes that the Page Set Id matches the Account Application Id if (for some reason) the AccountAppTypeId parameter is not passed to the Page Set, e.g., the Page Set has been run outside of the Account Applications form (useful for testing or for adding an Application Shortcut to run the Page Set).

Respecting the Page Set's Read-Only State

Typically, a Page Set is in read-only mode when viewing an Account Application from the Data Capture page of the Account Applications form.

Generally, all buttons and other Page Objects will be set to be read-only or disabled in this situation however, there may be certain other parts of the Page Set that you may wish to hide or disable if the Page Set is read-only. Some examples are:

- Hyperlinks in an HTML Panel.
- Grid drilldown buttons.

Showing Special Forms

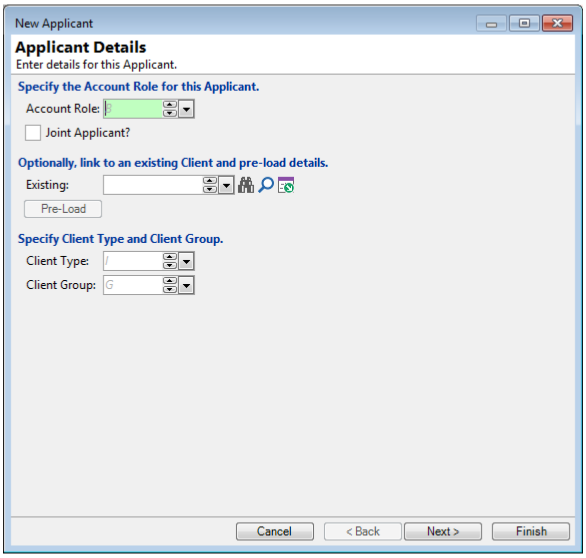
Special forms are displayed from a Page Set using one of the Page Set Handler's 'FormShow' methods.

This section details only those special forms related directly to Account Applications.

NOTE: Special forms are detailed fully in the **finPOWER Connect 2 Page Sets** document, Advanced Scripting, Showing Special Forms section.

Account Application Applicant

The `FormShowAccountAppApplicant` method displays the Applicant wizard to either edit an existing or add a new Applicant, e.g.:

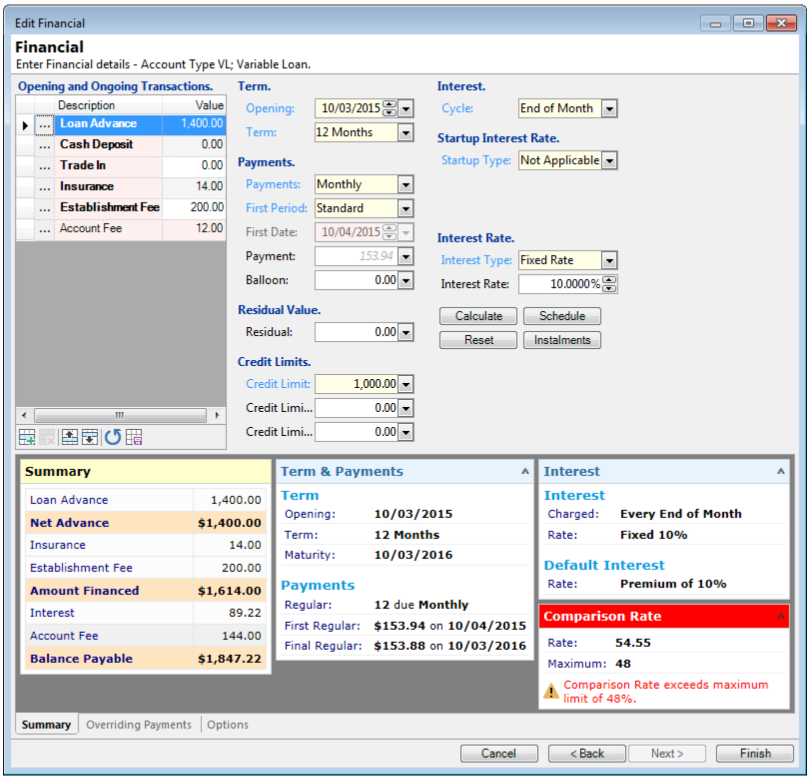


When this form is closed, the Page Set receives the 'AccountAppApplicantFormClosed' event which can then be used to update the Page Set as per the following code example:

```
Public Sub PageSet_AccountAppApplicantFormClosed(sender As Object,
                                                  e As
finPageSetHandlerAccountAppApplicantFormClosedEventArgs) Handles Me.AccountAppApplicantFormClosed

  ' Refresh Applicants Grid and select edited Applicant
  If Not e.Cancelled Then
    gridApplicants.VirtualDataRefresh(mAccountApp.Applicants.IndexOf(e.AccountAppApplicant))
  End If

End Sub
```

Account Financial

The `FormShowAccountFinacial` method displays the Account Financial form, e.g.:

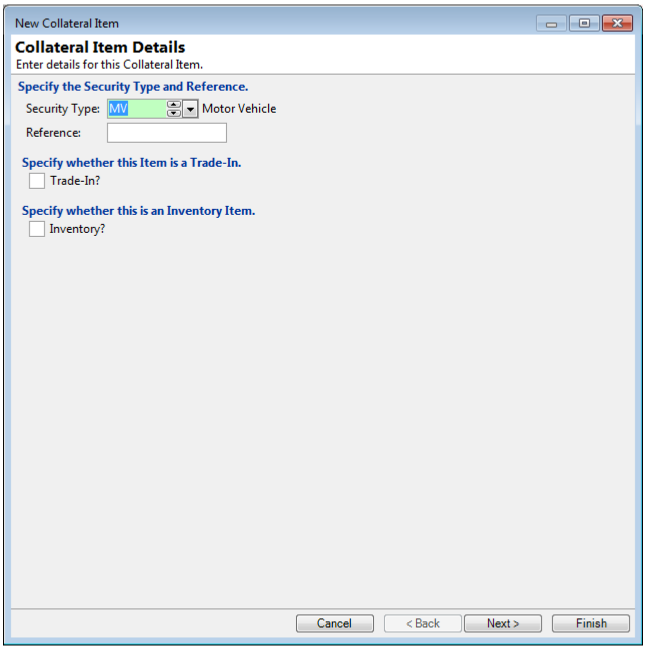


When this form is closed, the Page Set receives the 'AccountFinancialFormClosed' event which can then be used to update the Account Application Calculation (Quote) as per the following code example:

```
Public Sub PageSet_AccountFinancialFormClosed(sender As Object,
                                               e As
finPageSetHandlerAccountFinancialFormClosedEventArgs) Handles Me.AccountFinancialFormClosed

  ' Update Calculation
  If Not e.Cancelled Then
    ' Update Calculation
    mAccountApp.Calculation.UpdateFromAccount(e.Account)

    ' Load Fields
    Fields_Load()
  End If

End Sub
```

Collateral Item Wizard

The `FormShowAccountAppCollateralItem` method displays the Collateral Item wizard to either edit an existing or add a new Collateral Item, e.g.:



When this form is closed, the Page Set receives the 'AccountAppCollateralItemFormClosed' event which can then be used to update the Page Set as per the following code example:

```
Public Sub PageSet_AccountAppCollateralItemFormClosed(sender As Object,
                                                      e As
finPageSetHandlerAccountAppCollateralItemFormClosedEventArgs) Handles
Me.AccountAppCollateralItemFormClosed

  ' Refresh Collateral Items Grid and select edited Collateral Item
  If Not e.Cancelled Then

gridCollateralItems.VirtualDataRefresh(mAccountApp.CollateralItems.IndexOf(e.AccountAppCollateralIt
em))
  End If

End Sub
```

HTML Widgets

Account Applications can use HTML Widgets as a data-entry mechanism.

This has the benefit of allowing Applications to be entered or edited either within finPOWER Connect Desktop or finPOWER Connect Cloud versions.

The **finPOWER Connect 3 HTML Widgets** document should be used in conjunction with this document when creating HTML Widgets for Account Applications.

The Data Capture page on the Account Application Types form allows several HTML Widgets to be defined:

- General
  - This is the HTML Widget from which the initial data capture is performed.
- Reviewing
  - An alternative HTML Widget can be specified for when the Account Application's State has been changed from 'Incomplete', e.g., it is now 'Reviewing'.
  - If unspecified, the 'General' HTML Widget will be used.

Special HTML Widget Configuration

For an HTML Widget to be used with Account Applications, it is required to do the following:

- Handle the following parameters:
    - AccountAppPk
        - This is the primary key of the Account Application that the HTML Widget should display.
        - This parameter will be zero when adding a new Account Application.
    - AccountAppTypeId
        - The Id of Account Application Type when adding a new Account Application.
        - The HTML Widget should create the Account Application object and initialise it using this parameter.
- Validation of an existing Account Application. Check that, if editing an existing Account Application, that the HTML Widget can handle this, e.g.,:
    - Check it has the correct Applicants, e.g., if the HTML Widget requires both a 'Main' and 'Joint' Applicant, ensure that the Account Application record contains these.

Appendix A – Guidelines

Script Responsibilities

Responsibilities of the Account Application Type Script include:

- Adding default Applicants, e.g., a Main and a Joint Application.
  - Also, key-locking these if they are required elsewhere in the Script or by another Script such as a Page Set Script.
  - And, setting any defaults such as whether the Applicant is used and whether the Postal and Physical address should, by default, be the same.
- Adding default Collateral Items, e.g., a Motor Vehicle.
  - Also, key-locking these if they are required elsewhere in the Script or by another Script such as a Page Set Script.
  - And, setting any defaults such as whether the Collateral Item represents a 'Trade-In'.
  - **NOTE:** Adding default Collateral Items may not be as common as adding default Applicants since you may wish Users to enter any Collateral directly from the Account Applications form.
- Adding default Calculations (Quotes) if the Account Application should give the applicant multiple Quotes to choose from.
  - Also, key-locking these if they are required elsewhere in the Script or by another Script such as a Page Set Script.
  - **NOTE:** A default Calculation exists on the Account Application when it is first accessed so any Script code may wish to clear the `Calculations` collection before adding items.

A Page Set Script may do the following:

- Initialise a new Account Application, thereby automatically creating default Applicants and setting defaults (providing the Account Application Type has an 'InitialiseAccountApp' event that does this).
- Add additional Applicants or Collateral Items but ONLY IF THIS MAKES SENSE, e.g.:
  - It contains a grid of Applicants thereby allowing the User to add additional Applicants.

Appendix B – Collateral Items vs Security Items

Account Applications have built-in functionality to convert Collateral Items to Security Statement Items.

This is achieved as follows:

- The Account Application Type Script's 'UpdateSecurityStmt' event is called.
- A new Security Item is created and a call made to `finSecurityStmtItemBase.UpdateFromAccountAppCollateralItem` to update it from the Collateral Item.

To supplement this functionality, e.g., updated User Data on the Security Items, the following Script code can be used:

```
Public Overrides Function UpdateSecurityStmt(accountApp As finAccountApp,
                                             account As finAccount,
                                             securityStmt As finSecurityStmt) As Boolean

  ' Call Base Method to update the Security Statement which includes creating an item for
  ' each Collateral item providing it Is flagged As 'Used' and has a SecurityTypeId set
  UpdateSecurityStmt = MyBase.UpdateSecurityStmt(accountApp, account, securityStmt)

  ' Set other properties, e.g., UserData to record additional information from Application
  ' NOTE: Any Security Items created above will have their AccountAppCollateralItem property
  '       Set To the Collateral item from which they were created

End Function
```

Whereas Security Statement Items have different classes to represent different type of Security Statement Item, e.g., `finSecurityStmtItemMotorVehicle`, and each of the classes contains type-specific properties such as `RegistrationNumber`, Collateral Items have a super-set of properties of which only certain ones are used for certain Security Classes when updating the Security Statement Item from the Collateral Item.

The following table shows which Collateral Item properties are used by which Security Classes.

WARNING: Certain Collateral Item properties have different names but refer to the same underlying property, e.g., HIN and VIN.

These are documented in the business layer help for `finAccountAppCollateralItem` and also in the table below.

A - Aircraft

B - Boat

IN - Insurance

IV - Investment

L - Land

MV - Motor Vehicle

SN - Serial Numbered Good

S - Standard

Collateral Item Property	A	B	IN	IV	L	MV	SN	S	
Colour		X				X	X		
CountryOfOrigin		X				X	X		

Currency				X					
Description	auto	auto				auto	auto	X	
DescriptionFull							X	X	
DescriptionGoods							X		
DP					X				
EngineMake		X							
EngineNumber		X				X			
Engine2Make		X							
Engine2Number		X							
Engine3Make		X							
Engine3Number		X							
Engine4Make		X							
Engine4Number		X							
FuelType						X			
HullLength		X							
HullType		X							
InvesmentCode				X					
LandUse					X				
LandZone					X				
Lot					X				
Make						X	X		
MarketValue				X					
Model						X	X		
ModelSub						X	X		
MortgageDischargedDate					X				
MortgageRegistrationDate					X				
MortgageRegistered					X				
MortgageType					X				
NewUsed		X				X	X		
Notes				X					
Odometer						X			
PlateType						X			

Quantity				X					
SerialNumber IdentificationNumber LandTitle ManufacturerSerialNumber		X			X	X	X		
UnitPrice				X					
VehicleType BoatType		X				X			
VIN HIN		X				X			
Year		X				X	X		
<u>Insurance</u>									
InsuranceCompany	X	X	X	X	X	X	X	X	
InsuranceNotes	X	X	X	X	X	X	X	X	
InsurancePolicy	X	X	X	X	X	X	X	X	
InsurancePremium	X	X	X	X	X	X	X	X	
InsuranceRenewalDate	X	X	X	X	X	X	X	X	
InsuranceSumInsured	X	X	X	X	X	X	X	X	
<u>Valuation</u>									
SecurityValue		X			X	X	X	X	
Valuation1		X			X	X	X	X	
Valuation2 ValuationLandValue					X				
Valuation3 ValuationImprovementsValue					X				
Valuation4 ValuationOtherValue					X				
ValuationDate		X		X	X	X	X	X	
ValuationNotes		X			X	X	X	X	
ValuationReviewDate		X		X	X	X	X	X	
ValuationValuer		X			X	X	X	X	
ValuationPercentUsed		X			X	X	X	X	
ValuationPriorRanking					X				
ValuationPriorRanking2					X				
ValuationPriorRanking3					X				
ValuationPriorRanking4					X				

ValuationPriorRankingNotes					X				
ValuationPriorRankingNotes2					X				
ValuationPriorRankingNotes3					X				
ValuationPriorRankingNotes4					X				
ValueResidual		X			X	X	X	X	
ValueResidualPercent		X			X	X	X	X	