APTTUS

X-Author for Excel August 2015 Designer and Runtime Guide

9 September, 2015

Table of Contents

Overview

Welcome to X-Author for Excel. Apttus is delighted to provide you with this simple-to-use but extremely powerful tool for using Excel with Salesforce. X-Author for Excel addresses two key areas for using Excel and Salesforce together.

For Salesforce applications like CRM, Apttus CPQ or other custom applications. Here, X-Author lets you use Microsoft Excel natively as an amazing user interface (UI) for tasks that really need Excel rather than a browser UI. You can use a new or existing spreadsheet to connect to Salesforce, or you can launch a spreadsheet directly from within Salesforce. X-Author for Excel can accommodate many formats (not just grids), simultaneously use multiple objects, and access data that is spread across multiple worksheets. The query capabilities are powerful, resulting in remarkable efficiency gains and increased Salesforce adoption and usage.

For traditional stand-alone spreadsheets. In many companies, there are hundreds of thousands of spreadsheets. In these companies, people are routing spreadsheets through email and shared drives. This means you don't always know who has the latest version, worksheets are being cut and paste into other spreadsheets, it takes a lot of time to get properly consolidated reporting, and integrity is often questionable. For these use cases, consider having some or all the data in these spreadsheets housed in Salesforce. With X-Author, you can retain the Microsoft Excel UI and formats, but the data is instantly available in the Salesforce cloud for easy enterprise control, reporting, and routing. There are also no version control issues because X-Author centralizes the Microsoft Excel templates in Salesforce, and you can create and enforce security and controls over the data even though it's being used in Microsoft Excel.

In both cases, you can use just the Microsoft Excel UI and the data will move between Excel and Salesforce. Or, if you still need the actual Microsoft Excel file for audit purposes—or perhaps because it contains more data than what you might choose to store in Salesforce—X-Author provides file check-in and check-out capabilities to provide centralized control.

To become proficient with X-Author for Excel as quickly as possible, we suggest you work through the documentation in the order in which it is presented and also watch the video tutorials. The documentation begins with core terms and concepts in the Product Orientation section, and then you immediately build your first two applications. Once you have the basic hands-on experience, each product component is described in more detail. Finally, to round out your training, there are more advanced application samples in the X-Author for Excel Sample Apps Guide. Many of the examples refer to the Salesforce Account and Opportunity objects. If these are not available to you, please substitute any other objects.

See what's new in this release!

Release Notes for X-Author for Excel August 2015

How to get Information from the Release Notes

The release notes provide you packages required to upgrade and high-level descriptions of new features and enhancements to existing features. Every new feature and enhancements section has a Get Started section to help you setup and use the new feature or enhancement. The release notes also contains a Resolved Issues and Known Issues section.

Links in this Document

To access the content provided by external links in this document you must be logged into the Apttus Community Portal. If you encounter a Salesforce login page, enter your Apttus Community login and password. For assistance with your Community credentials, please contact Apttus Support.

Patch Release Notes

To review release notes for X-Author for Excel August 2015 patch releases, refer to X-Author for Excel August 2015 Patch Release Notes.

Packages

The following packages and dependent packages are required to upgrade to this release. The packages marked as **(New)** are newer packages from the previous releases and must be installed for this version.

Product	Version

Product	Version
Apttus X-Author™ For Excel **(New)**	4.0.0040 | 4.4
Apttus X-Author™ Designer For Excel	1.0.0007 | 1.7
Apttus X-Author™ for Excel Add-in: Runtime Only **(New)**	3.8.0922 | 3.8
Apttus X-Author™ Designer + Runtime Add-in **(New)**	3.8.0922 | 3.8

Feature Summary

X-Author for Excel August 2015 contains the following new features.

- Quick App Enhancements
- Add Rows and Columns to Matrix Apps in Runtime Mode
- Add Row and Paste Actions
- Preview apps from X-Author Designer
- Two new supported field types

Quick App Enhancements

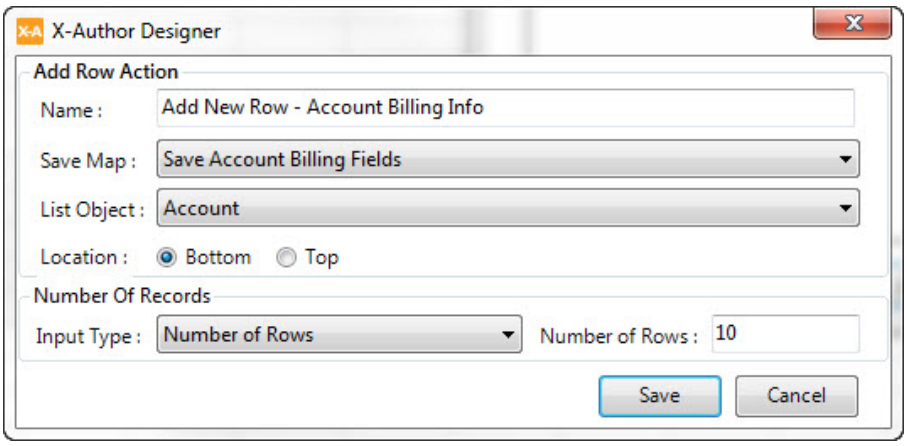
Quick Apps are now available as a standalone **Starter Edition of X-Author for Excel**. This simplified version of X-Author for Excel allows app designers (like operations and IT staff, or Salesforce administrators) to create and modify X-Author for Excel apps using the Quick App wizard (but not the broader X-Author Designer capabilities), and then assign these apps to end users or profiles. The Starter Edition allows your end users to open and use assigned X-Author apps within Excel, where they can retrieve data from Salesforce, modify it and add to it, and then save it to Salesforce.

Add Rows and Columns to Matrix Apps in Runtime Mode

You can now add rows and columns to your matrices when running X-Author for Excel apps. Depending on the app you've designed, adding rows and columns may allow you to create new records in Salesforce by simply adding those records to your spreadsheet and saving it back to Salesforce. To learn more about this feature, refer to Adding Rows and Columns to a Matrix.

Add Row and Paste Actions

Two new actions have been added to this version of X-Author for Excel, **Add Row** and **Paste**. The Add Row action lets you automatically add rows to the top or bottom of your spreadsheet as part of your flow. You can add a row containing data from a specified set of cells or from another action flow step, or you can add a specified number of empty rows and allow users to insert the data.

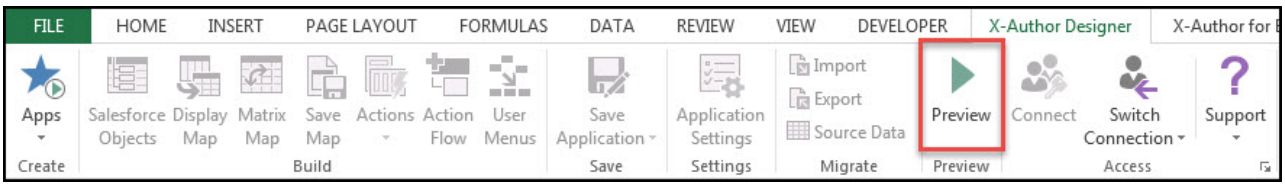


The **Paste action** lets you automatically add content to your workbook.

To learn more, refer to Add Row Action and Paste Action.

Preview apps from X-Author Designer

In this version, you can preview your apps directly from X-Author Designers. By clicking the new Preview button on the ribbon menu, you can open runtime version of your app while you are still in design mode.



For more information, refer to Previewing Apps.

Two new supported field types

In this version, both **email** and **formula** fields are supported for your X-Author for Excel apps. You can now add editable Email fields to your apps and use them in Save Maps. Formula fields are read-only fields, so they cannot be edited or added to Save Maps. (Use of these fields as search filters is not supported.)

X-Author for Excel August 2015 Patch Release Notes

View Release Notes on this Page

- Patch 1
- Patch 2

Download PDFs

If you need a PDF version of the release notes, click a link below.

- Apttus X-Author for Excel August 2015 Patch 1 Release Notes.pdf
- Apttus x-Author for Excel August 2015 Patch 2 Release Notes.pdf

Patch 1

Packages

The following packages and dependent packages are required to upgrade to this release. The packages marked as **(New)** are newer packages from the previous releases and must be installed for this version.

Product	Version

Product	Version
Apttus X-Author™ For Excel **(New)**	4.0.0040 | 4.4
Apttus X-Author™ Designer For Excel	1.0.0007 | 1.7
Apttus X-Author™ for Excel Add-in: Runtime Only **(New)**	3.8.0922 | 3.8
Apttus X-Author™ Designer + Runtime Add-in **(New)**	3.8.0922 | 3.8

Resolved Issues

This patch release resolves the following existing issues in X-Author for Excel.

Case Number	Description	Apttus Internal ID
	You could not add a row with the **Save Other** field even when you selected the **Add Row** check box in the User Menu. This issue is now resolved. You can add a row with **Save Other** field.	AB-1720
	You could not save the App that you created when you configure the **Lookahead** feature on a Save field without specifying a Return field. This issue is now resolved. You can successfully save the App that you create when you configure the **Lookahead** feature on a Save field even without specifying a Return field.	AB-1722

Patch 2

Packages

The following packages and dependent packages are required to upgrade to this release. The packages marked as **(New)** are newer packages from the previous releases and must be installed for this version.

Product	Version
Apttus X-Author™ For Excel **(New)**	4.0.0040 | 4.4
Apttus X-Author™ Designer For Excel	1.0.0007 | 1.7
Apttus X-Author™ for Excel Add-in: Runtime Only **(New)**	3.8.0930 | 3.8
Apttus X-Author™ Designer + Runtime Add-in **(New)**	3.8.0930 | 3.8

Resolved Issues

This patch release resolves the following existing issue in X-Author for Excel.

Case Number	Description	Apttus Internal ID
	You could not create **Presto X-Author** app if one of the filters is using **In** operator with input data. This issue is now resolved. You can create a **Presto X-Author** app even if you use **In** operator with input data in any of the filters.	AB-1721

Installing X-Author for Excel August 2015

The following managed packages must be installed in the following order.

1. Apttus X-Author for Excel (Apttus_XApps / Runtime)
2. Apttus X-Author Designer Apttus_XAppsDS / Designer)

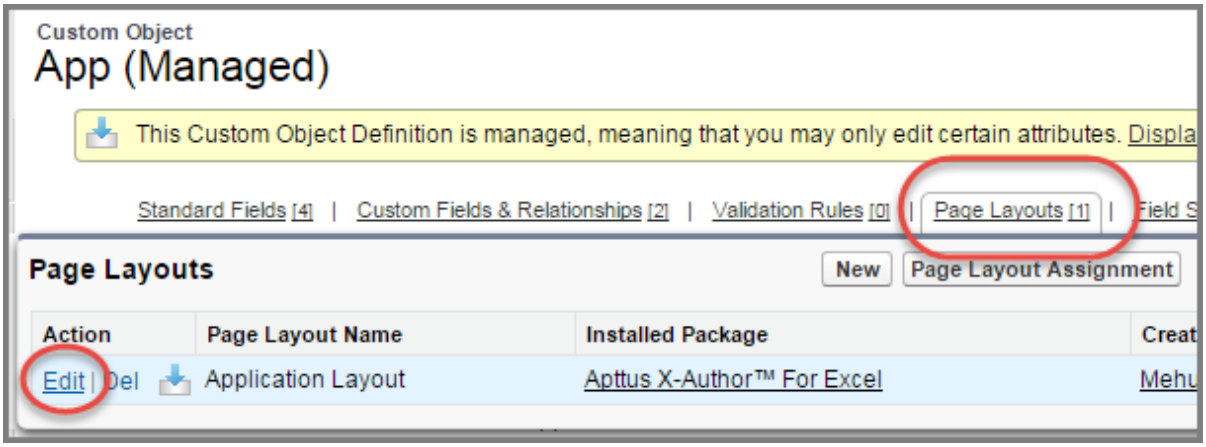
If you are going to design apps, you are required to have both X-Author Designer and X-Author for Excel licenses. If you are only going to use apps, you need only an X-Author for Excel license. Navigate to the relevant installed package in Salesforce Setup and add users accordingly. For more help, refer to Assigning Licenses for Installed Packages on Salesforce's website.

Configuring Salesforce Layouts

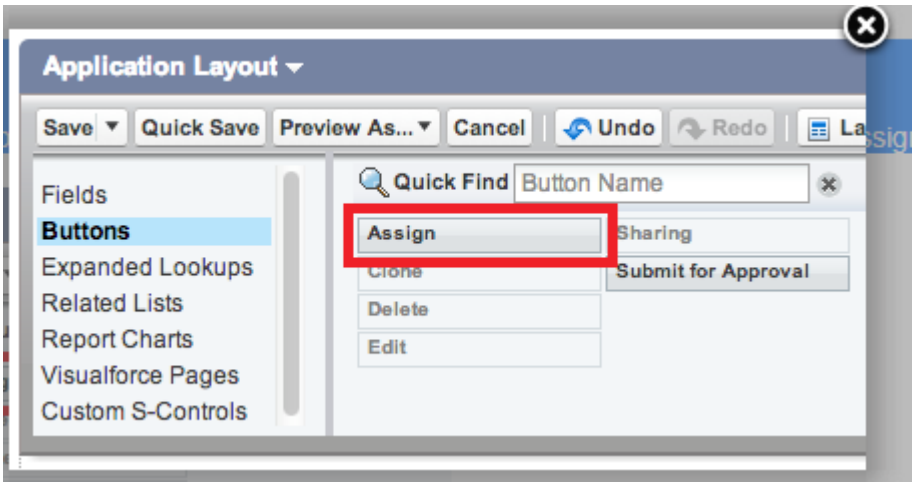
Designers must have two buttons added to their layouts before they can use X-Author Designer, the Assign button and the Excel Setup button.

To add the Assign button

1. Go to Setup > App Setup > Create > Objects.
2. Click the **App** object.
3. Hover over the **Page Layouts** button (located at the top of the page) and click **Edit** next to Application Layout.



4. In the left panel, click **Buttons**.
5. If the Assign button isn't in the displayed as shown in the following figure, you will need to add it for designers.



Designers must have the Edit in Excel Setup button added to their layout on the Apps List view in order to use the Launch from Salesforce feature.

To add the Edit in Excel Setup button

1. Go to Salesforce Setup > Create > Objects and click App.
2. In the Search Layouts section, click Edit next to *Apps List View* and move the *Edit in Excel Setup* to *Selected Buttons*.

Search Layouts

Action	Layout	Columns Displayed	Buttons Displayed
Edit	Search Results	Application Name	
Edit	Lookup Dialogs	Application Name	N/A
Edit	Lookup Phone Dialogs	Application Name	N/A
Edit	Apps Tab	Application Name	N/A
Edit	Apps List View	N/A	New,Accept,Change Owner,Edit In Excel Setup
Edit	Search Filter Fields		N/A

3. Click Save.

Installing the X-Author Add-ins

There are two executable files:

- X-Author Designer, which comes bundled with X-Author for Excel
- X-Author for Excel, which is to be installed for end users who will not be designers

The installation process creates a folder on your computer that contains the files required to use the X-Author Designer and the X-Author for Excel add-ins. Depending on the version of Windows you are using, the folder is created either in your Program Files or Program Files (x86) directory.

Installation Components

Your computer must meet the minimum requirements listed below to use X-Author Designer or X-Author for Excel.

> The Mac OS X operating system is not supported; however X-Author can be used on a Mac that is running a Windows Virtual Machine in a VMWare or Parallels type environment.

Component	Requirement
Microsoft	Microsoft Excel 2007 (32 bit & 64 bit), Microsoft Excel 2010 (32 bit & 64 bit), Microsoft Excel 2013 (32 bit & 64 bit)
Microsoft	Microsoft .NET Framework 4.5
Microsoft	Microsoft Visual Studio 2010 Tools for Office Runtime
Microsoft	Microsoft Office 2010 Primary Interop Assemblies
Salesforce	Any Salesforce edition that provides API support for the Salesforce objects that will be accessed in an App.

To Install X-Author Designer or X-Author for Excel

Before you install X-Author Designer:

- You must be connected to the Internet.
- You must have access to the X-Author for Excel or X-Author Designer executable file.
- You must have Administrator permissions on the computer being used for the install.
- You must close all instances of Microsoft Excel running.

1. Double-click the installer executable to display the InstallShield Wizard. In rare instances, a restart of Windows may be required. Click Yes to confirm the restart and, once you log in to Windows, the installer should resume automatically.
2. Click Next to start the wizard.
3. On the License Agreement screen, select I accept. and click Next. If you do not accept the terms of the agreement, X-Author will not be installed on your computer.
4. Click Install to begin the installation process. If any components need to be installed for X-Author to work as expected, they will be automatically detected as missing and installed. If components are required but cannot be accessed through the Internet, an appropriate message will be displayed and the install will need to be completed after all missing components have been manually installed.
5. Click Finish to close the wizard and complete the installation.
6. Open Microsoft Excel.

The X-Author Designer and/or X-Author for Excel ribbon tab should be available in Excel.If the ribbon tab is not displayed, you may need to register X-Author Designer or X-Author for Excel with Excel.

Enabling Proxy Authentication

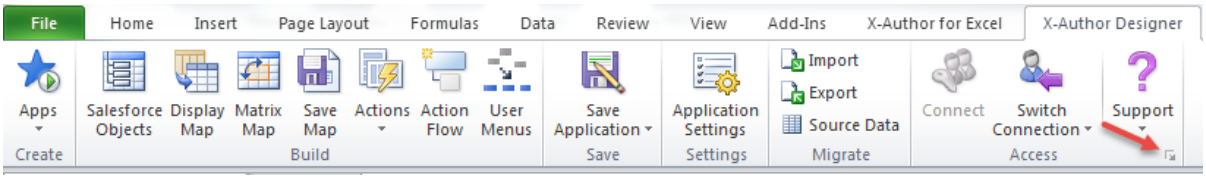
By default, X-Author for Excel will use your Microsoft Internet Explorer (IE) proxy settings. Since the IE settings are used to connect to your Salesforce org, the default configuration should work for X-Author in most cases. However, if your work environment is behind a firewall and you need manually to configure proxy authentication to connect to Salesforce, you need to complete the procedure in this section.

Microsoft Internet Explorer Proxy Settings

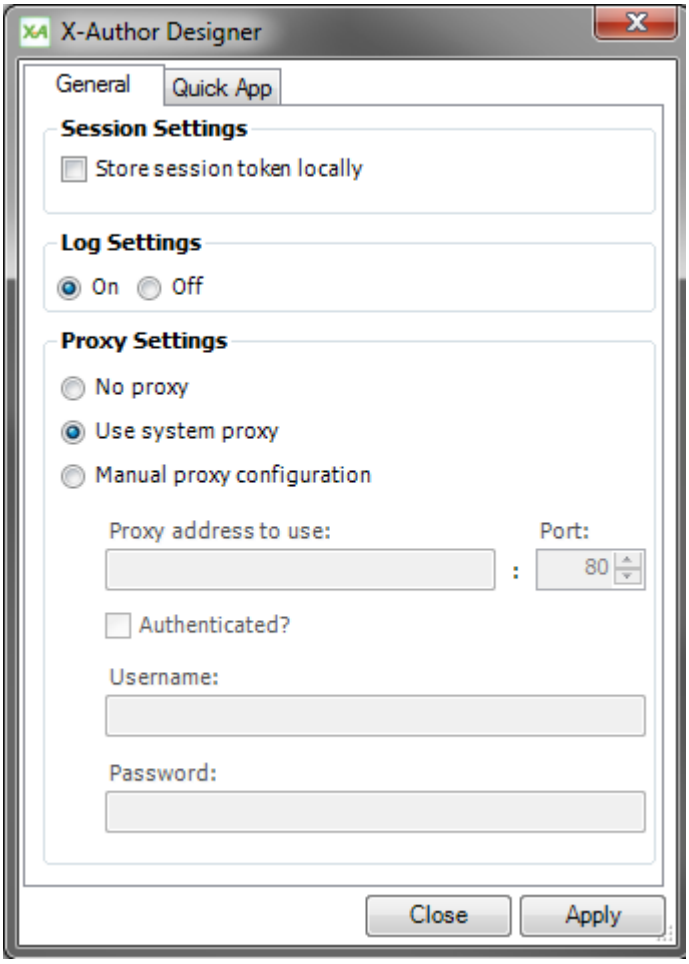
Proxy server settings in IE are located at Tools > Internet Options > Connections > LAN Settings. However, the location of the proxy server settings in IE may be different due to your IE version. Refer to Microsoft IE's documentation for more information.

To enable X-Author for proxy authentication

1. Open Excel and click either the X-Author Designer or X-Author for Excel tab.



2. In the lower right corner of the Access section, click . The X-Author Designer dialog opens.



3. On the General tab, click one of the following radio buttons.

Setting	Description
No proxy	Click this radio button if you do not use a proxy connection.
Use system proxy	Click this radio button to use the Microsoft Internet Options setting to connect to determine your proxy server settings. If you choose this option, X-Author will automatically resolve the connection so that you can successfully login to X-Author and connect to your Salesforce org. This is the default setting.
Manual proxy configuration	Click this radio button to enter the settings for the proxy server you will be using. If you choose this option, you must enter the address and port number of your proxy server int he Proxy Address to Use and Port fields, respectively. If you are required to provide authentication information, click the Authenticated check box and enter a user name and password in the provided fields.

4. Click **Apply** and **Close**.

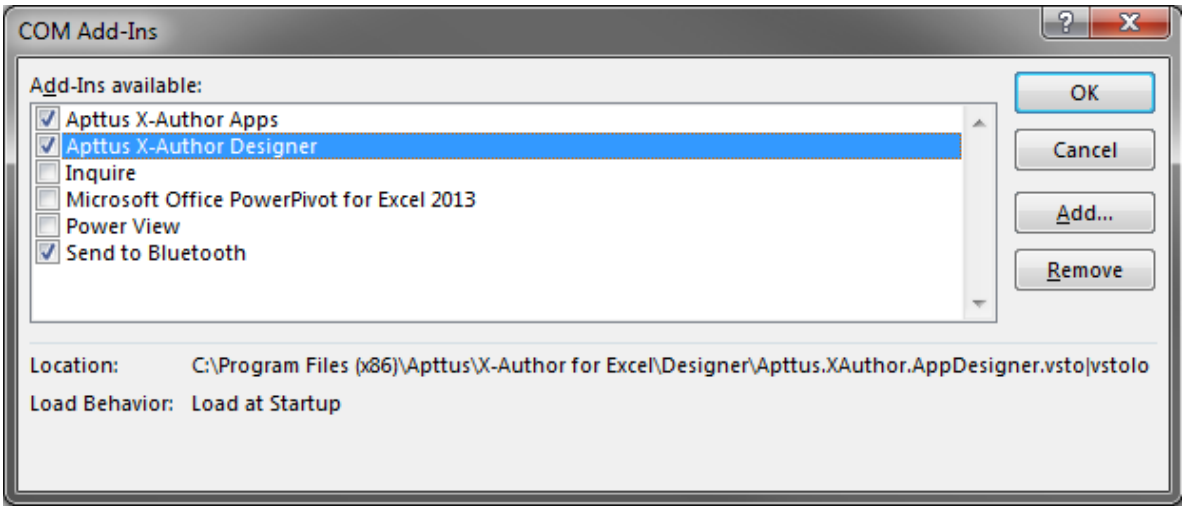
Your system proxy settings are now associated with X-Author Designer and will take effect the next time you login.

Registering X-Author Designer in Microsoft Excel

In rare cases, the X-Author Designer or X-Author for Excel tabs are not properly installed and, therefore, are not visible in your Microsoft Excel ribbon menu. If this occurs, you must manually register the X-Author add-in to add it to the Excel ribbon.

To manually register the Add-Ins

1. Start Excel and select File > Options to display the Excel Options dialog.
2. Click Add-Ins, select COM Add-ins from the Manage list, and click Go.
3. Click the Apttus X-Author Designer and X-Author for Excel check boxes and click OK.



The add-ins will now be available in the Excel ribbon; however, you may need to restart Excel for it to display as expected.

Setting Salesforce Permissions

After you install X-Author, you must provide users with the proper permissions for accessing the managed packages. Additionally, you must set field-level permissions on the App object that is created during the installation of the managed package.

Assigning User Profile Permissions

You must assign the following permissions in Salesforce to the user profiles of users who will use X-Author for Excel. Unless they have a System Administrator profile, users will not automatically have access to the package contents that are required to access X-Author functionality. The following permissions must be configured.

Runtime Users	
Custom App Settings	Apttus X-Author for Excel: Visible
Tab Settings	X-Author for Excel: Default on

Runtime Users	
Custom object permissions	• Apps: Read • App Assignments: Read • App Files: Read / Create / Edit / Delete
Enabled Apex Class Access	Apttus_XApps.AppBuilderWS
Enabled Visualforce Page Access	• Apttus_XApps.SearchSelectAction • Apttus_XApps.Landing

Designer Users	
Custom App Settings	Apttus X-Author for Excel: Visible
Tab Settings	• X-Author for Excel: Default on • X-Author Designer: Default on • Apps: Default on • App Licenses: Default on
Custom object permissions	• Apps: Read / Write • App Assignments: Read / Write • App Files: Read / Create / Edit / Delete
Enabled Apex Class Access	Apttus_XApps.AppBuilderWS

Designer Users	
Enabled Visualforce Page Access	• Apttus_XApps.SearchSelectAction • Apttus_XApps.Landing • Apttus_XApps.LicenseManager • Apttus_XAppsDS.DesignerInfo • Apttus_XAppsDS. AppAssignmentOverview • Apttus_XAppsDS.AppAssignmentUsers • Apttus_XAppsDS.AppAssignmentProfiles

Assigning Field-Level Permissions

The “Activated” and “UniqueId” fields on the App object must be set to visible. Usually, these permissions will be set during the installation of the managed package. However, depending on org-specific security settings, you might need to set these permissions manually.

To set the field-level permissions

1. In Salesforce App Setup, go to Create > Objects > App.
2. Click Custom Fields and Relationships.
3. Click Activated.
4. Click Set Field Level Security.
5. Make sure that the Visible check box is selected for every profile that requires app access.
6. Repeat steps 3 through 5 for the UniqueId custom field definition.
7. Click Save.

Getting Started

This section contains information to help you get started with X-Author for Excel.

What's in the box?

X-Author for Excel is comprised of the following components.

Two Salesforce Managed Packages (**always required**):

- Apttus X-Author for Excel (Apttus_XApps / Runtime)
- Apttus X-Author Designer (Apttus_XAppsDS / Designer)

Two Salesforce Managed Packages for optional modules:

- Apttus X-Author for Excel Admin
- Apttus X-Author for Excel Presto

Two executable files installed as Excel add-ins:

- X-Author Designer: used for creating Apps
- X-Author for Excel: used for running Apps created with X-Author Designer

Follow the installation instructions to get the packages and add-ins installed and to get connected to your Salesforce org.

> Designers will see both the X-Author Designer and X-Author for Excel tabs in the Excel menu. End users will see only X-Author for Excel.

What is an App?

An X-Author for Excel application, or "app," is created within Microsoft Excel using X-Author Designer. There is no coding because all configuration is accomplished through the point-and-click designer.

An app is the combination of:

- A Microsoft Excel template that contains the Excel workbook layout.
- An XML file that contains the App-specific user menus, action flows, action maps, and other user-experience elements.

Apps are stored in the Applications object within a Salesforce org. Each app is represented by one application record, the Microsoft Excel spreadsheet template, an XML file, and other configuration data. (These should not be edited without using X-Author or it may corrupt the App.)

X-Author for Excel apps are not related to the Salesforce definition of an application or app. A single X-Author app can include any number of objects that span any number of Salesforce applications.

What licenses are required?

App creation requires licenses from both Apttus and Salesforce.

From Apttus

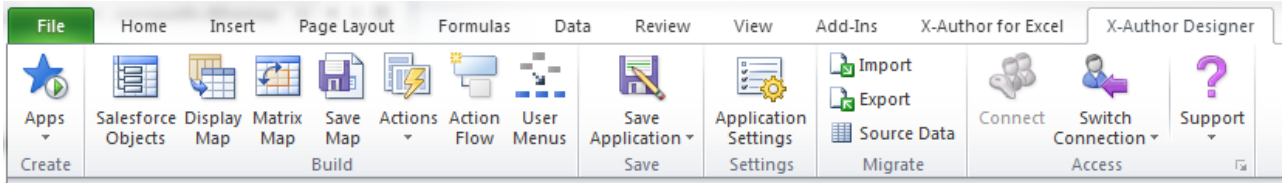
- X-Author Designer—This license is required for app designers.
- X-Author for Excel—This license is required for runtime users who can run any app that has been both activated and assigned to them by a designer.
- X-Author for Chatter—This license is required to use the embedded X-Author for Chatter functionality.

From Salesforce

- To access Salesforce.com from Microsoft Excel using any of the above X-Author licenses, you must have API-enabled Salesforce.com licenses.
- To use X-Author for Chatter, Salesforce Chatter licenses with API access are required.

X-Author Designer's Components

The X-Author Designer ribbon in Microsoft Excel contains all of the X-Author for Excel components.



The following table describes each of the components.

Apps	Click this button to create a new app or open an existing one.
Salesforce Objects	Click this button to select Salesforce objects, fields, and behaviors for an app.
Display Map	Click this button to define the associations between Excel spreadsheet cells and Salesforce fields for the Salesforce data you want to retrieve. When data is retrieved from Salesforce, using one of the X-Author data-getter actions, it is temporarily stored in Excel memory before being placed into the worksheet. X-Author then uses a Display Map to place the data in the correct Excel cells.
Matrix Map	Click this button to define a matrix maps. Matrix maps extend the capabilities of X-Author for Excel well beyond list formats. In lists, one Salesforce record is represented per Excel row or column. Using matrices, a record can be represented by one or more cells by mapping columns and rows (the X and Y axes) of a matrix as well as the data contained in the cell at the intersection of the mapped column and row. Depending on the type of matrix, various data cells can be updated by a user. If no value exists in a cell, adding a value can create a new record depending on the matrix design.

Save Map	Click this button to define the associations between Excel spreadsheet cells and Salesforce fields for Salesforce data you want to save. Save maps define the relationship between cells in your Excel spreadheet and fields in the Salesforce database for the data that will be saved to Salesforce. Updates, inserts, upserts, and deletes are all supported.
Actions	Click this button to create a variety of user experience actions that query, save, display, and act on the data in the workbook and Salesforce. Actions retrieve, display, or save data or the file itself. They can also automate changes within the worksheet as a macro execution or by clearing data from the cells. Actions can be used interchangeably in action flows.
Action Flow	Click this button to define a set of actions that execute sequentially when an end user clicks on a user menu button during runtime.
User Menus	Click this button to create menu groups and buttons in your app's Excel ribbon menu. These UI elements guide the end user in using the app and initiate the interaction between Microsoft Excel and Salesforce. Each menu button is associated with one action flow.

App Creation and Execution Overview

Apps can take as little as 10 minutes to create. There is also a Quick App function to create simple list apps and parent-child apps in less than one minute. Apps are built by working from left to right on the X-Author Designer ribbon, starting with the Apps button and ending with the Application Settings button. You begin by adding the Salesforce objects and fields and then follow with creating maps, actions, action flows, and user menus.

During runtime, the order is reversed. The end user clicks on a ribbon button that calls an action flow, which executes a sequential set of actions that retrieve, save, or manipulate the data..

App Migration and Cloning

Using X-Author for Excel's import and export capabilities, you can quickly transfer apps from one Salesforce org to another.

> The objects and fields in the source Salesforce org (the org you are exporting from) must also exist in the target Salesforce org.

You can also clone apps using the Clone function located in the Save Application menu in the X-Author Designer ribbon.

Testing Apps

App design is performed in the X-Author Designer tab, but app testing is performed in the X-Author for Excel tab. When designing and testing apps, you will need to switch ribbon tabs using the following procedure.

To switch between tabs

1. Click the X-Author Designer tab and connect to your Salesforce org.
2. Open an existing app or create a new one.
3. Save the application.
4. Click the X-Author for Excel taband connect to your Salesforce org.
5. Open the app.
6. To make changes in the app, switch to the Designer in the Excel section of the task bar at the bottom of your screen.
7. Make the changes and click the Save Application button.
8. Click the X-Author for Excel tab and reopen the app.

> If you did not previously close the runtime window, X-Author will prompt you to close the previous app.

Connecting to Salesforce from Excel

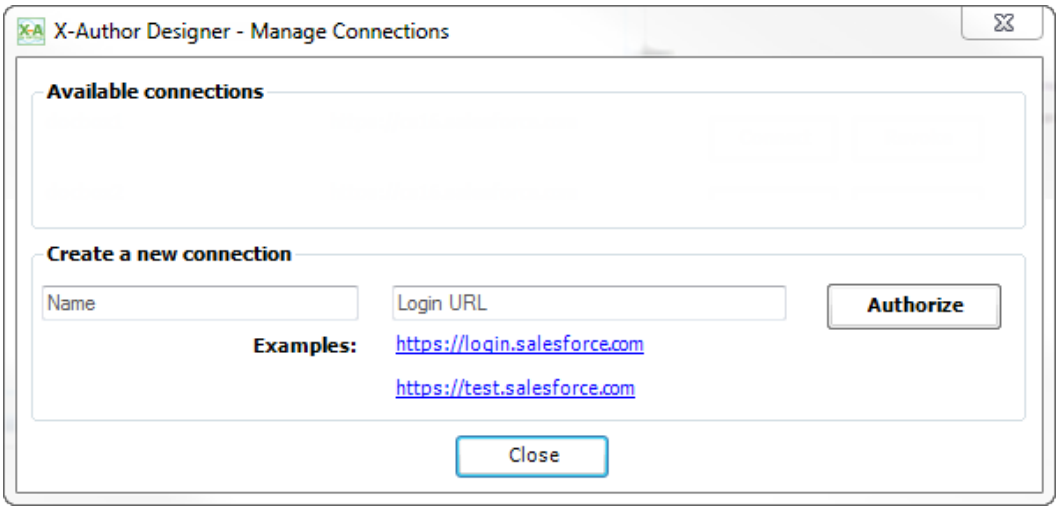
Now that you have installed X-Author, you need to open Excel and connect X-Author to your Salesforce org. You can create connections to multiple Salesforce orgs, switch between orgs without leaving Excel, and revoke existing connections.

Connecting to Salesforce from Excel

Use the following process to connect to your Salesforce org.

To connect to Salesforce from X-Author Designer

1. Open Excel.
2. In the X-Author Designer or X-Author for Excel ribbon menu, click Connect.
   If this is the first time you've connected to Salesforce, the Manage Connections window opens.



3. Under the Create a new connection section and in the Name field, enter the Salesforce.com user name.

> If you have previously connected to Salesforce, but need to connect to a different org, refer to S*witching Salesforce Connections*.

4. In the Login URL field, type the URL you use to log in to Salesforce (In many cases, the URL is login.salesforce.com for production orgs and test.salesforce.com for sandboxes. You can click the shortcut for either of these in the Examples section.).
5. Click **Authorize**. The Salesforce login window opens.
6. Type your username and password in the provided fields and click **Log in to Salesforce**.

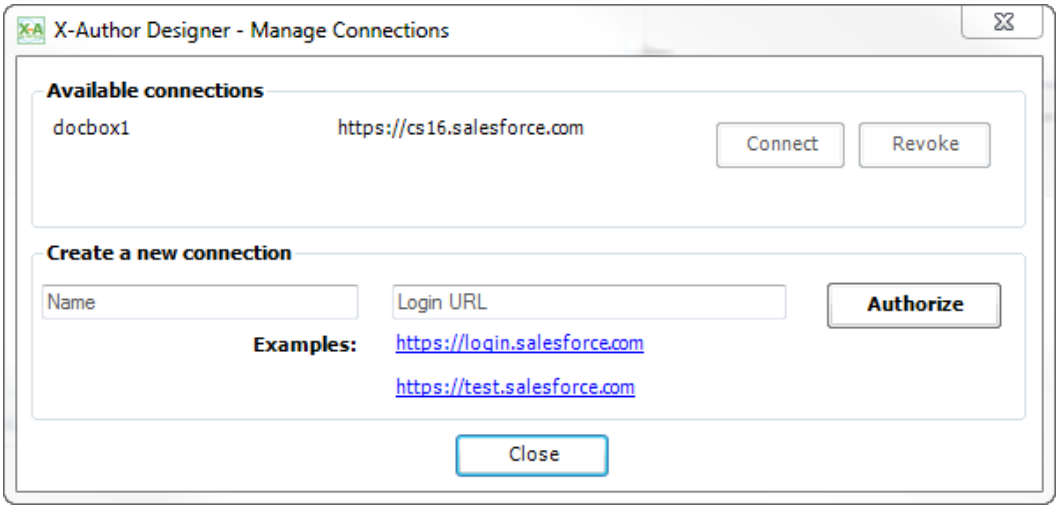
The X-Author Designer ribbon is fully enabled. Now you can start designing an app or using existing apps.

Adding More Connections

If you use X-Author for Excel with more than one Salesforce org, you can add

To add a Salesforce connection

1. Open Excel.
2. In the X-Author Designer or X-Author for Excel ribbon menu, click **Switch Connection** and select **Manage Connections**. The Manage Connections window opens.



3. Under the Create a New Connection section, enter the Salesforce.com user name.

4. In the Login URL field, type the URL you use to log in to Salesforce (In many cases, the URL is login.salesforce.com for production orgs and test.salesforce.com for sandboxes. You can click the shortcut for either of these in the Examples section.).
5. Click **Authorize**. The Salesforce login window opens.
6. Type your username and password in the provided fields and click **Log in to Salesforce**.

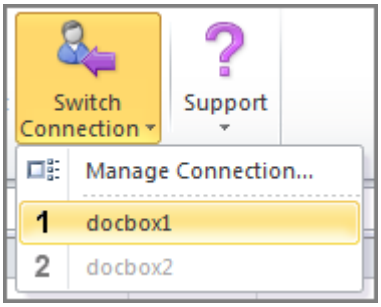
The X-Author Designer ribbon is fully enabled. Now you can start designing an app or using existing apps.

Switching Salesforce Connections

If you have previously connected to a Salesforce org and now need to connect to a different org, you will need to switch connections because the connection will default to the org you most recently connected to. Use the following procedure to switch from one org to another, regardless of whether or not you are already connected to an org.

To switch Salesforce connections

1. In the X-Author Designer or X-Author for Excel ribbon menu, click **Switch Connection**.



2. Select a connection from the drop-down menu. The Salesforce login window opens.
3. Type your password and click **Log in to Salesforce**.

The X-Author Designer ribbon is fully enabled. Now you can start designing an app or using existing apps.

Revoking Connections

If you no longer need to connect to a Salesforce org, you can revoke the connection. Once a connection is revoked, you will no longer see that connection as an option when switching or managing connections.

To revoke a Salesforce connection

1. Open Excel.
2. In the X-Author Designer ribbon menu, click **Switch Connection** and select **Manage Connections**. The Manage Connections window opens.
3. Click the **Revoke** button next to to the connection you want to remove. The Revoke Access confirmation window opens.
4. Click **Yes** to confirm the revocation.

The connection will no longer be available in X-Author.

Creating Your First Apps

In this section, you will learn how to create two Applications: Opportunity Forecast and Opportunities by Account. These are two of the simplest Apps you can build with X-Author for Excel and although they would take less than a minute or two to create using the Quick Application function, it's important to work through the steps in detail here to understand the Designer functions and how they can be applied in more complex use cases.

Before you build your first app, be sure you have read Getting Started.

Create an Opportunities by Account app.

Create an Opportunity Forecast app.

Create an Opportunities by Account Quick App.

Opportunities by Account Application

App Objectives

1. The end user will search for and select an *Account* from Salesforce.
2. Based on the account selected, the app will automatically retrieve all Opportunities into Excel.
3. For the *Account*, the app will include the *Name* and *Phone* fields.
4. For each *Opportunity,* the app will include the *Name*, *Amount*, *Close Date,* and *Stage* fields.

5. Allow the end user to save modifications to just the *Amount* and *Stage* fields for each Opportunity.

App Components

You will use the following app components to build your new app.

Component	Component Detail
Salesforce Objects	Account (Individual) and Opportunity (List)
Display Map	Individual and List fields per objectives
Save Map	Using fields from the Display Map
Actions	• Search and Select Action for the Account • Query Action to retrieve the *Opportunities* for the *Account* • Display Action to transfer the retrieved data to the Display Map • Save Action to save back to Salesforce any user modifications
Action Flows	One to retrieve the data and display it and one to save it
User Menus	One button for the user to retrieve opportunities and one to save them

Creating the Opportunity Forecast Application

Application creation is a multi-step process, starting with naming the app and ending with the creation of your end-user menus. The steps are:

1. Create the new App.
2. Select Salesforce Objects to use in the app.
3. Create Display and Save Maps.
4. Create Query, Display, and Save actions.
5. Create an Action Flow.
6. Create end-user menus.

The steps are described in detail in the following sections.

Prerequisites

You must be connected (through the X-Author Designer) to a Salesforce org that is enabled for X-Author for Excel.

To create the Opportunities by Account App

1. Click **Apps** in the ribbon menu and select **New Application**.
2. Type *Opportunities by Account* in the App Name Field and click the **New** radio button in the Select Template section.
3. Click **Create**

To select Salesforce Objects

1. Click Salesforce Objects and locate Account > Opportunity and then click Add Child Object.
2. Click Account (Ind) in the right-hand window.
3. Select fields from Add Fields below: Region, Name and Phone.
4. Click Apply.Note: You must click Apply after you have selected the fields for any object.
5. Click on Opportunity (List) in the right-hand window.
6. Select fields from Add Fields below: Account Id (is automatically selected for you because this is the lookup field to the parent object in the object hierarchy), Amount, Close Date, Name, and Stage.
7. Click Apply.
8. Click Save and Close.
9. Save the Application.

To create a Display Map

1. Select Display Maps and click Create.
2. Enter *Main* into the Name field and select Individual from the Type list.
3. Drag Account Name to C3 and Phone to C4. You will see the link between the cell and field stamped in the bottom right-hand window.

4. Switch to List.
5. Drag Name to B7, Amount to C7, Close Date to D7, Stage to F7, and Id to H7. Now click on the orange account id field (with a + next to it) and it will take you to the Account object. Select the Region field from there and drag it to G7. X-Author allows you to have fields from any related object on the same list row.
6. Select Options and change the Sort by picklist to Name.
7. Save the Display Map and click Close.
8. Add a title to B1: *Opportunities by Account*.
9. Save the Display map.

To create a Save Map

1. Select Save Map and then click Create.
2. Enter *Main* in the Name field.
3. Click Add Display Map Field and select Amount and Stage.
4. Click Apply and then click Save.
5. Save the Application.

To create a Search and Select Action

1. Click the arrow under Actions on the menu and select Search and Select.
2. Enter *Select Account* into the Action Name field.
3. Choose Account from the Salesforce Objects list.
4. Click the check box next to Name in the Search window.
5. Click the check box next to Name in Fields to Display in Search Results and the Sort check box to the right.
6. Choose Single from the Results area.
7. Click Save.

To create a Query Action

1. Click the arrow under Actions on the menu and select Query.
2. Enter *Query Opportunities* into the Action Name field.

3. Choose Account.Opportunity from the Object list.
4. In the Filters section, click the lookup icon and select the account id field to restrict the *Opportunity* records to those that match the *Account* that was selected by the user. Because this is a lookup field, click just to the right of the orange field to select the row and click Apply. Select the other parameters based on the table below.

Salesforce Object and Field	Value Type	Operator	Value
Account ID	Input	Equals	Account Id from the Accounts object will default. (i.e. Account.AccountID)

5. Click Save.

To create a Display Action

When data has been retrieved from Salesforce, it is placed in Excel memory, waiting to be placed into the specific worksheet based on a Display Map. You may want to display different maps at different times when your App is in use, so a Display Action lets you decide which map to use and when to use it.

1. Click the arrow under Actions and select Display.
2. Enter *Display Opportunities* as the Action Name.
3. Choose Main from the Display Map list.
4. Click Save.

To create a Save Action

1. Click the arrow under Actions and select Save.
2. Enter *Save Opportunities* into the Action Name field.
3. Select Main from the Save Map list.

Now that you have created your actions, you need to **create Action Flows** to retrieve and display data and to save data.

To create a flow to retrieve and display Opportunities

1. Click Action Flow and then click Create.
2. Enter *Retrieve Opportunities* into the Action Flow Name field and click Apply. The left-hand window is updated.
3. Click Add Step.
4. Enter *Select Account* into the Step Name field.
5. Select Select Account from the Action list.
6. Select the check box next to Persist Data as Output and enter *Account* in the Action Output field.
7. Click Apply and then click Add Step.
8. Enter *Query Opportunities* in the Step Name field.
9. Select Query Opportunities from the Action list.
10. Select the check box next to Persist Data as Output and enter *Opps* in the Action Output field.
11. Select the check box next to Provide Input and then select Account from the picklist.
12. Click Apply and then click Add Step.
13. Enter *Display Opportunities* in the Step Name field and select Display Opportunities from the Action list.
14. Select the check box next to Provide Input and select Account and Opps from the Action Input list.
15. Click Apply and then click Save.

To create a flow to save Opportunities

1. Click Action Flow and then click Create.
2. Enter *Save Opportunities* in the Action Name field and then click Apply.
3. Click Add Step and choose Save Opportunities from the Action list.
4. Enter *Save Opportunities* into the Step Name field and click Apply.
5. Click Save.

6. Save the Application.

The **final step** is to **add the end-user menus**. Creating menus is a multi-step process.

1.
   - You first add a *Group* and highlight the group to add buttons underneath it.
   - For each group or button, you can change the names of each by overwriting the *Default Group* or *Display Item* in the right window.
   - When you add a button, *Order* is the order in which the buttons will appear on the end user menu.
   - *Action Flow* is for associating a menu item to the Action Flow that will be executed by the end user.
   - For icons, X-Author uses the standard Microsoft Office icon list which can be found here: http://www.microsoft.com/en-us/download/details.aspx?id=21103. Use what is provided below or any other icon from the list. You must enter the icon capitalization exactly as it is described.

The steps are described in detail in the following sections.

To create User Menus

1. Click User Menus.
2. Click Add Group.
3. Enter *Opportunities* in the Name field and *1* in the Order field.
4. Click Add Button and enter *Retrieve Opportunities* in the Name field.
5. Enter *GetExternalDataFromOtherSources* in the Icon field.
6. Enter *1* in the Order field.
7. Select Retrieve Opportunities from the Action Flow list.
8. Click on Opportunities to highlight the group and then click Add Button.
9. Enter *Save Opportunities* in the Name field.
10. Enter *SaveObjectAs* in the Icon field.
11. Enter *2* in the Order field.
12. Select Save Opportunities from the Action Flow list.
13. Click Save

14. Save the Application.

To make it look good

1. Enlarge the title.
2. Highlight the row headings.
3. Identify to the end user the columns or cells that are modifiable by highlighting them in a different color.
4. You can delete the text in brackets. This does not remove the fields because those are in the map. This just makes the presentation more pleasing.
5. Make sure to place your cursor in the cell and use the *Delete* key. Do not use row delete!

A new App with the name *Opportunities by Account* has been created.

Opportunity Forecast Application

App Objectives

1. The end user will retrieve into Excel all opportunities for which he/she is the owner.
2. For each *Opportunity* include: *Name*, *Amount*, *Close Date*, *Stage*, and *Owner Id*.
3. Allow the end user to save modifications to just the *Amount* and *Stage* for each *Opportunity*.

App Components

You will use the following app components to build your new app.

Component	Component Detail
Salesforce Objects	*Opportunity* (List)
Display Map	*Opportunity* Fields as above
Save Map	Using fields from the Display Map
Actions	• Execute Query Action with a restriction on "Owner ID"

Component	Component Detail
	• Display Action to transfer the retrieved *Opportunity* data to the Display Map • Save Action to save back to Salesforce any user modifications
Action Flows	One to retrieve opportunities and one to save them
User Menus	One button for the user to retrieve opportunities and one to save them

Creating the Opportunity Forecast Application

Application creation is a multi-step process, starting with naming the app and ending with the creation of your end-user menus. The steps are:

1. Create the new App.
2. Select Salesforce Objects to use in the app.
3. Create a Display and Save Maps.
4. Create Query, Display, and Save actions.
5. Create and Action Flow.
6. Create end-user menus.

The steps are described in detail in the following sections. You will complete each step as you build your Opportunity Forecast app.

Prequisites

You must be connected (through the X-Author Designer) to a Salesforce org that is enabled for X-Author for Excel.

To create the Opportunity Forecast App

1. Click **Apps** in the ribbon menu and select **New Application**.
2. Type *Opportunity Forecast* in the App Name Field and click the **New** radio button in the Select Template section.

3. Click **Create**.

To select Salesforce Objects

1. Click **Salesforce Objects** in the ribbon menu.
2. Locate Opportunity in the Objects list either by scrolling or by typing in the first letter to jump to the nearest match and then scrolling.
3. Click **Add Parent Object**. This will always be true if you have just one object. The object is now in the selected box on the right and is automatically assigned default *List* behavior.
4. In Add Fields below, select the following field names: Account ID, Amount, Close Date, Name, Owner ID, and Stage.
5. Click **Apply**.
   Note: You must click Apply after you have selected the fields for any object or these will not be saved if you have more than one object.
6. Click **Save and Close**.
7. Click the **Save Application** button.

To create a Display Map

1. Click **Display Map** and select **Create**.
2. Type *Main* into the Name field and select List from the Type drop-down.
3. Drag Account Name to A7. This was not an available field in Salesforce Objects but X-Author for Excel always brings in the name field for any lookup field selected in Salesforce Objects.
4. Drag Name to B7, Amount to C7, Close Date to D7, Stage to F7, Opportunity Id to H7.
5. Click **Options** and change the Sort by list to Account Name.
6. Click **Save** and then click **Close** to save the Display Map.
7. Add a title in A1, e.g. *Opportunities Forecast*.
8. Save the Application.

To create a Save Map

1. Click **Save Maps** and select **Create**.

2. Enter *Main* into the Name field.
3. Click Add Display Map Fields and select Amount and Stage.
4. Click **Apply** and then click **Close** for the Save Map.
5. Click the **Save Application** button.

To create a Query Action

1. Click the arrow under Actions and select Query.
2. Type *Query Opportunities* into the Action Name field and select Opportunity from the Object picklist.
3. In the Filters section, select or enter the following to restrict the Opportunity records to those for which the current user is the *Opportunity* owner. Click the lookup icon and select the owner idOK field. Because this is a lookup field, click just to the right of the orange field to select the row and click Apply. Select the other parameters based on the table below.

Salesforce Object and Field	Value Type	Operator	Value List
Owner Id	SystemVariables	Equals	CurrentUser

4. Click Save.

To create a Display Action

When data has been retrieved from Salesforce, it is placed in Excel memory, waiting to be placed into the specific worksheet based on a Display Map. You may want to display different maps at different times when your App is in use, so a Display Action lets you decide which map to use and when to use it.

1. Click the arrow under Actions and select Display.
2. Enter *Display Opportunities* into the Action Name field.
3. Choose Main from the Display Map list.

To create a Save Action

1. Click the arrow under Actions and select Save.

2. Enter *Save Opportunities* into the Action Name field.
3. Select Main from the Save Map list.
4. Click Save.
5. Save the Application.

Now that you have created your actions, you need to **create Action Flows** to retrieve and display data and to save data.

To create a flow to retrieve and display Opportunities

1. Click Action Flow and then click Create.
2. Type *Retrieve Opportunities* into the Action Flow Name field and click Apply. The left-hand window is updated.
3. Click Add Step.
4. Type *Query Opportunities* in the Step Name field.
5. Select Query Opportunities from the Action list.
6. Check Persist Data as Output.
7. Enter *Opps*, then click Apply.
8. Click Add Step.
9. Enter *Display Opportunities* into the Step Name.
10. Select Display Opportunities from the Action list.
11. Check Provide Input and choose *Opps* as default.
12. Click Apply and then click Save.
13. Save the Application.

To create an action flow to save Opportunities

1. Click Action Flow and then click Create.
2. Enter *Save Opportunities* into the Action Flow Name field and click Apply. The left hand window is updated.
3. Click Add Step and enter *Save* into the Step Name.
4. Select Save Opportunities from the Action list.

5. Click Apply.
6. Click Save.
7. Save the Application.

The **final step** is to **add the end-user menus**. Creating menus is a multi-step process.

1. You first add a *Group* and highlight the group to add buttons underneath it.
2. For each group or button, you can change the names of each by overwriting the *Default Group* or *Display Item* in the right window.
3. When you add a button, *Order* is the order in which the buttons will appear on the end user menu.
4. *Action Flow* is for associating a menu button to the Action Flow that will be executed by the end user.
5. For icons, X-Author uses the standard Microsoft Office icon list which can be found here: http://www.microsoft.com/en-us/download/details.aspx?id=21103. Use the icons describe below or use any other icon from the list. You must enter the icon name, including capitalization, *exactly* as it is documented.

The steps are described in detail in the following sections.

To create User Menus

1. Click User Menus.
2. Click Add Group.
3. Enter *Opportunities* in the Name field and *1* in the Order field.
4. Click Add Button and enter *Retrieve Opportunities* in the Name field.
5. Enter *GetExternalDataFromOtherSources* in the Icon field and *1* in the Order field.
6. Select Retrieve Opportunities from the Action Flow list.
7. Click on Opportunities to highlight the group and then click Add Button.
8. Enter *Save Opportunities* in the Name field.
9. Enter *SaveObjectAs* in the Icon field and *2* in the Order field.
10. Select Save Opportunities from the Action Flow list.
11. Click Save.

12. Save the Application.

To make it look good

1. Enlarge the title.
2. Highlight the row headings.
3. Identify to the end user the columns that are modifiable by highlighting them in a different color.
4. You can delete the text in brackets. This does not remove the fields because those are in the map. This just makes the presentation more pleasing.

> ⚠ Make sure to place your cursor in the cell and use the *Delete* key. **Do not use Delete Row.**

An App with the name *Opportunities Forecast* has been created. Now you can run it by clicking the Preview button in the ribbon menu.

The user menus you created now appear in the ribbon. Click the Retrieve Opportunities button and the list of opportunities is displayed. Make a few changes. Click Save Opportunities, and you should see a message that the records were saved. The changes are reflected in Salesforce.

Quick App for Opportunities by Account

You will use the Quick App wizard to create your first app, and it will take you only a few minutes.

You'll build your app in only four steps.

1. Specify which type of app you need.
2. Select your objects and fields.
3. Select your display and save options and specify a method for getting the data from Salesforce.
4. Format your workbook.

Let's get started. Using these four steps, you are going to create an app that will:

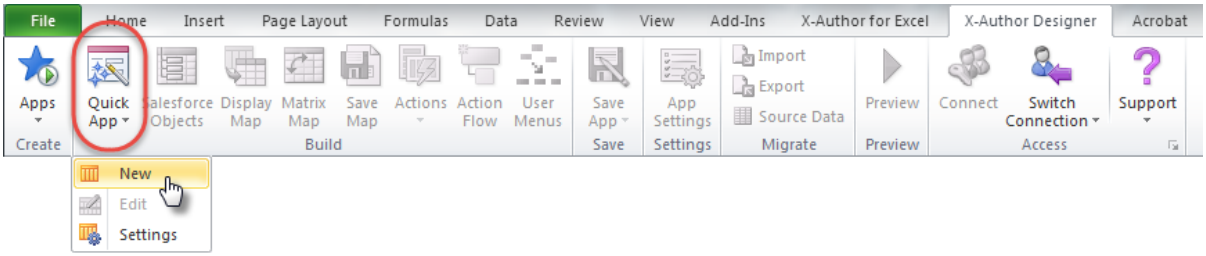
- Retrieve data from the Account and Opportunity objects in Salesforce
- Place that data in a Microsoft Excel worksheet.

- Allow you to edit the data in Excel and then save your changes in Salesforce.

Prerequisite

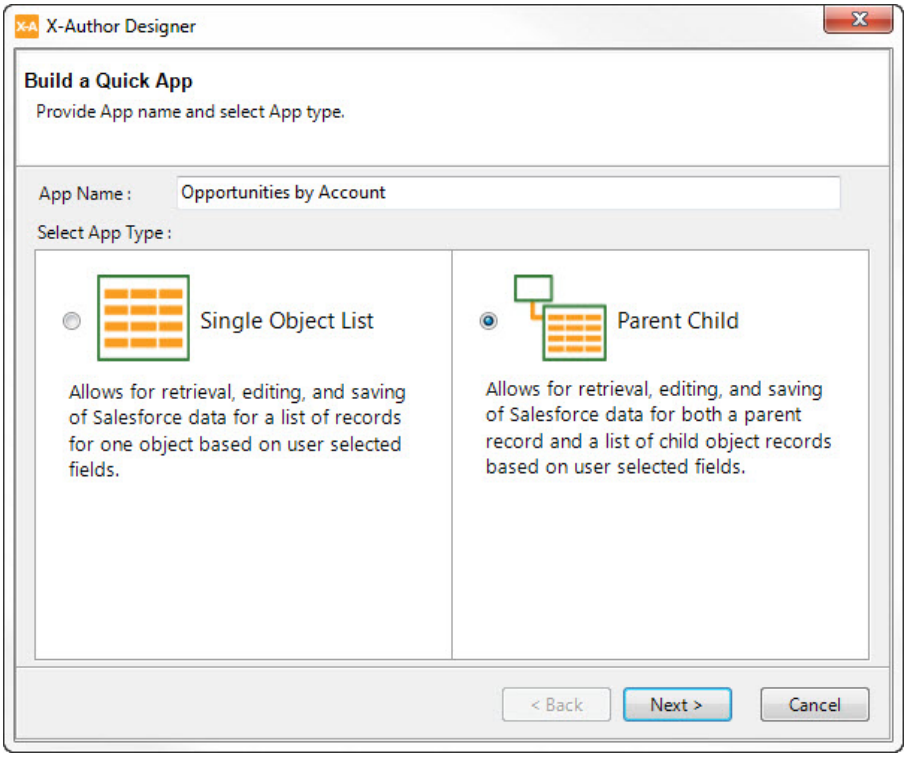
Before you can create an app, you must connect to a Salesforce org that is configured for X-Author for Excel. For information on configuring and connecting to Salesforce, refer to Setting Salesforce Permissions and Connecting to Salesforce from Excel.

1. Click **Quick App** and select **New**.



The Build a Quick App window opens.

2. Type the application name into the App Name field. The name will appear as the Excel window title (located at the top of the Excel window) in the completed app.

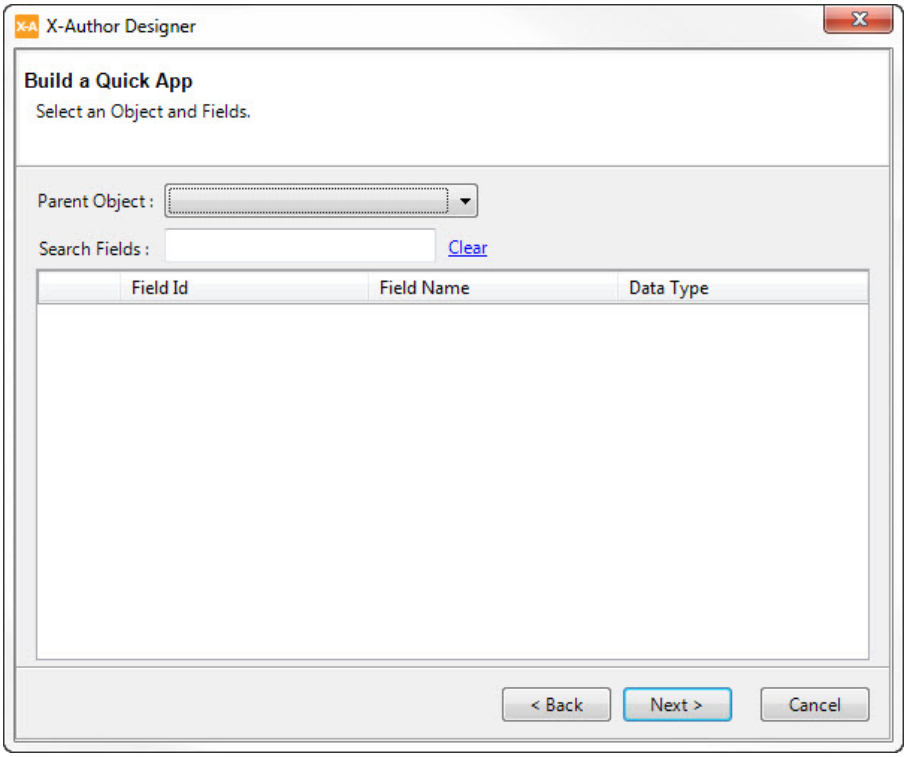


> Give your app a name that is concise but also summarizes the basic purpose of the app.

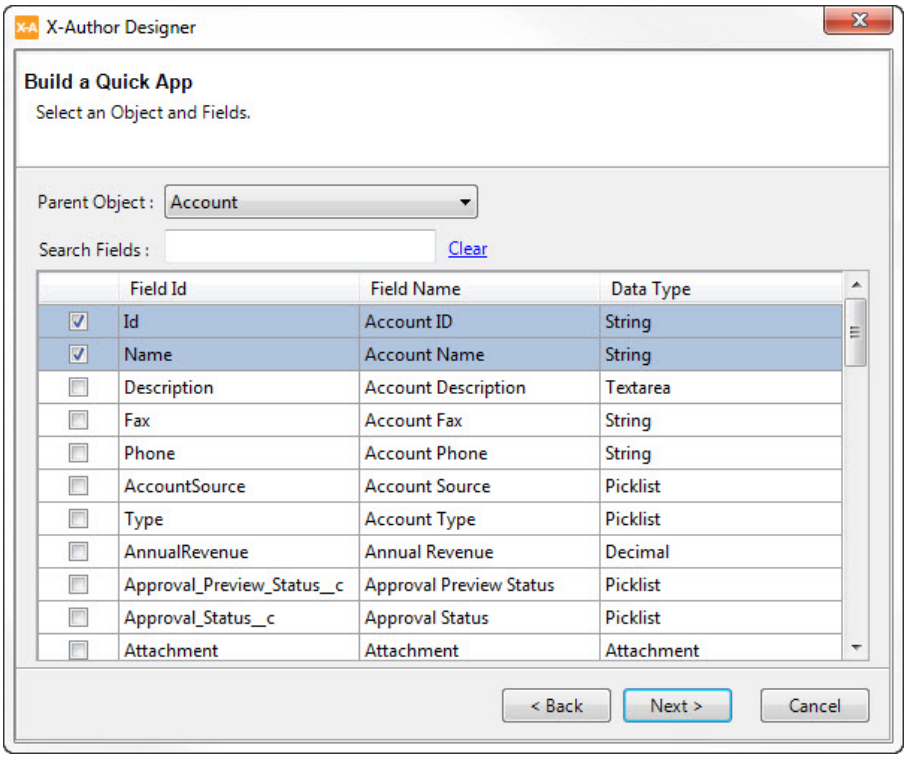
For this example, you are going to create an Opportunity forecasting app. The app allows you to search for and select an account record, and then displays all of the account's related opportunities, so type "Opportunities by Account" in the App Name field.

3. Select an app type by clicking the appropriate radio button.

   Single Object List apps allow you to retrieve data from a single object, such as Accounts. If you need to retrieve data from more than one object, you will need to create a Parent Child app. For this example, you need to retrieve data from both the Accounts object and the Opportunity object, so **select Parent Child and click Next**. Now, you will select the parent object and fields to use in the app.



4. The parent object for your app will be the Account object, so click the Parent Object drop-down menu and select Account. The Account fields are displayed in the field list.



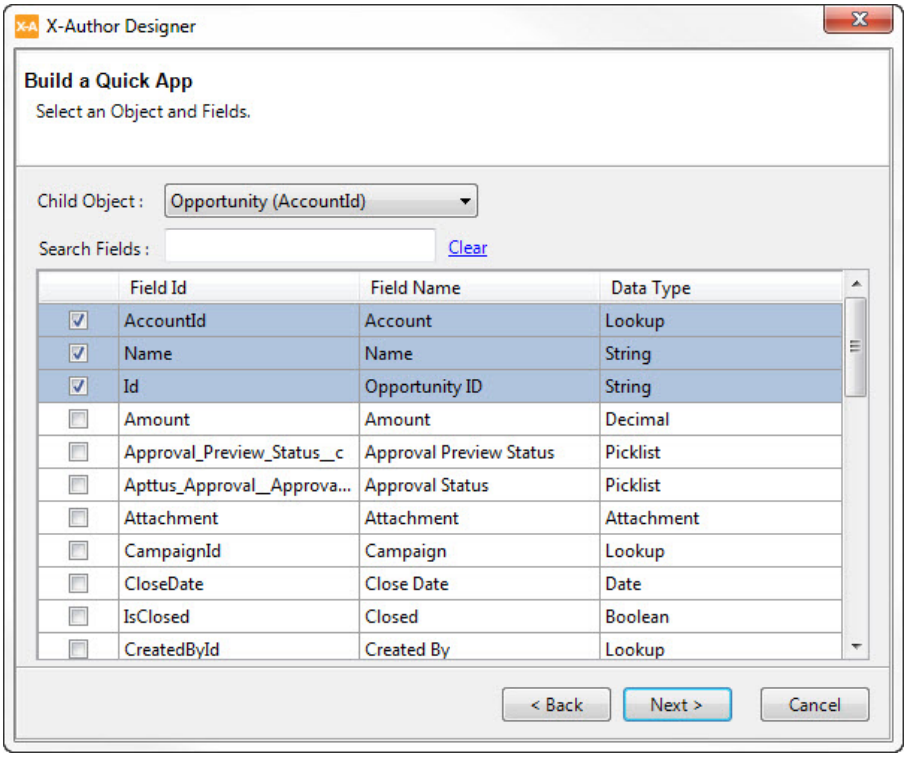
Notice that the two primary fields are selected by default. Select the Account fields that you will be using in your app. Click the check box next to the following fields:

- Type
- Region
- Owner
- Industry
- Website

> You can find fields by scrolling through the list, but it might be easier to use the Search Fields feature. Just type the field name or field ID in the search box. You can also sort the fields by Data Type by clicking on the top of the Data Type column.

5. Click Next.

6. The child object for your app is the Opportunity object, so click the Child Object drop-down menu and select Opportunity. The Opportunity fields are displayed in the field list.

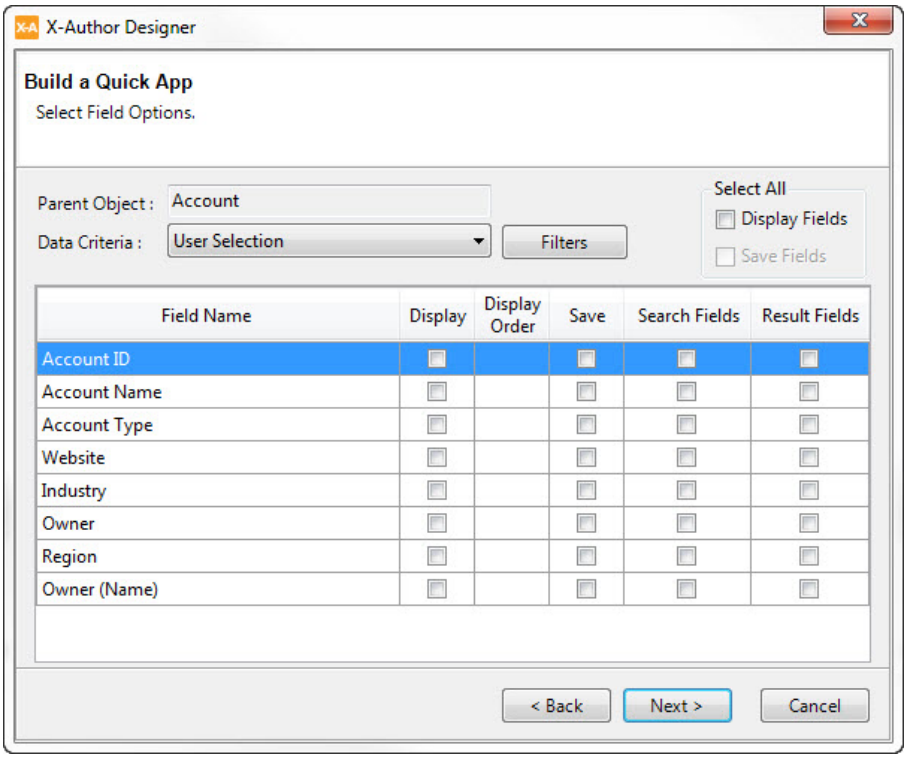


7. Select the Opportunity fields to be used in your app. In a child object, the primary lookup field is also preselected. For this app, you want to allow the user to add new opportunities, so you must select all the required fields for creating a new opportunity record. (With Quick App, this only applies to the child, not to the parent. When creating apps using X-Author Designer, the app can include an ability to create both parent and child records simultaneously.) For this example, the following additional fields are required, so click the check box next to each of these fields.

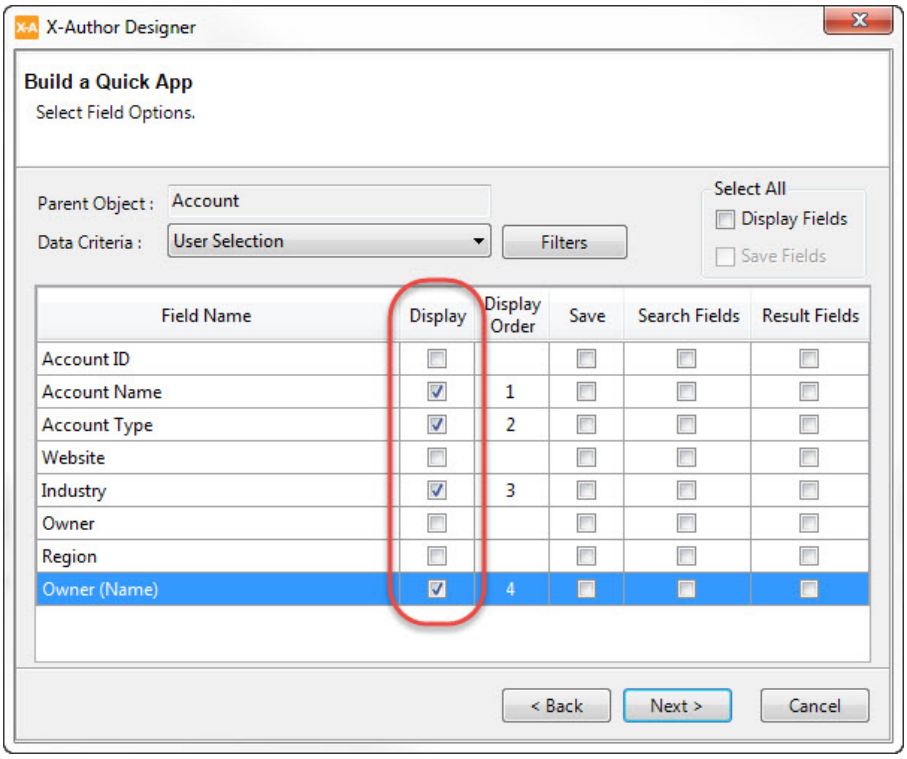
8. • Amount
   • Close date
   • Stage

9. Click Next. You have completed the first two steps.

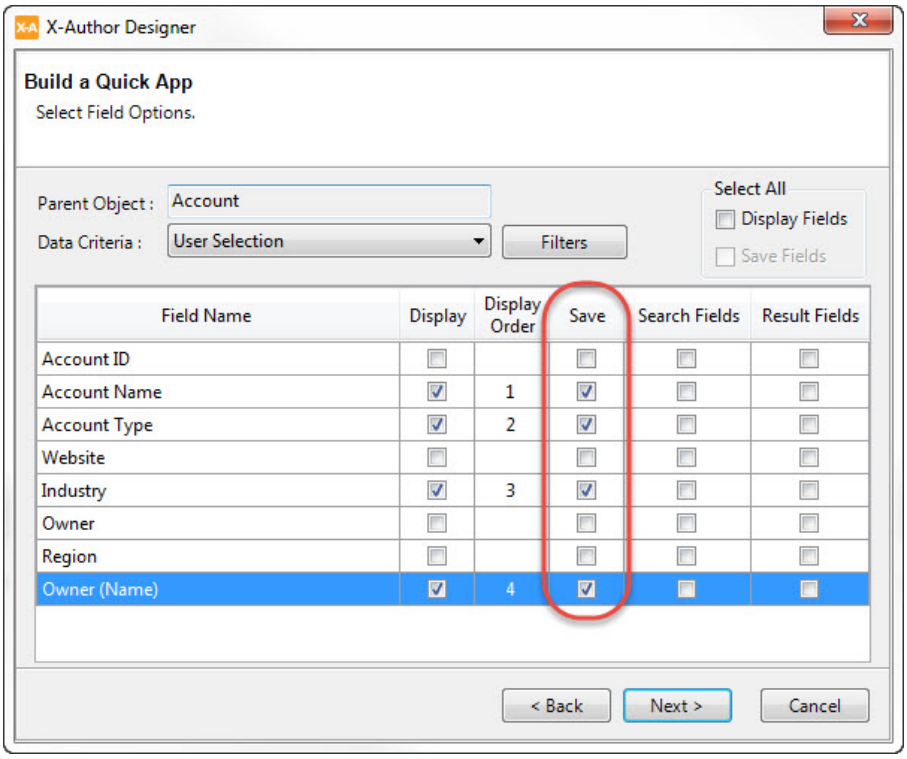
   Now we go to step three, where we decide which fields to display on the Excel workbook, which fields to allow your users to save back to Salesforce, and which method to use to get the data from Salesforce..



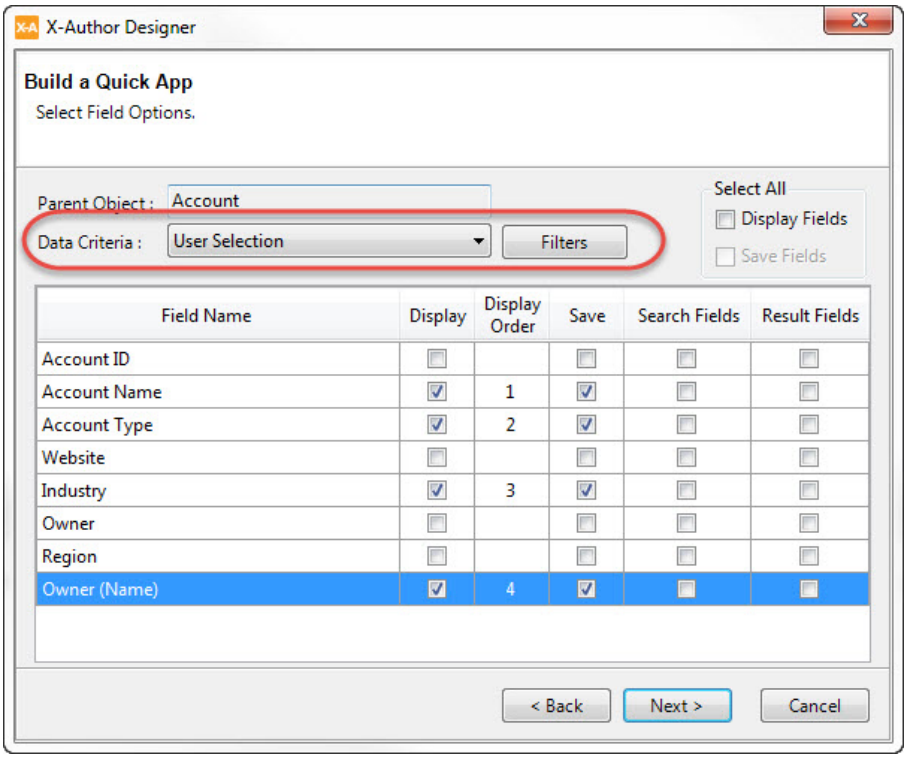
10. Select the fields that you want displayed to the user in the Excel workbook. Notice that both the owner ID—which we previously selected—and its lookup name are available. You can select all the fields by clicking the Display Fields check box in the Select All section (You can change their display order later.), or you can individually select the fields in the order you want them displayed. The parent object fields will display vertically in Excel. For your app, select the following Display fields.



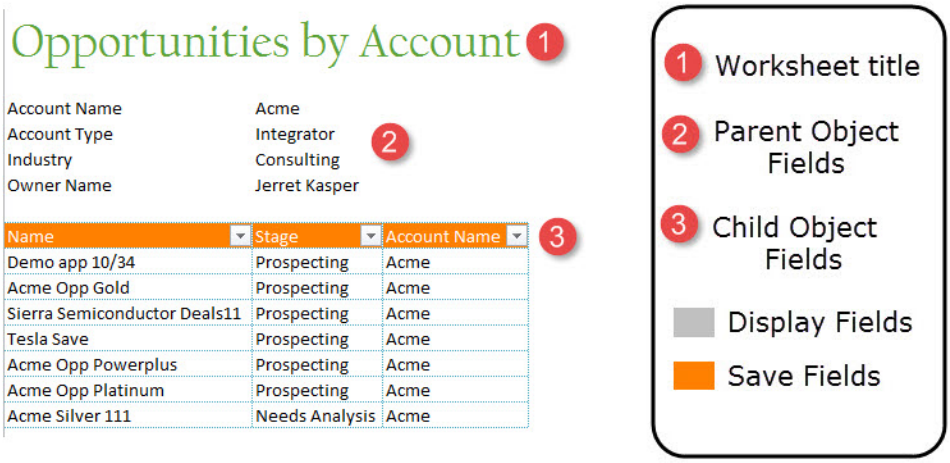
11. Next, specify which fields your users can update and save back to Salesforce by clicking the corresponding check box in the Save column. Once again, you can use the Save Fields check box in the Select All section or individually select the fields. Notice that you cannot select fields that were not specified as Display fields in the previous step. For your app, select the following Save fields.



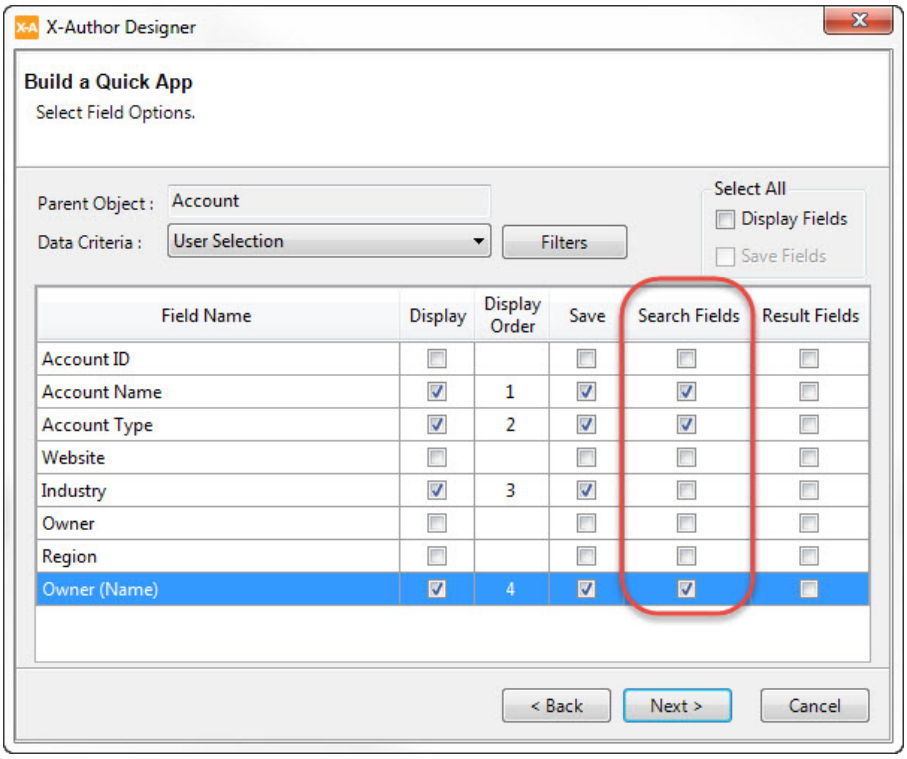
12. Specify how to retrieve the data from Salesforce by selecting an option from the Data Criteria drop-down menu. For the parent record, Quick App provides only one option, User Selection, which allows the user to input search criteria, view the search results, and then select the record they want to retrieve from Salesforce. (This assumes you are opening the app from Excel. If you open the app by launching it from an account page in Salesforce, then that account is automatically selected as the parent record.)



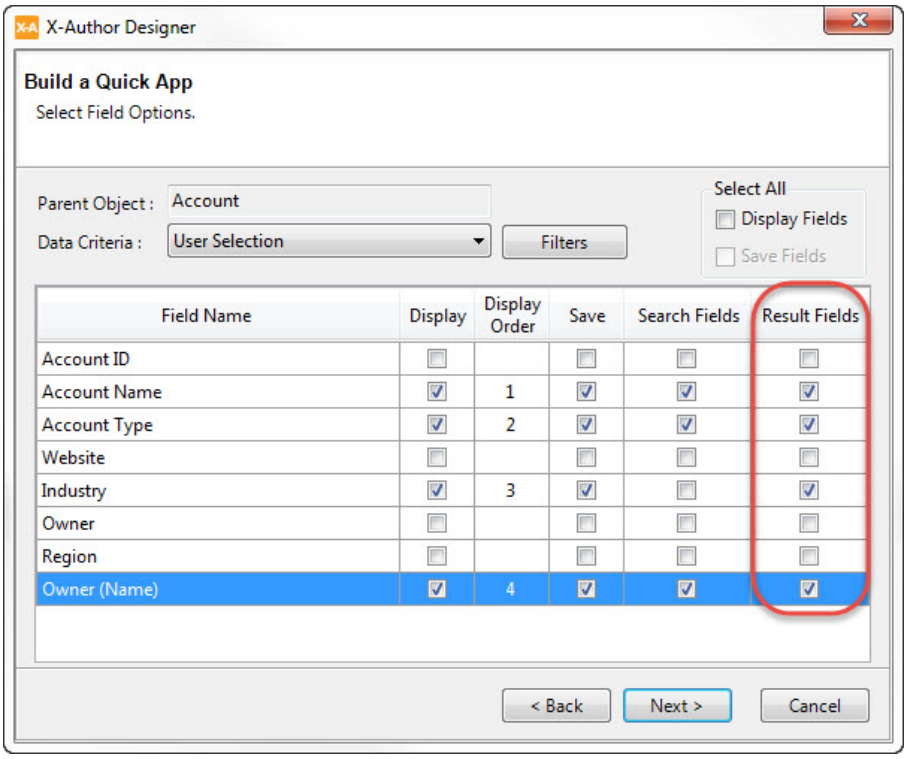
This is how the Display fields will appear when you run the app and retrieve data from Salesforce.



13. Specify the fields that you want your app users to be able to use as search filters To find an account by clicking the corresponding check box in the Search Fields column. For this app, select the following Search fields.



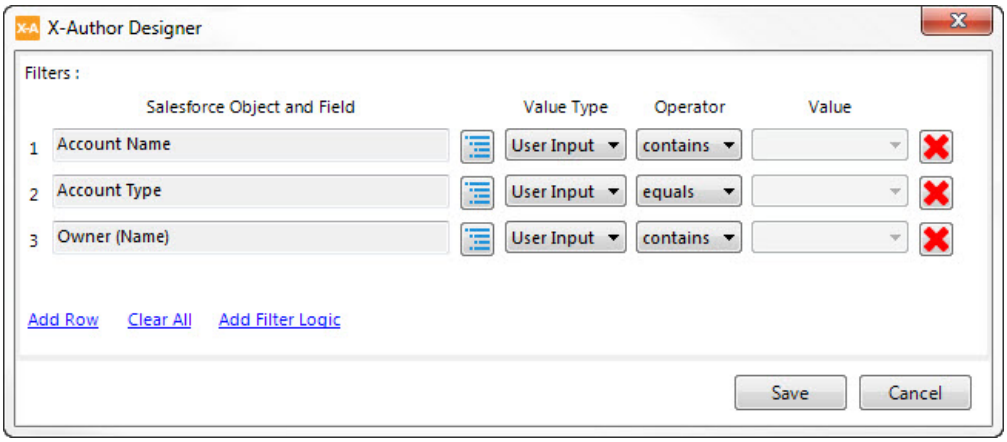
14. Specify the fields that you want to be displayed in the search results by clicking the corresponding check box in the Result Fields column. For this app, select the following Result fields.



Did you know...?

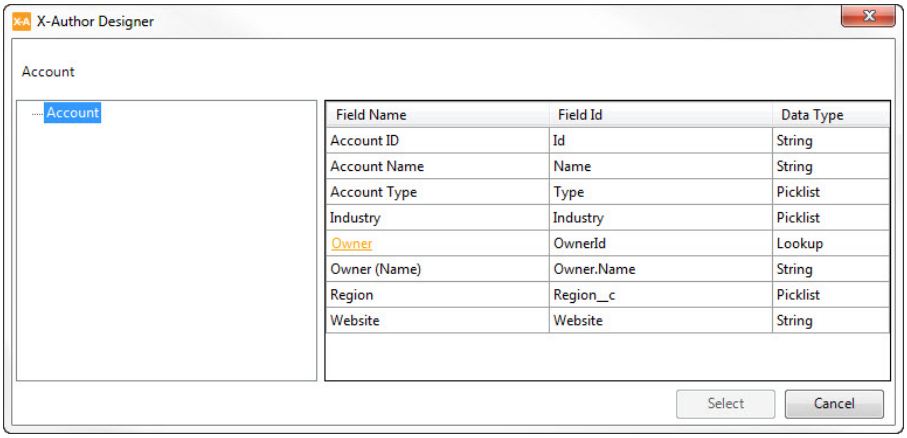
By specifying fields to be used as Search Fields, you just created filters. In the upcoming steps, you will see how to add other filters based on the fields that weren't selected as Search Fields.

15. Next, you will add two filters that will allow users to further filter search results. Start by clicking the Filters button. The Filters window opens.

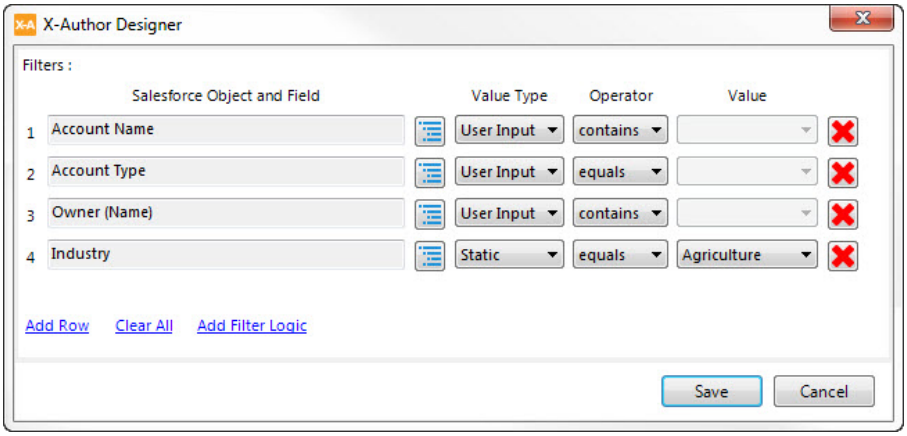


Notice that the three fields you specified as Search Fields are already present as User Input filters. You are going to add two more filters now, one to show only those opportunities in manufacturing, and another to allow your app users to further filter opportunities by region.

16. Click the Add Row button to add a new filter.
17. In the new row, click the Salesforce Object and Field button ( ) to select a field.



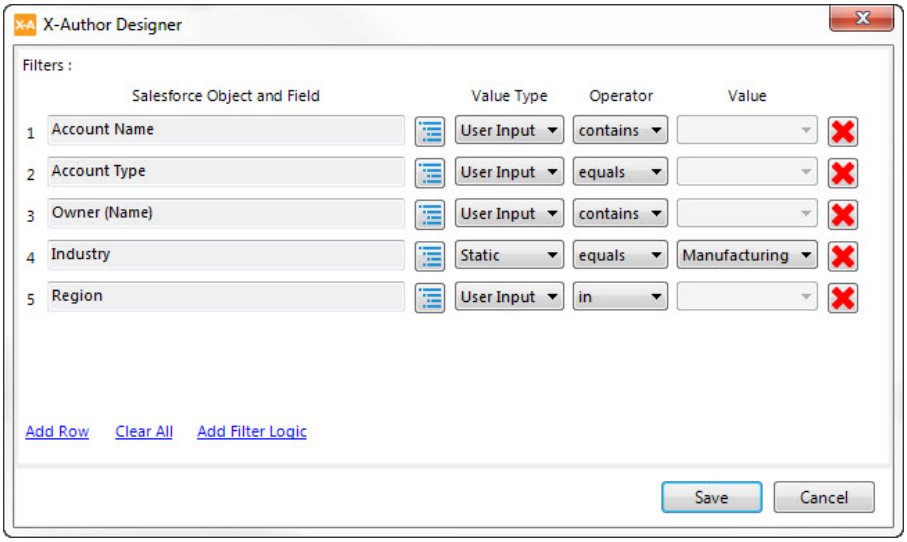
18. Double click the Region row to add that field to your filters.



.

19. For your app, you only want to see those opportunities that are in the Manufacturing industry, so click the Value drop-down menu in the Industry row and select Manufacturing.
20. Now, follow the same process to add another filter, this time choose the Region field to filter on.
21. This time, change the Value Type to User Input and, instead of setting a value, change the Operator from **"equals"** to **"in"** so your users can select from the drop-down list and filter data by region.

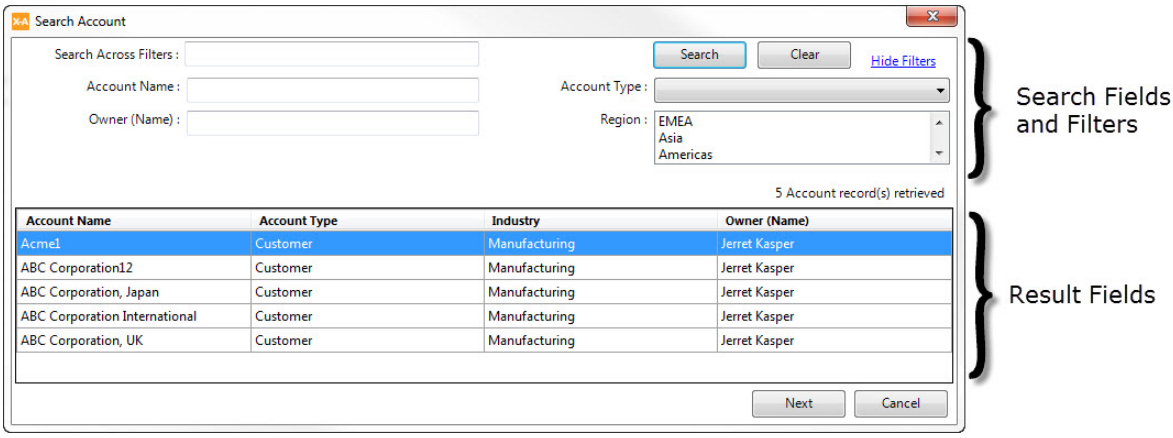
Now your filters should look like this.



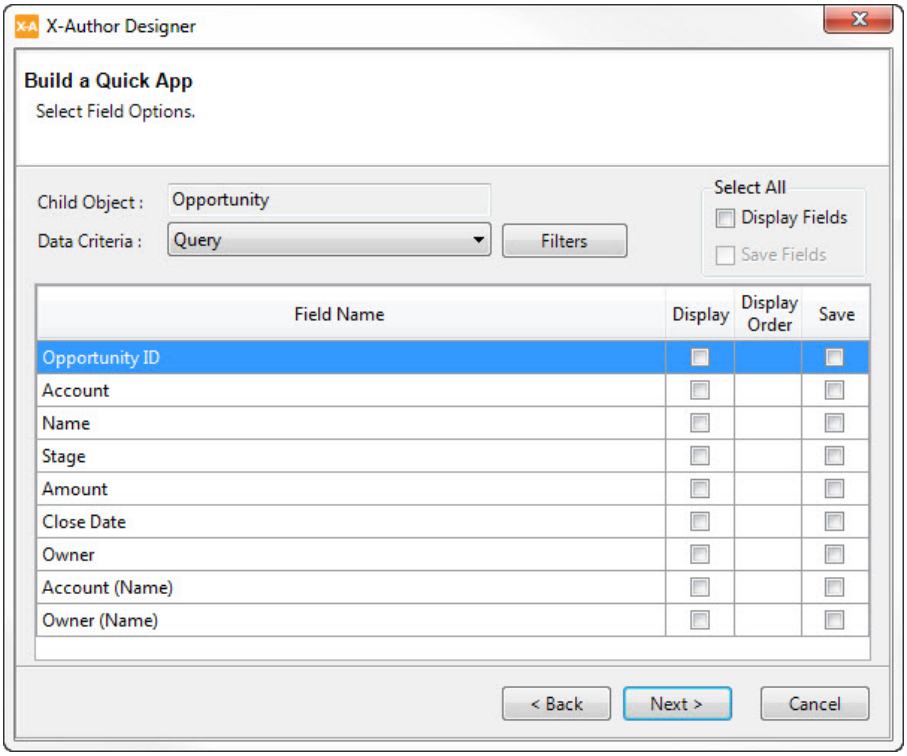
By default, filters are all related by the AND Boolean condition, which means that, in this case, all five of these conditions must be met for a record to be returned as a search result.
To change the way the filters are related, you can add Boolean logic to the filters by clicking the Add Filter Logic button. Filter logic in X-Author for Excel works the same as it does in Salesforce.com. Learn more.

22. Click Save.

This is how your search filters and results fields will look when you run the app.



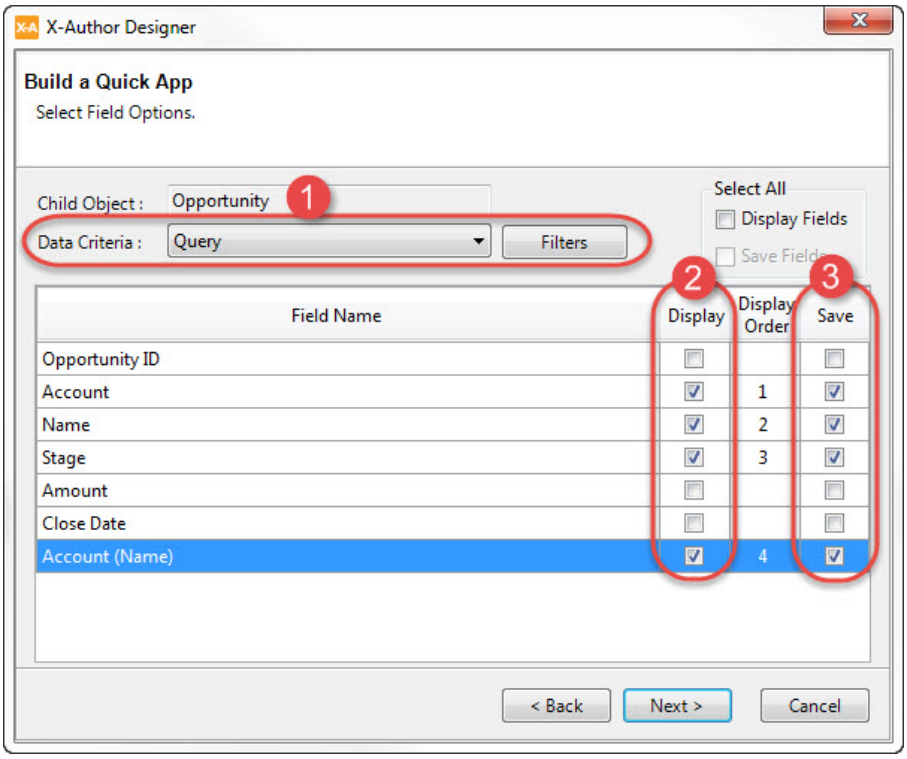
23. You are done configuring the parent object, so click Next to move on to the child object, Opportunity.



Hopefully this window looks familiar to you. It looks different (It has fewer columns) than the window we configured parent-object options in because the data criteria is set to Query.

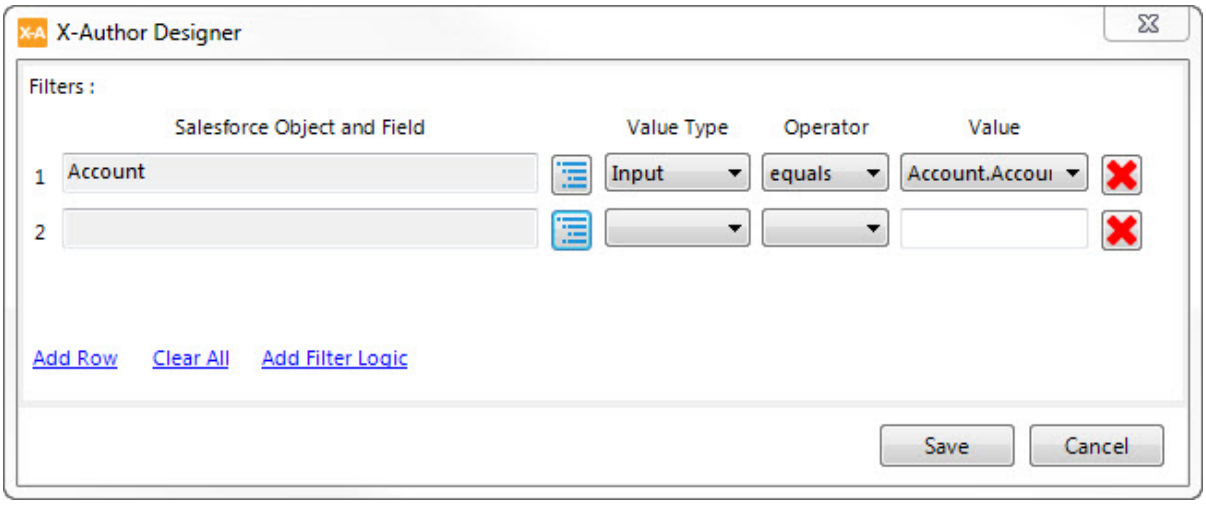
24. If the Data Criteria field is not set to Query by default, set it to Query now.

25. Select the following Display and Save fields. (Note that the fields from the child object will render horizontally.)



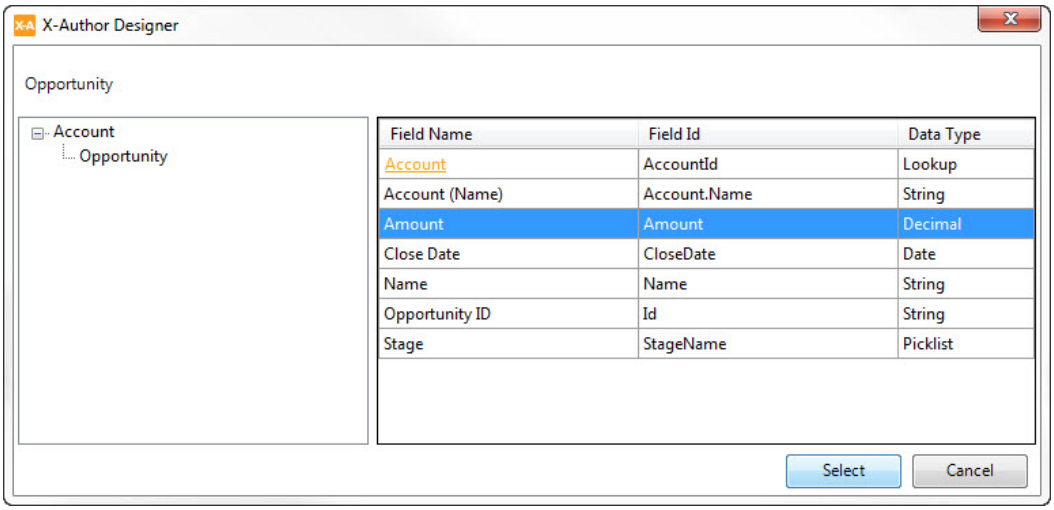
Now, you will add two filters to ensure that you see only those opportunities worth a certain dollar value and in a particular sales stage.

26. Click the Filters button.

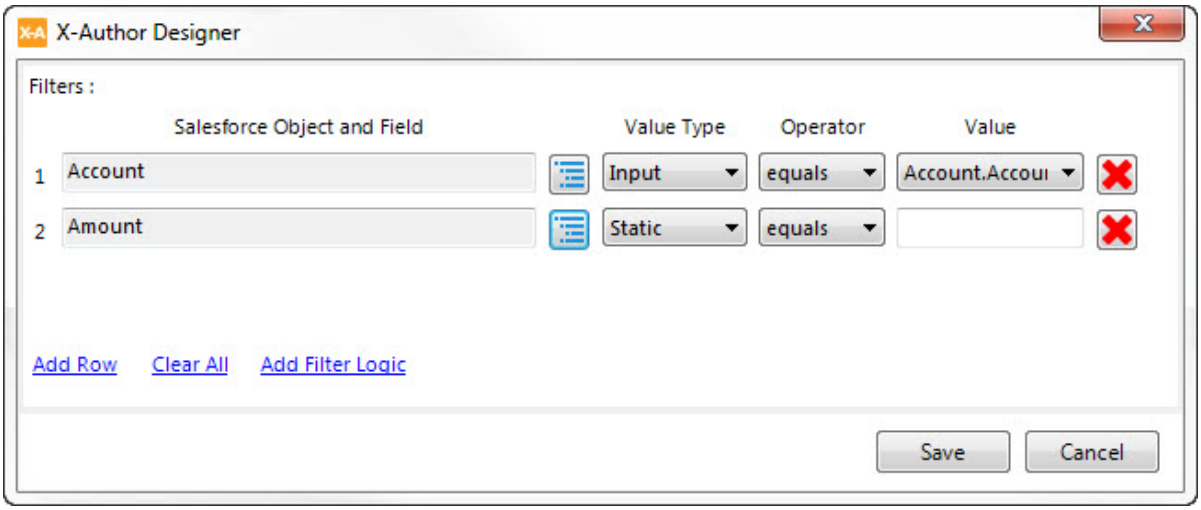


Notice that the parent object is already present as an Input filter. You are going to add two more filters to ensure that you see only those opportunities worth more than $20,000 and only those that are not set to the stage of Closed Won.

27. Click the Add Row button to add a new filter. A new row is added.
28. In the new row, click the Salesforce Object and Field button ( ) to select a field.



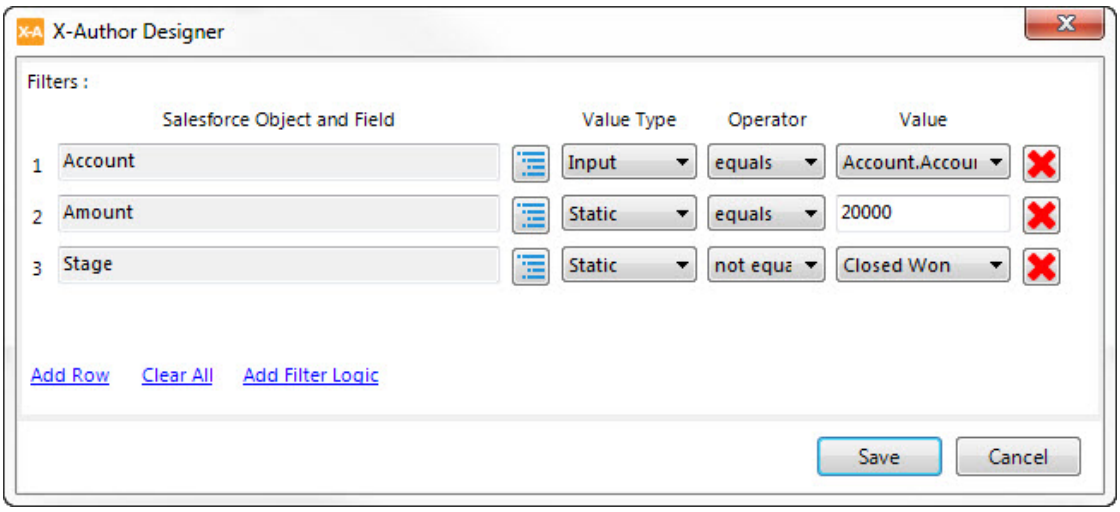
29. Click the Amount field and then click Select.



30. For the Amount field, click the Operator drop-down menu and select **greater than**.
31. In the Value text box, type 20000.
32. Click the Add Row button to add a new filter. A new row is added.
33. In the new row, click the Salesforce Object and Field button ( ) to select a field.
34. Click the Stage field and then click Select.
35. For the Stage field, click the Operator drop-down menu and select **not equal to**.

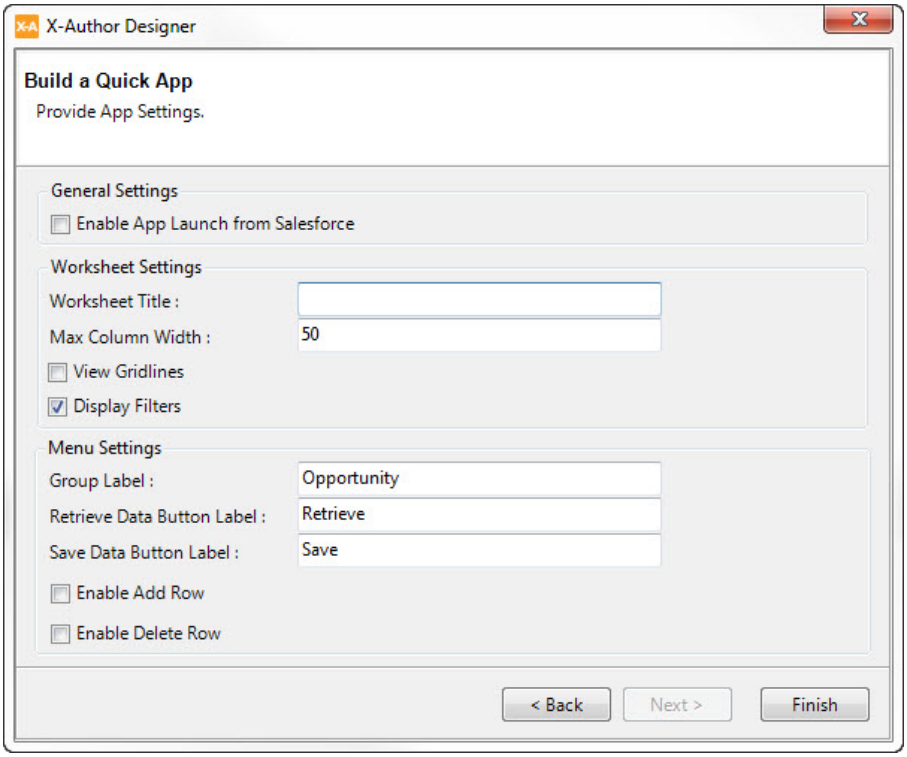
36. Click the Value drop-down menu and select Closed Won.

    Your filters should now look like this.



37. Click Save.

38. Click Next. The App Settings window opens.



All you have to do now is some label work, allow users to add or delete rows in the worksheet, and decide whether the app should be launchable from within Salesforce, and you are done.

39. If this is an app that users will access from a button on a Salesforce page, select the **Enable App Launch from Salesforce** check box. The app you have just built would be a great candidate for this because, when the user is on an account in Salesforce and wants to edit the related Opportunity list, they can just click on a button and do their editing in Excel. (You will also need to do some setup work inside Salesforce, such as putting the button on the right page. For more information, refer to Launching an X-Author for Excel App from Salesforce.)
40. In the Worksheet Title text box, type Opportunities by Account. Remember, this title will appear in field A1 on your Excel workbook.
41. Change the Group Label to "Opportunities."
42. Click the Enable Add Row check box.
43. Click Finish.

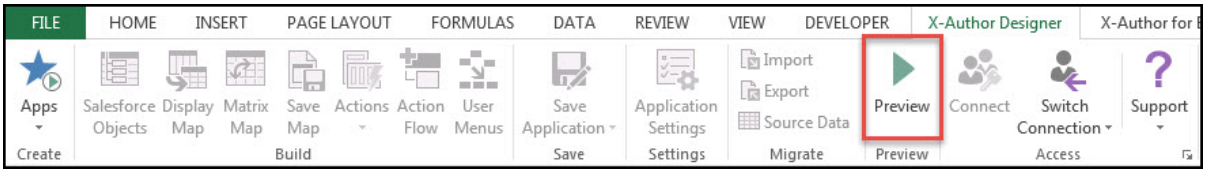
Congratulations! You've completed your first Quick App. Now, you can preview the app to see how it works.

Previewing Apps

Now that you have created an app, you can preview the app, running it as a user would.

To preview an app

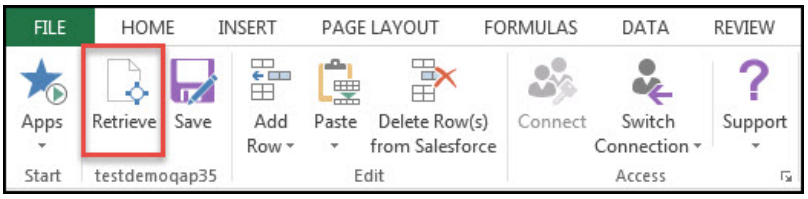
1. After creating an app, click the Preview button in the ribbon menu.



The app opens in the X-Author for Excel tab.

When you click Preview, a new worksheet opens and X-Author for Excel must reconnect to the Salesforce org. This may take a few seconds, so please be patient.

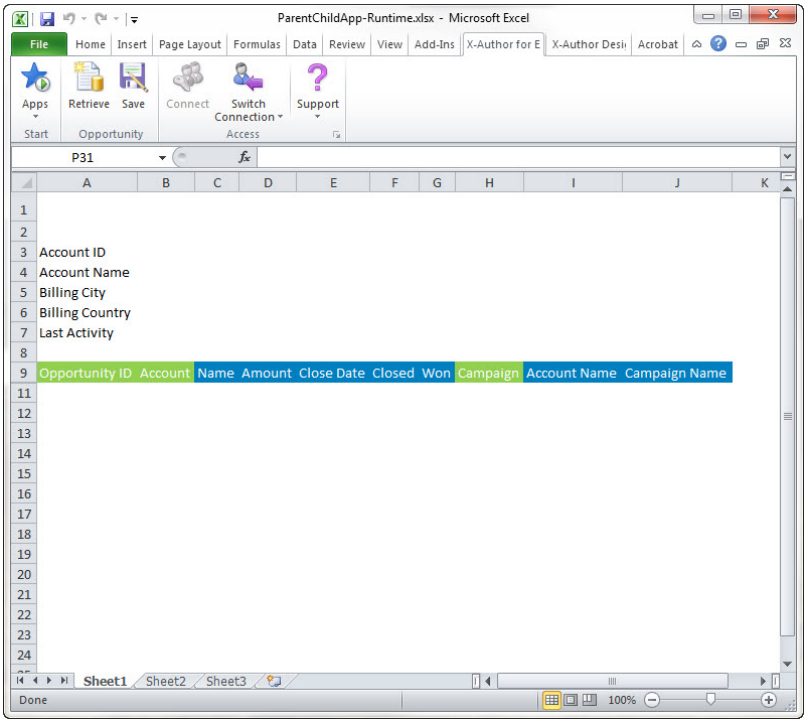
2. In the ribbon menu, click the Retrieve button to download data from Salesforce into Excel.



You can now test your app by retrieving data and adding, pasting, and deleting rows.

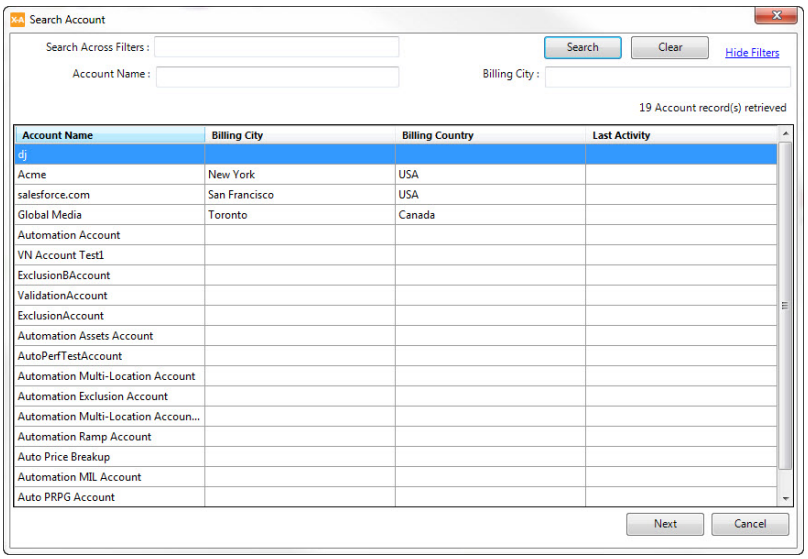
Example

In this example, we'll preview a parent-child app (Accounts and Opportunities, respectively) that allows us to search by Account Name or Billing City. This is what our app looks like when it first opens.



In this example, Display fields are indicated by the blue background and Save fields are indicated by the green background. These colors can be set when creating apps.

When you click the **Retrieve** button, the Search Account window opens. (Account is the parent object in our app.)



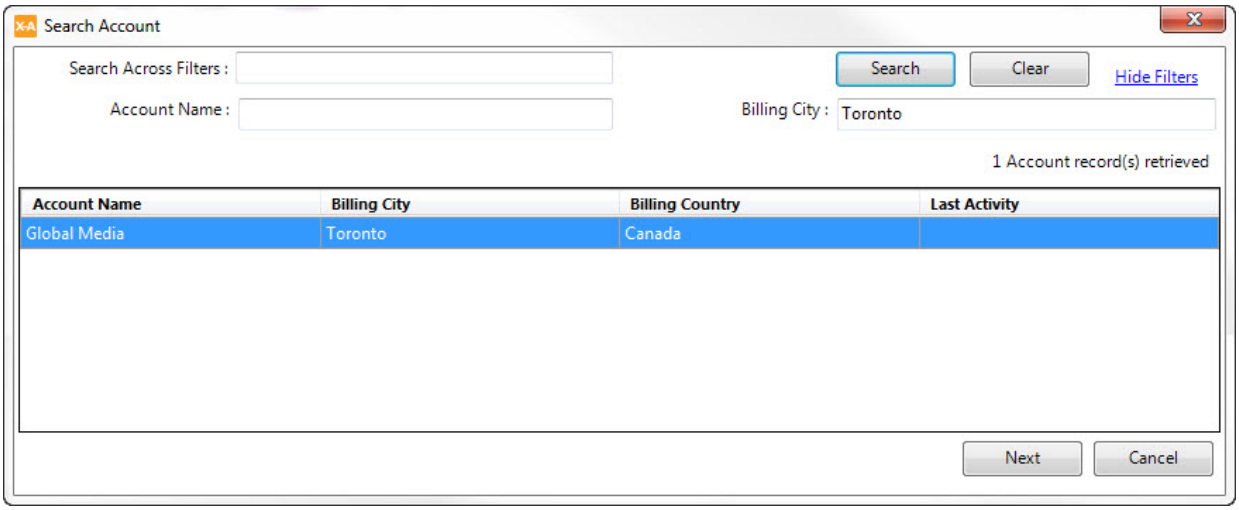
Notice that you can search by Account Name, Billing City. These are the fields that were set as Search fields for the Accounts object when the app was designed.

Notice that the Account Name, Billing City, Billing Country, and Last Activity fields are all displaying in the results panel. These are the fields that were set as Results fields when the app was designed.

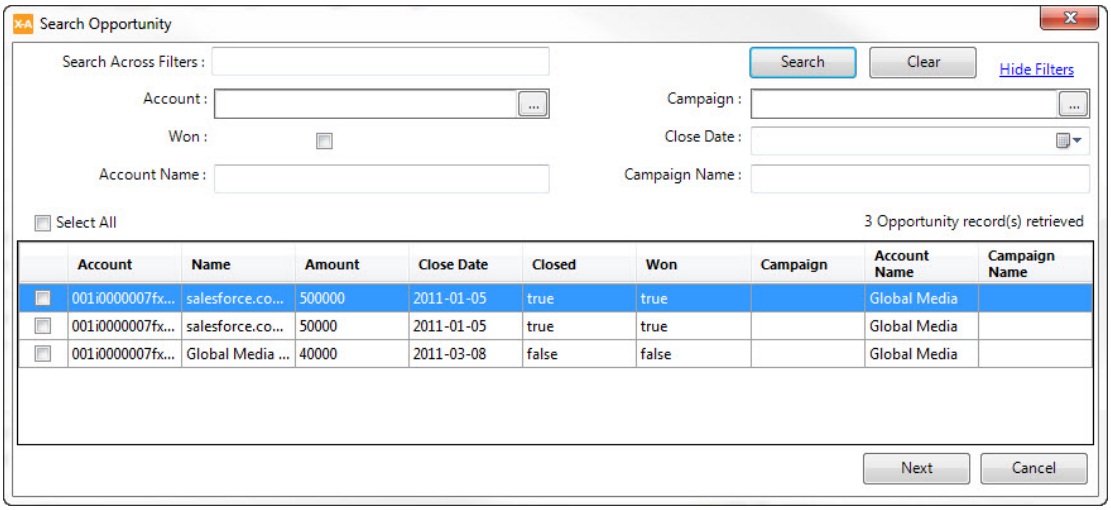
If you want to see all Opportunities for only the Toronto office of Global Media, you can type search text in both search fields.

To search all available search fields at one time, you would type your search text in the Search Across Filters text box.

For this example, we'll search for all opportunities in Toronto, Canada.



Double-click the search result you want to retrieve data from, or select a row and click **Next**. The Search Opportunities window opens. (Opportunities is the child object in our app.)



Again, all of the search fields available on this page are the fields that were configured to be search fields for the Opportunities object when the app was created. You can use these search fields to narrow down your results.

In our case, we want to retrieve all three of the opportunities that were retrieved, so we'll click the **Select All** check box and click **Next**.

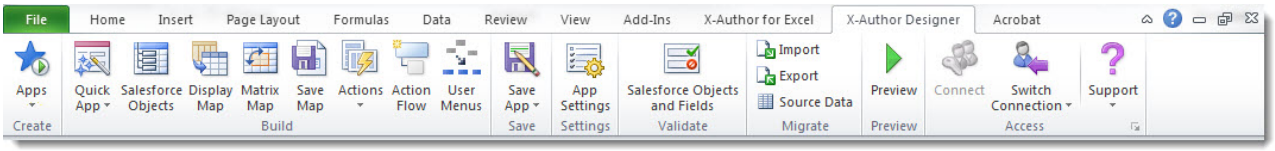
The search window closes and our data is retrieve and added to our Excel worksheet.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3	Account ID	001i0000007fxwDAAQ								
4	Account Name	Global Media								
5	Billing City	Toronto								
6	Billing Country	Canada								
7	Last Activity									
8										
9	Opportunity ID	Account	Name	Amount	Close Date	Closed	Won	Campaign	Account Name	Campaign Name
11	006i0000003Y7gHAAS	001i0000007fxwDAAQ	salesforce.com - 5000 Widgets	500,000.00	1/5/2011	TRUE	TRUE		Global Media	
12	006i0000003Y7gIAAS	001i0000007fxwDAAQ	salesforce.com - 500 Widgets	50,000.00	1/5/2011	TRUE	TRUE		Global Media	
13	006i0000003Y7gJAAS	001i0000007fxwDAAQ	Global Media - 400 Widgets	40,000.00	3/8/2011	FALSE	FALSE		Global Media	
14										
15										
16										
17										

Sheet1 Sheet2 Sheet3

Designing Apps

All of the controls you need to use X-Author Designer are located in the X-Author Designer ribbon menu in Microsoft Excel.



Before you start designing apps, you should **familiarize yourself with the X-Author for Excel components** that are accessed through the X-Author Designer ribbon menu. **Click the link in the first column** to learn more about the component.

Component	Description
Apps	Click this button to create a new app or open an existing app. (Click the **Quick App** button to instead create a simple single-object app or a parent-child-type app.)
Salesforce Objects	Click this button to specify the Salesforce objects and fields you will use in the app. (This section only covers single object and parent-child objects.)
Display Map	Click this button to define relationships between Salesforce fields and Excel cells for data that will be **retrieved from** Salesforce.
Save Map	Click this button to define relationships between Salesforce fields and Excel cells for data that will be **saved to** Salesforce.
Actions	Click this button to assign actions to your app. Actions can be used to retrieve, display, and save data. They can also automate tasks in Excel worksheets and call Salesforce methods.
Action Flow	Click this button to define a sequence of one more actions that will be executed when end users click buttons on the app's ribbon menu.

Component	Description
User Menus	Click this button to create the ribbon menus that your end users will see in the X-Author for Excel ribbon in Excel.
Matrix Maps	Click this button to define matrix maps in your Excel spreadsheets. Matrix maps allow a Salesforce record to be represented by one or more cells by mapping the rows and columns of a matrix and the data contained at the intersection of the row and column.
Save Application	Click this button to save or clone your application. See also, Cloning an Application.
Application Settings	Click this button to restrict runtime functions (such as saving, printing, rich-text editing) and password-protect worksheets.
Import and Export	Click these buttons to migrate X-Author for Excel apps from one Salesforce org to another. **Note:** All Salesforce objects used in the source org must also exist in the destination org.
Source Data	Click this button to easily migrate data from one Salesforce org to another using X-Author for Excel.
Connect	Click this button to connect to your Salesforce org, to switch to another org, or to revoke a previously configured connection.

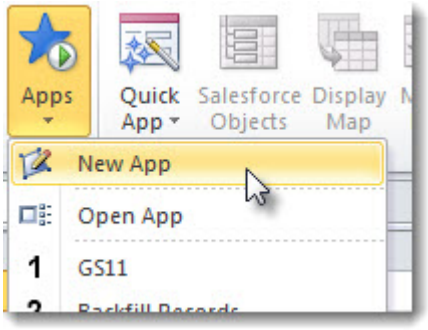
Creating Apps

You can create apps in two different ways. The quickest and simplest way is to use the **Quick App** option. However, that method allows you to create only simple list- and parent-child-type apps. To create a more customizable app, you will need to use the **full-featured New Application** option.

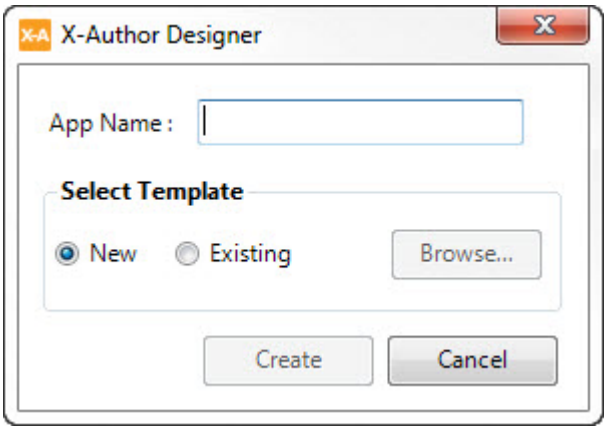
To learn how to create a Quick App, refer to Creating Quick Apps**.**

To create a new full-featured app

1. In the ribbon menu, click the **Apps** button and select **New App**.



2. Type the app name in the App Name field. This name will display as the window title in the Excel worksheet at runtime.



> Give your app a name that is concise but also summarizes the basic purpose of the app.

3. To create a new app from scratch, click the **New** radio button.
4. To create an app from an existing template, click the **Existing** radio button and then click **Browse** to select the template file.
5. Click **Create**.
6. in the ribbon menu, click **Save App**.

You've created an app, but it doesn't do anything...yet. Go to the next pageto learn how to: add Salesforce objects and fields to your apps; define Display, Save, and Matrix maps; create actions and action flows, and design your runtime user menus.

Creating Quick Apps

Prerequisite

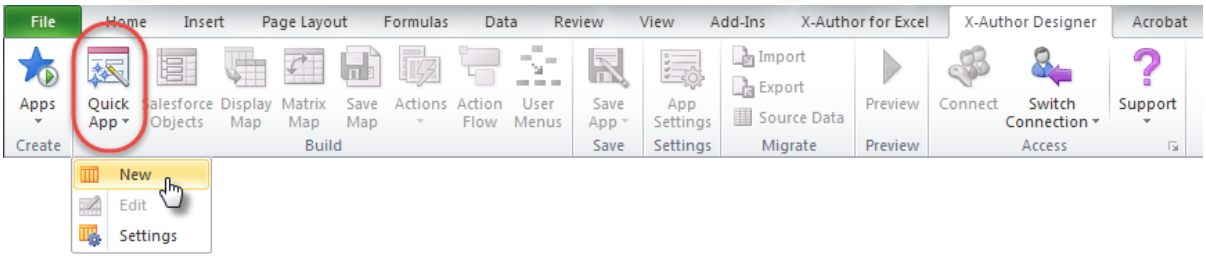
Before you can create an app, you must connect to a Salesforce org that is configured for X-Author for Excel. For information on configuring and connecting to Salesforce, refer to Setting Salesforce Permissions and Connecting to Salesforce from Excel.

Click the links below to see how to create single-object and parent-child Quick Apps, and to learn how to change your app's color scheme.

To create a Single Object App

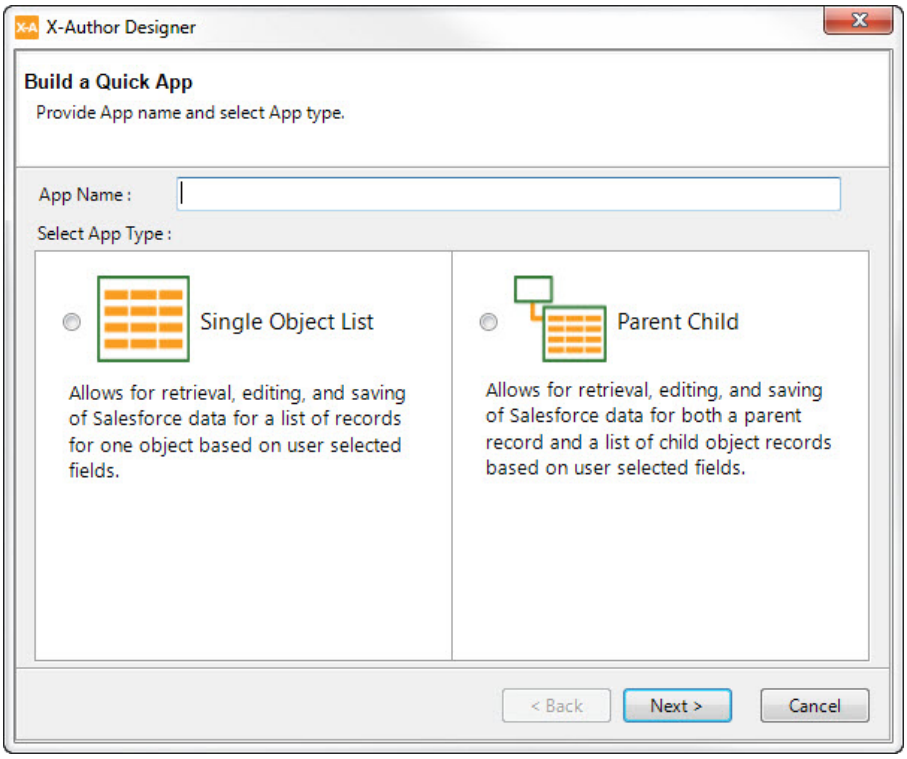
To create a Single Object App

1. Click **Quick App** and select **New**.



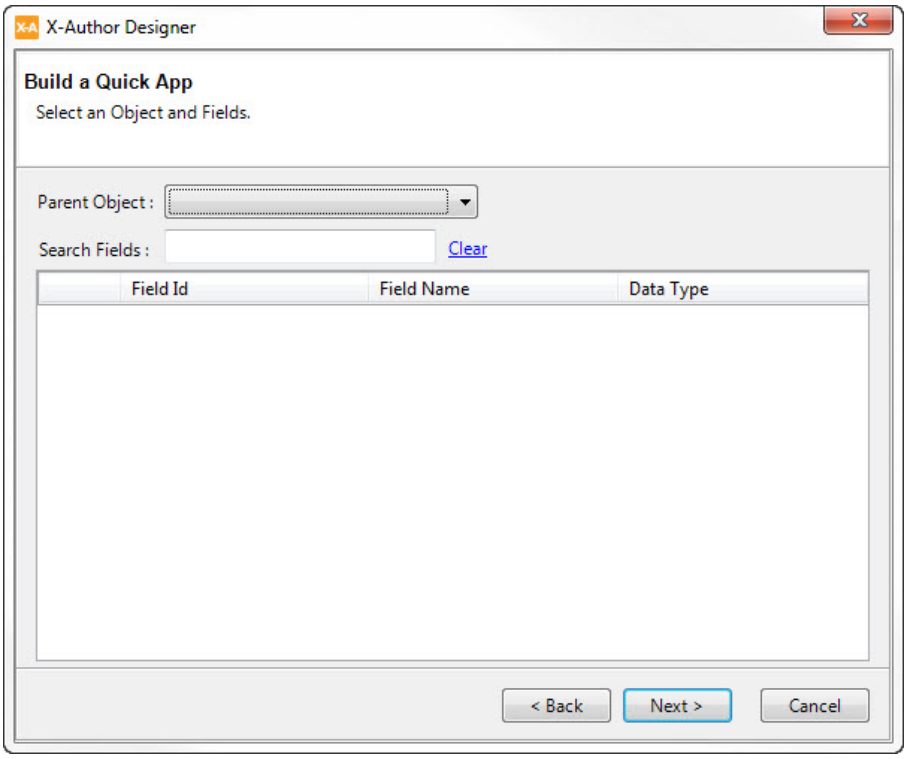
   The Build a Quick App window opens.

2. Type the application name into the App Name field.

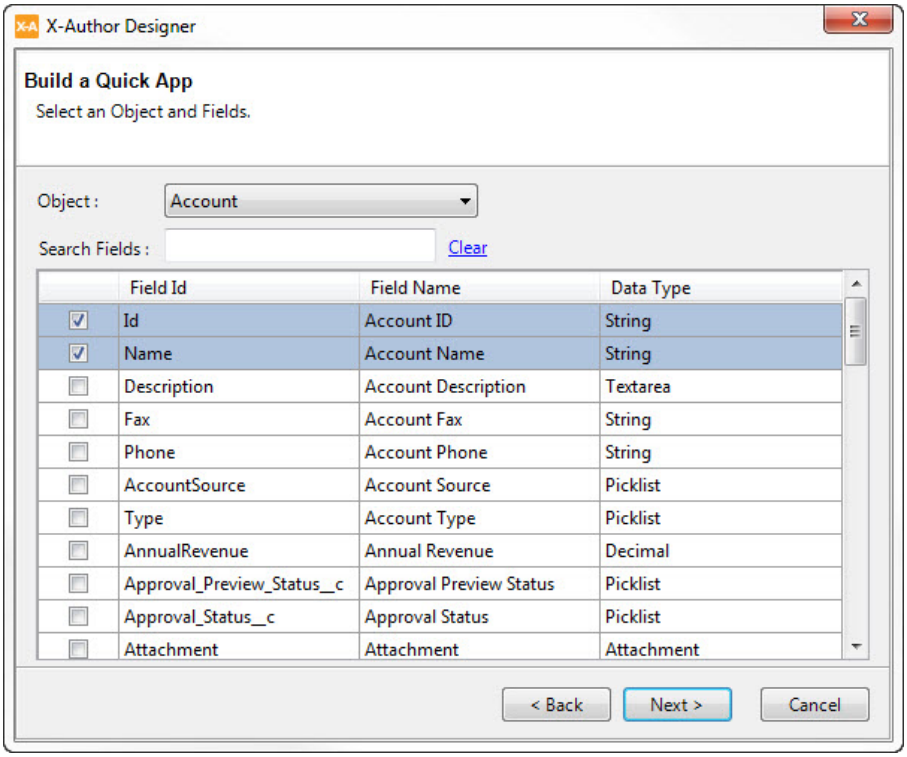


Give your app a name that is concise but also summarizes the basic purpose of the app.

3. Click the **Single Object List** radio button and then click **Next**. The Build a Quick App page opens.



4. Click the **Object** drop-down menu and select a Salesforce object. The Select Fields section will display the fields associated with the object.

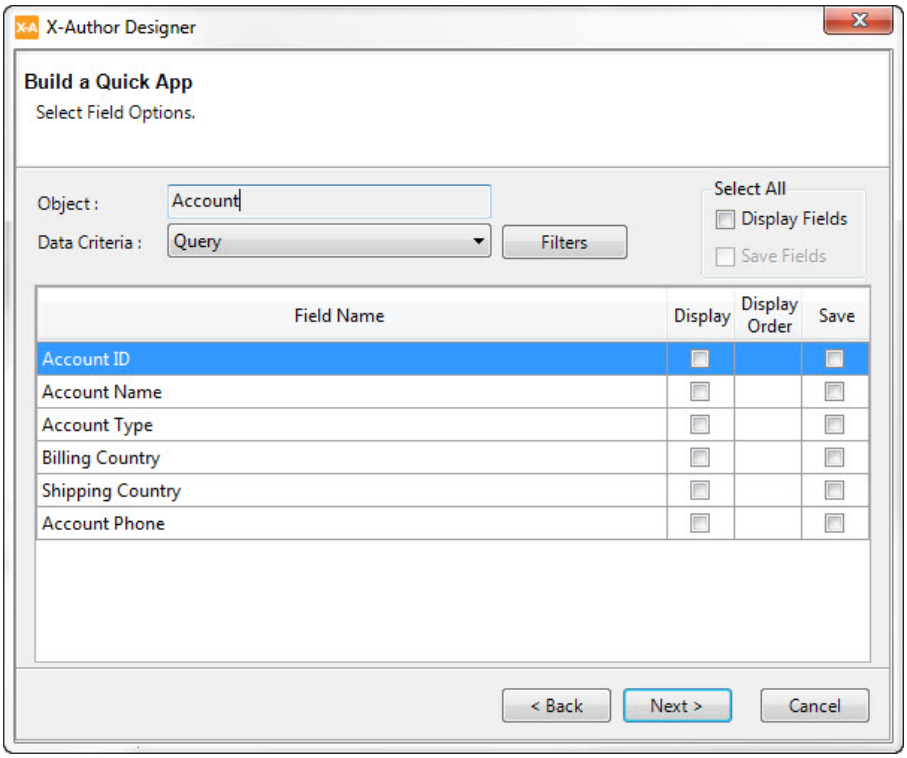


5. Click the check box next to the fields you want to include in your app. (Notice that the object's two primary fields are selected by default.)

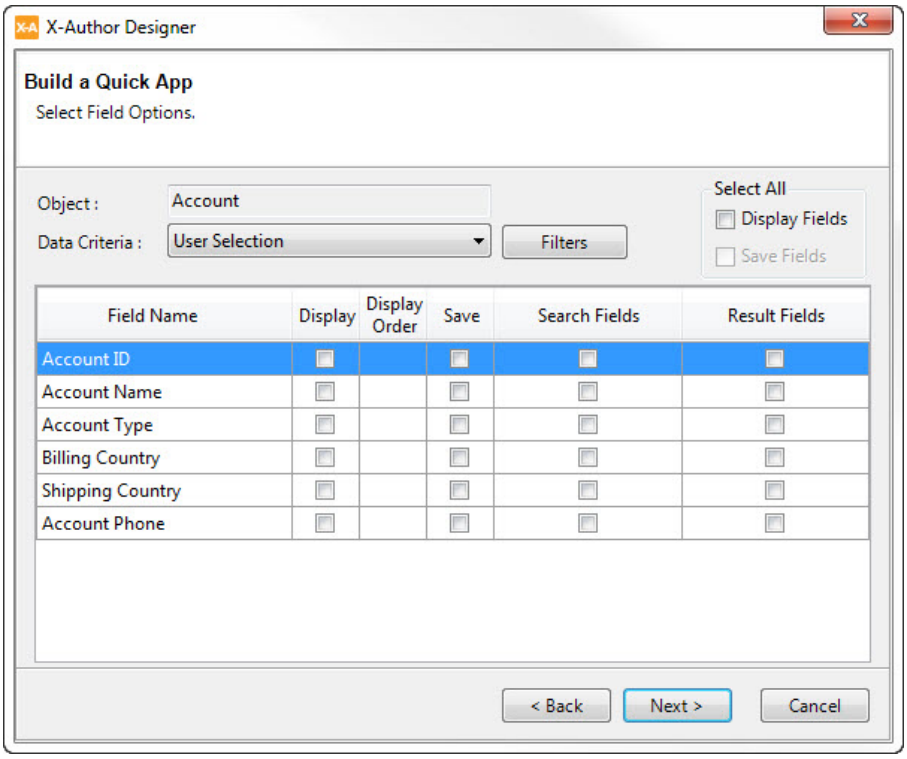
> You can search for Salesforce fields using the Search Fields search box. Click **Clear** to clear search terms and view the full list.

6. Click **Next.** The Select Field Options window opens, displaying the fields that you selected in the previous step.

   As shown in the following images, the displayed columns will vary based on the Data Criteria you select.



Selecting Field Options with a *Query* Data Criteria



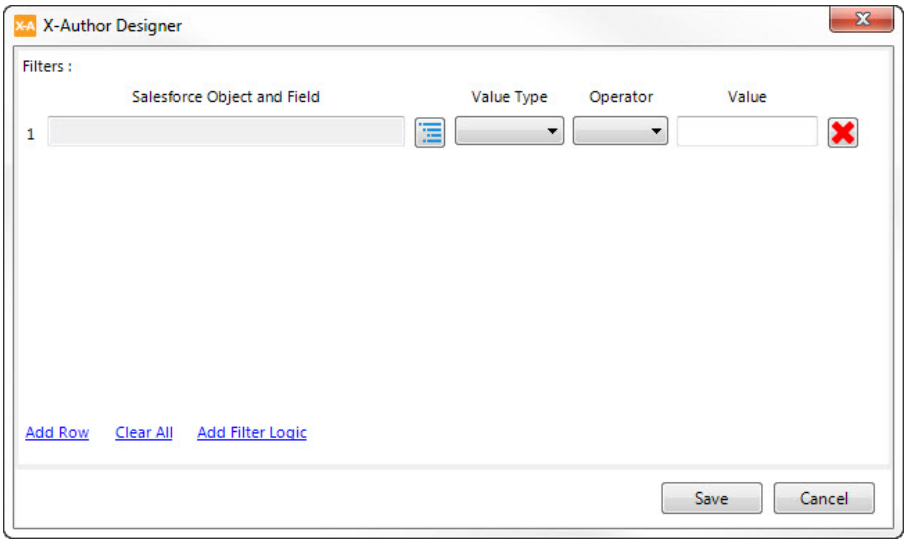
Selecting Field Options with a *User Selection* Data Criteria

7. Select from the options in the following table.

Column	Description
Data Criteria	Select the method you would like to use to retrieve data. **Query**—Select this method to have the query automatically run, based on your app's predetermined search filters, and retrieve the data without user intervention. **User Selection**—Select this method to allow users to apply search filter criteria before retrieving data.
Select All	
Display Fields	Click this check box to include all object fields in the Display Map.

Column	Description
Save Fields	Click this check box to include all object fields in the Save Map. Note: This check box is inactive unless Display Fields is checked.
Grid Columns	
Display	Click this check box to include this field's data in the the data retrieved from Salesforce and displayed in the Excel workbook.
Order	This represents the order, left to right, that fields will be displayed in. Click in the cell to edit the order.
Save	Click this check box to allow users to change the data in these cells in the Excel workbook and then save those changes to Salesforce. **Note:** Fields that are not Display fields cannot be Save fields.
Search Fields	Click this check box to add a run-time search filter for this field. These filters will be used to limit the type of records returned when the run-time user retrieves data. For example, might make Billing Country a Search field so that you can retrieve data from only those accounts who have a billing address in Canada. **This field appears only when the Data Criteria is set to User Selection.** **Note:** You must select at least one field, and ID fields are **not** searchable.
Result Fields	Click this check box to display this field in the search results when retrieving and filtering data from Salesforce. **This field appears only when the Data Criteria is set to User Selection.** **Note:** You must select at least one field and ID fields **cannot** be used.

8. To add a filter to the data, click **Filters**.



a. Complete the fields described in the following table to build your filter.

Field	Description
Salesforce Object and Field	Click to select the object and field that you want the filter to apply to. **Note:** You cannot type the field name in the text field.
Value Type	Select the value type from the drop-down menu. **The value types change based on the Salesforce field or object selected.** • **Input**—Select this value to allow users to type in a value to search for. For example, if the field is Account ID, you might choose to let them type or paste the account ID in the filter at runtime. • **Static**—Select this value to prevent users from changing the Value field at runtime. • **System Variable**—Select this value to specify a system variable, such a CurrentUser or CurrentDate, as the filter value. • **User Input**— Select this value to allow users to change the Value field at runtime.
Operator	Select a relational operator from the drop-down menu. • **Equals**—The field value is equal to the specified value. • **Not equal to**—The field value is **not** equal to the specified value. • **Starts with**—The field value begins with the specified value.

Field	Description
	• **Contains**—The field value contains the specified value string. • **Not contains**—The field does not contain the specified value string. • **In**—The field value is in the specifed list of hard-return-separated values. • **Not In**—The field value is in the specifed list of hard-return-separated values.
Value	Type the comparison value in this field or select the value from the drop-down menu. **Your options will vary depending on the type of field you are filtering on.**
	Click this button to delete a filter row.
Add Row	Click this button to add a new filter row.
Clear All	Click this button to remove all filter rows.
Add Filter Logic	Click this button to display the Filter Logic field and define the Boolean logic used to relate filters. For example, you might want display results only when **all** of your filter conditions are met or you might want to display results when **any** filter conditions are met. You can use filter logic to do this. Filter logic in X-Author for Excel works the same as it does in Salesforce.com. Learn more .

b. Click **Save**.

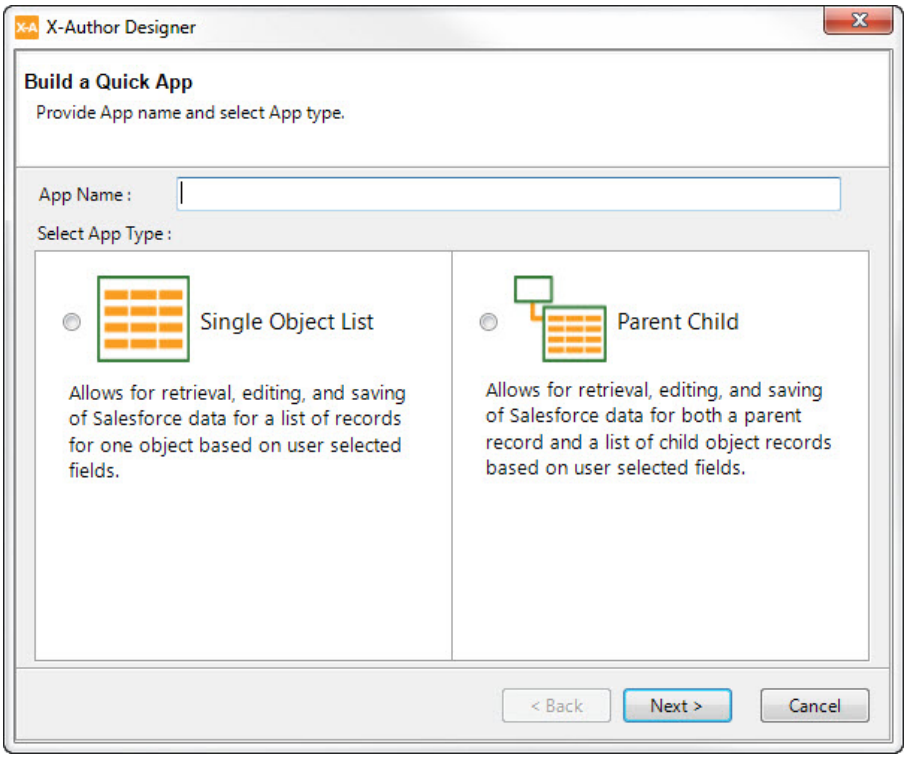
9. Click **Next**.

An App has been created with all design functions complete.You can now Preview the app, run the app, or use X-Author Designer to make additional modifications.

To create a Parent Child app

To create a Parent Child app

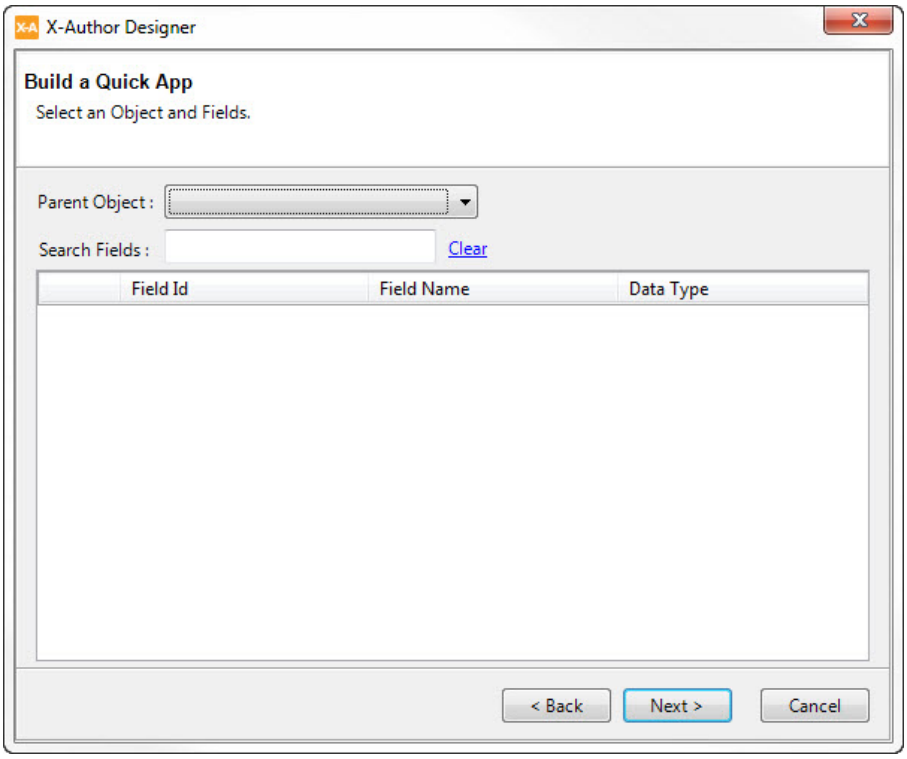
1. Click **Apps** and then **Quick App**. The Build a Quick App window opens.



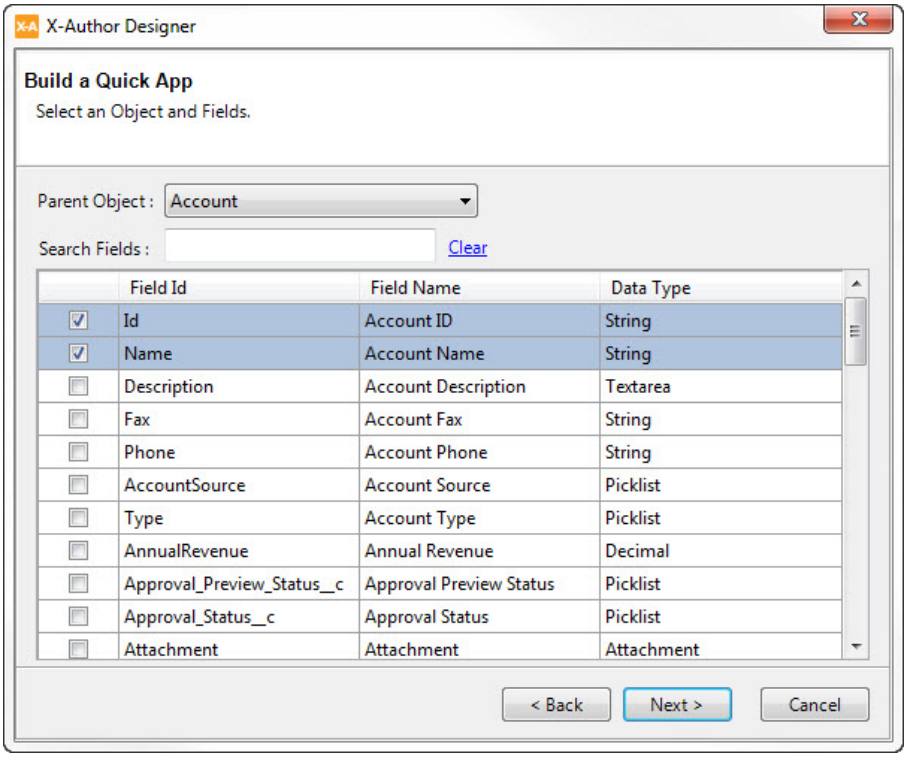
2. Type the application name into the **App Name** field.

> Give your app a name that is concise but also summarizes the basic purpose of the app.

3. Click the **Parent Child** radio button and then click **Next**.



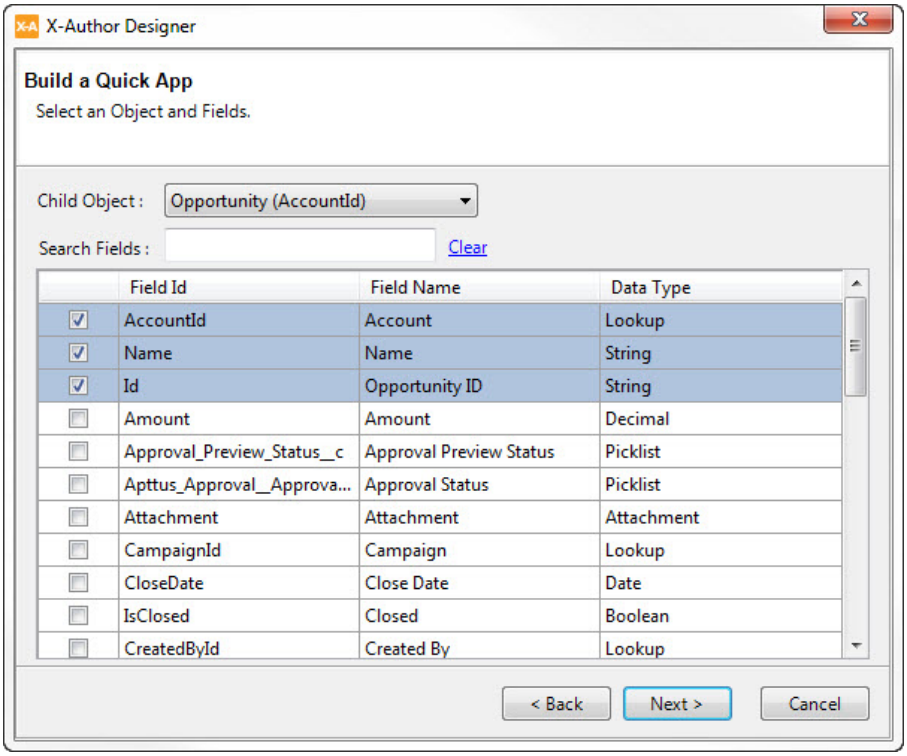
4. Click the **Parent Object** drop-down menu and select a master Salesforce object. The Select Fields section will display the fields associated with the object.



5. Click the check box next to the fields you want to include in your app.

> You can search for Salesforce fields using the Search Fields search box. Click **Clear** to clear search terms and view the full list.

6. Click **Next.**



7. Click the **Child Object** drop-down menu and select a child object. The fields associated with the object are displayed.
8. Select from the options in the following table.

Column	Description
Data Criteria	

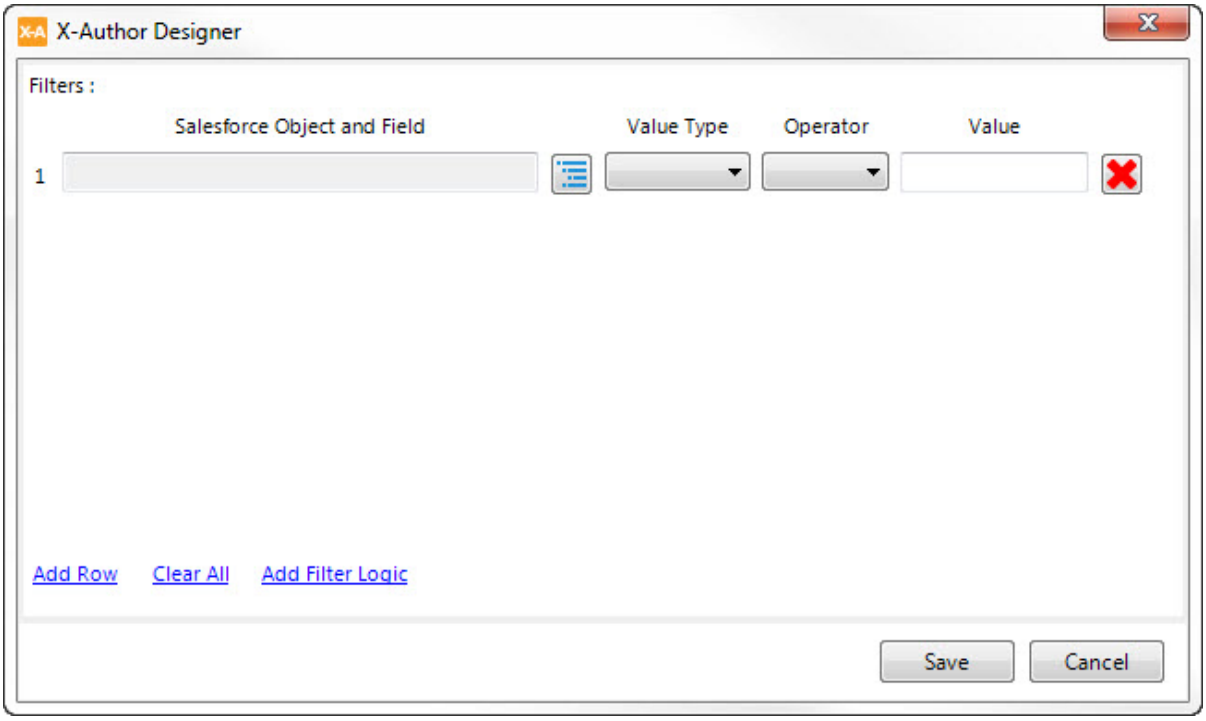
Column	Description
	Select the method you would like to use to retrieve data. **Query**—Select this method to have the query automatically run, based on your app's predetermined search filters, and retrieve the data without user intervention. **User Selection**—Select this method to allow users to apply search filter criteria before retrieving data.
Select All	
Display Fields	Click this check box to include all object fields in the Display Map.
Save Fields	Click this check box to include all object fields in the Save Map. Note: This check box is inactive unless Display Fields is checked.
Grid Columns	
Display	Click this check box to include this field's data in the the data retrieved from Salesforce and displayed in the Excel workbook.
Order	This represents the order, left to right, that fields will be displayed in. Click in the cell to edit the order.
Save	Click this check box to allow users to change the data in these cells in the Excel workbook and then save those changes to Salesforce. **Note:** Fields that are not Display fields cannot be Save fields.
Search Fields	Click this check box to add a run-time search filter for this field. These filters will be used to limit the type of records returned when the run-time user retrieves data. For example, might make Billing Country a Search field so that you can retrieve data from only those accounts who have a billing address in Canada. **This field appears only when the Data Criteria is set to User Selection.** **Note:** You must select at least one field, and ID fields are **not** searchable.

Column	Description
Result Fields	Click this check box to display this field in the search results when retrieving and filtering data from Salesforce. **This field appears only when the Data Criteria is set to User Selection.** **Note:** You must select at least one field and ID fields **cannot** be used.

Did you know...?

By specifying fields to be used as Search Fields, you just created filters. In the upcoming steps, you will see how to add other filters based on the fields that weren't selected as Search Fields.

9. To add a filter to the data, click **Apply Filter**. Any fields that you chose to be Search fields in the previous step will already appear here as filters.



a. Complete the fields described in the following table to build your filter.

Field	Description

Salesforce Object and Field	Click to select the object and field that you want the filter to apply to. **Note:** You cannot type the field name in the text field.
Value Type	Select the value type from the drop-down menu. **The value types change based on the Salesforce field or object selected.** • **Input**—Select this value to allow users to type in a value to search for. For example, if the field is Account ID, you might choose to let them type or paste the account ID in the filter at runtime. • **Static**—Select this value to prevent users from changing the Value field at runtime. • **System Variable**—Select this value to specify a system variable, such a CurrentUser or CurrentDate, as the filter value. • **User Input**— Select this value to allow users to change the Value field at runtime.
Operator	Select a relational operator from the drop-down menu. • **Equals**—The field value is equal to the specified value. • **Not equal to**—The field value is **not** equal to the specified value. • **Starts with**—The field value begins with the specified value. • **Contains**—The field value contains the specified value string. • **Not contains**—The field does not contain the specified value string. • **In**—The field value is in the specifed list of hard-return-separated values. • **Not In**—The field value is in the specifed list of hard-return-separated values.
Value	Type the comparison value in this field or select the value from the drop-down menu. **Your options will vary depending on the type of field you are filtering on.**
	Click this button to delete a filter row.
Add Row	Click this button to add a new filter row.

Field	Description
Clear All	Click this button to remove all filter rows.
Add Filter Logic	Click this button to display the Filter Logic field and define the Boolean logic used to relate filters. For example, you might want display results only when **all** of your filter conditions are met or you might want to display results when **any** filter conditions are met. You can use filter logic to do this. Filter logic in X-Author for Excel works the same as it does in Salesforce.com. Learn more .

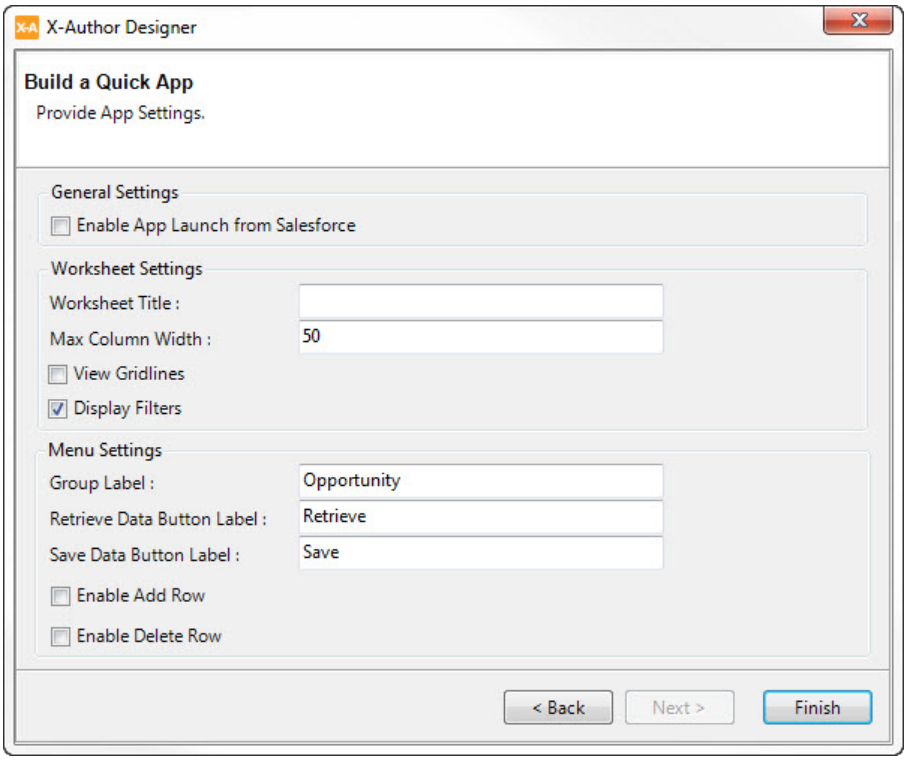
b. Click **Save**.

10. Click **Next**.
11. Select from the options in the following table.

Field	Description
Salesforce Object and Field	Click  to select the object and field that you want the filter to apply to. **Note:** You cannot type the field name in the text field.
Value Type	Select the value type from the drop-down menu. **The value types change based on the Salesforce field or object selected.** • **Input**—Select this value to allow users to type in a value to search for. For example, if the field is Account ID, you might choose to let them type or paste the account ID in the filter at runtime. • **Static**—Select this value to prevent users from changing the Value field at runtime. • **System Variable**—Select this value to specify a system variable, such a CurrentUser or CurrentDate, as the filter value. • **User Input**— Select this value to allow users to change the Value field at runtime.
Operator	

Field	Description
	Select a relational operator from the drop-down menu. • **Equals**—The field value is equal to the specified value. • **Not equal to**—The field value is **not** equal to the specified value. • **Starts with**—The field value begins with the specified value. • **Contains**—The field value contains the specified value string. • **Not contains**—The field does not contain the specified value string. • **In**—The field value is in the specifed list of hard-return-separated values. • **Not In**—The field value is in the specifed list of hard-return-separated values.
Value	Type the comparison value in this field or select the value from the drop-down menu. **Your options will vary depending on the type of field you are filtering on.**
	Click this button to delete a filter row.
Add Row	Click this button to add a new filter row.
Clear All	Click this button to remove all filter rows.
Add Filter Logic	Click this button to display the Filter Logic field and define the Boolean logic used to relate filters. For example, you might want display results only when **all** of your filter conditions are met or you might want to display results when **any** filter conditions are met. You can use filter logic to do this. Filter logic in X-Author for Excel works the same as it does in Salesforce.com. Learn more .

12. To add a filter to the data, click **Apply Filter** and follow the same process you used to create a filter for the parent object.
13. Click **Next**. The Provide App Settings window opens.



14. Complete the fields described in the following table.

Field	Description
Enable App Launch from Salesforce	Select this check box to allow users to open this X-Author app from within the Salesforce application (as well as from within Excel).
Worksheet Title	Type the title of the Excel worksheet. The title appears in the top row of the worksheet.
Max Column Width	Type the maximum width of an Excel column (in characters).
View Gridlines	Select this check box to display the gridlines in the Excel worksheet.
Display Filters	Clear this check box if you do not want the filters that you created to appear in the Excel worksheet.

Field	Description
Group Label	Type the ribbon menu group name. This value appears beneath the buttons in the X-Author for Excel ribbon tab.
Retrieve Data Button Label	Type the text you want to appear on the ribbon menu button used for data retrieval.
Save Data Button Label	Type the text you want to appear on the ribbon menu button used for saving data to the Salesforce database.
Enable Add Row	Select this check box to allow users to add rows to the Excel worksheet. This option could be used to add records to the workbook so those records can be uploaded to Salesforce.
Enable Delete Row	Select this check box to allow users to delete rows from the Excel worksheet.

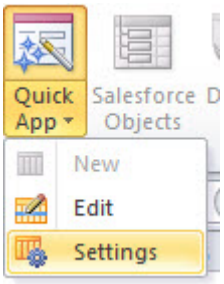
15. Click **Finish**.

An App has been created with all design functions complete.You can now Preview the app, run the app, or use X-Author Designer to make additional modifications.

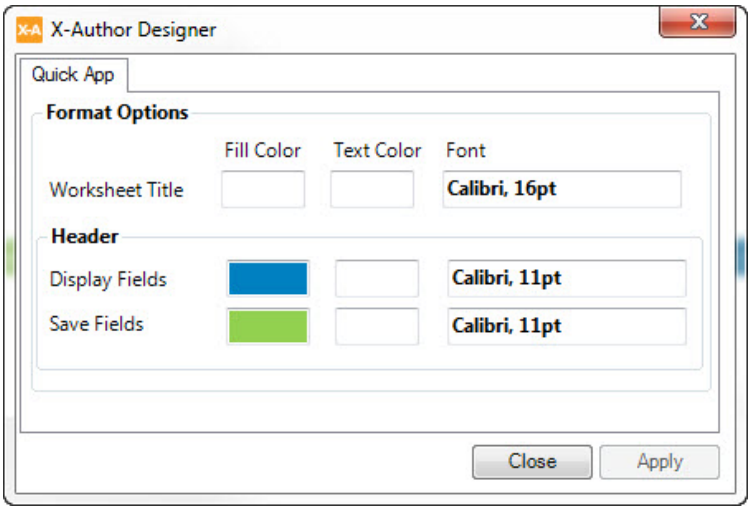
To customize the color scheme

To customize the color scheme

In the ribbon menu, click Quick App and select Settings.



The Format Options window opens.



The following table describes the fields in the Format Options window.

Field	Description
Worksheet Title	
Fill Color	Click this field to open the color picker and specify the background color for the worksheet title.
Text Color	Click this field to open the color picker and specify the text color for the worksheet title.
Font	Click this field to open the Font window and specify font properties for the worksheet title.
Header Display Fields	
Fill Color	Click this field to open the color picker and specify the background color for the Display fields.

Field	Description
Text Color	Click this field to open the color picker and specify the text color for the the Display fields.
Font	Click this field to open the Font window and specify font properties for the the Display fields.
Header Save Fields	
Fill Color	Click this field to open the color picker and specify the background color for the Save fields.
Text Color	Click this field to open the color picker and specify the text color for the Save fields.
Font	Click this field to open the Font window and specify font properties for the Save fields.

Salesforce Objects

Use the *Salesforce Objects* button to select the Salesforce objects and fields to be used in an app.

This section covers only **single object and parent-child objects.**

To add objects and fields

Object Behavior

All objects used in an app are assigned a behavior, which determines their availability in maps and actions. Object behavior can be changed at any time by right-clicking on any object once it has been added to *Selected Objects*. There are two behaviors:

- Individual: Work on an individual record at a time for that object.
- List : Work on more than one record simultaneously for that object.

Default Object Behavior

When parent-child objects are selected, the parent will default to *Individual* because there will just be one header record until you get to the lowest child level. The lowest child node will default to *List* because it is a related list of its parent.

When a single object is selected, it will default to List.

You can change either at any time by right-clicking on an object once it has been added to *Selected Objects*.

Object Combinations

- Any number of objects can be selected.
- The same object can be selected multiple times in different hierarchies if it needs to be used in different ways.
- A combination of related and unrelated objects can be selected.

Changing Objects

Removing an object will invalidate any maps and actions in the app. You must remove those links manually before you remove the object. Generally, if you have selected the incorrect objects and have already created maps and actions, you should start over. If it is just unrelated objects that need to be changed, then you can remove that object after you correct the maps and actions. Then you can add in a new object.

Selecting Fields

- Select as many fields as necessary for each object.
- The following fields are automatically selected:
    - ID
    - Name, if you have an existing name.
    - Look-up, for a parent object that is selected as part of the object hierarchy.

  All other fields must be manually selected. The Search Fields box provides a type-ahead that searches for fields across both the field ID and field Name columns. The middle column displays the field names and is sorted alphabetically. (You can also sort the fields in data type order by clicking at the top of the data type column.)

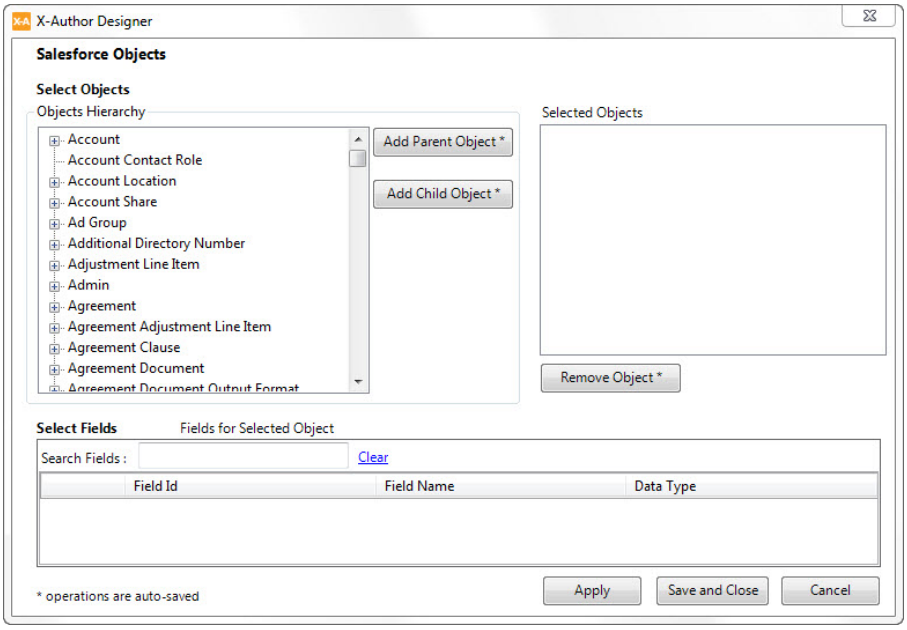
- You must click Apply to save your field selections as you move between objects. When you are done with all objects, click Save and Close.
- When a picklist field is selected, all picklist values will automatically be retrieved during runtime. Regular, dependent and multi-select picklists are supported.

Lookup Fields

- Selecting a lookup field will automatically retrieve the lookup name from the parent object for use in actions and maps. Note, the name field is not displayed here. For example, if you select Account Id, then Account Id (name) will be automatically selected and available for later use.
- If all you require from the lookup object is the lookup name, then you do not need to select the lookup object as its own object. If any other fields are required from the lookup object, you will need to add that object to *Selected Objects*.

To add objects and fields

1. In the ribbon menu, click the **Salesforce Objects** button. The Select Objects window opens.



2. From the Objects Hierarchy section, select an object or drill down to other objects by clicking the **+** next to the object name.
3. To select one object, click **Add Parent Object** to move the object to Selected Objects.

4. To add one or more nested objects, click **Add Child Objects** to add all child objects below the parent object.
5. Object behavior can be changed by right-clicking on an object in Selected Objects.
6. In the Select Fields section, click the check box next to at least one field for a selected object to add fields to your app.

> Primary fields are selected by default and cannot be deselected.

7. Click **Apply**.
8. After you have selected all of the objects and fields you need for your app, click Save and Close.

The Salesforce objects and fields for the app have been selected and saved and are available to be used in maps and actions.You can now create maps or actions.

Display Maps

- To create a Display Map
- To edit an existing Display Map
- To delete an existing Display Map
- To add Individual Object fields to a Display Map
- To add List Object fields to a Display Map

Display Maps are accessed from the *Display Map* button in the ribbon menu. A Display Map is used to link Salesforce fields to Excel cells for data that will be retrieved from Salesforce. When data is retrieved from Salesforce using one of the X-Author retrieve actions, it is temporarily stored in Excel memory before being placed into the worksheet. X-Author then uses a Display Map to place the data in the correct Excel cells.

Each field to be linked in a Display Map is presented in the Designer based on one of the two object behavior types (Individual and List) assigned to each object and, by default, its selected fields in Salesforce Objects.

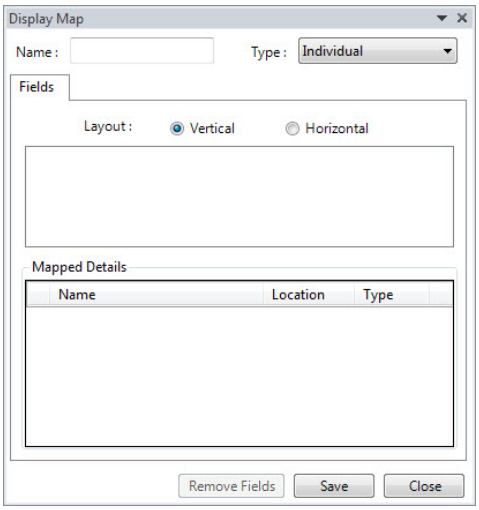
The following table describes the display map options.

Unlimited Display Maps	There is no limit to the number of Display Maps in a single app.
Unlimited Objects	A Display Map can include fields from multiple objects.
Multiple Worksheets	A Display Map can cross worksheets, but not workbooks.
Changing Layout	You can insert columns or rows after you have created a Display Map. When you save the app, the cell locations in a Display Map are adjusted based on the new layout. If you need to move a field, please see *Moving and Removing Fields* .
Template Presentation	To prepare the template for the end user once a field has been mapped, place your cursor in a cell and delete the value in the brackets using the *Delete* key. Do not delete the row.
Removing Fields	To remove selected fields from Excel, select the fields from the *Mapped Details* section in the bottom right hand window of the Display Map and click Remove Fields. If you remove a field that is already part of a Save Map, a warning message is displayed that the field will be removed from the Save Map as well. Do not delete a field from the worksheet in any other way, otherwise the name range that governs that field will not get properly removed.

To create a Display Map

You must have at least one object selected in *Salesforce Objects*.

1. In the ribbon menu, click **Display Map** and click **Create**. The Display Map panel opens in Excel.



2. In the Name field, type the display map name.
3. Select an object type from the Type drop-down menu to see the fields for those objects.

> Objects are listed based on the object behavior from Salesforce Objects: Individual or List. Individual fields are displayed by default. You can switch to List fields from the Type list.

4. Add a field to the Display Map by dragging it from the Display Map panel and dropping it on the worksheet.
5. Click **Save** and **Close**.
6. Save the Application.

You have created a Display Map.

To edit an existing Display Map

1. Click Display Map.
2. From the list of existing Display Maps, select one and click Edit.
3. Make the desired changes and click Save and Close.

The updated Display Map is saved.

To delete an existing Display Map

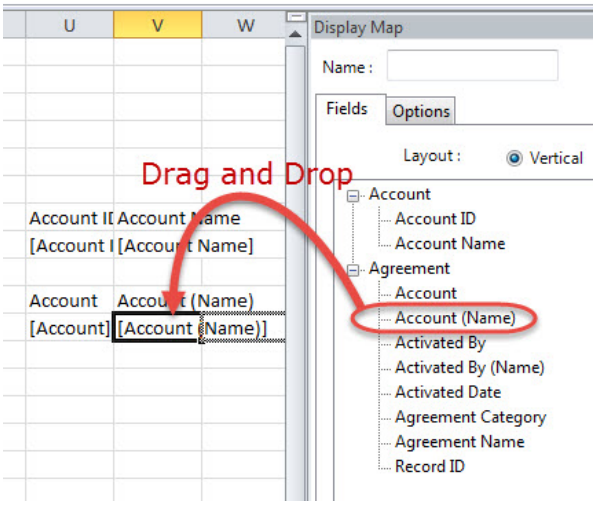
> You cannot delete display maps that are used in a Display action.

1. Click the Display Map button.
2. From the list of existing display maps, select a display map and click Delete.

The Display Map is deleted.

To add Individual Object fields to a Display Map

1. Set the Type field to Individual. All Individual objects and their fields will be displayed.
2. Drag the field from the Display Maps panel and drop it in a cell on the worksheet.
   Note: You can place the field anywhere on any worksheet except in the same row that contains fields from a List object.



   Note: Based on the hierarchical navigation, you can also select any field from a related list object and drag it.

The mapped field is displayed in brackets and the field label will display on the left. The mapped field/cell link is displayed in Mapped Details.

To add List Object fields to a Display Map

1. Select **List** from the Type picklist.
2. Click the **Vertical** radio button. (Horizontal layouts are covered later in this section.)
3. Drag the field from the Display Map panel and drop it in a cell on the worksheet.

When fields from List objects are mapped, they form a list range. Based on the hierarchical navigation, you can select any field from a related list object and place it in the same row. They do not need to be placed adjacent to each other, but there must be a start and end cell that represent the list range. For example, you could have mapped fields in A5, C5, and F5 with formulas in B5, D5, and E5.The mapped field is displayed in brackets and the field label will display above. The mapped field/cell link is displayed in Mapped Details. For more information about List objects and ranges, refer to List Ranges.

Hierarchical Navigation

There are various areas within the Designer where hierarchical navigation is used. Fields can be added to the Display Map from either the object listed in the window or from a related object in an object hierarchy. The objects and the specific fields required for the Display Map must have all been selected in Salesforce Objects.

To add a field from the listed object, click on a field, hold the left mouse button down, and drag it onto the worksheet. To add a field from a related object, click on a + key (designates a related object is available) and navigate to the object

List Ranges

This section contains the rules and guidelines that apply to the use of list ranges.

Must include the record id if you are planning to save data	• If you are planning to save back to Salesforce any data that has been retrieved into a List Range, you must include the record ID somewhere in the range. The best place to put this field is at the end of the range but it can be placed anywhere. • The record ID is used to make sure that the Excel rows and Salesforce records remain connected should the end user sort data in the range.
Formulas in a List Range	• Formulas can be added into either mapped or non-mapped cells in a List Range.

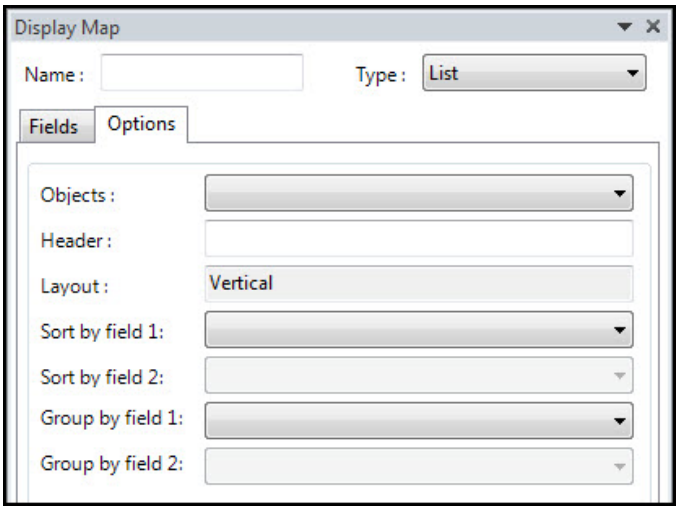
	• Formulas are inserted into the List Range cells as each record is retrieved into the worksheet via the Display Map. • Formulas in cells where Salesforce data is being retrieved and displayed will be overwritten by the Salesforce data but are still very useful if you allow users to add rows. This is covered in *Adding Rows to Worksheets*.
List Rows are Inserted	• As each List Salesforce record is retrieved and displayed, an Excel row is inserted in the workbook. • Anything you have placed below or alongside a List Range (for example: a total formula below the range or a pivot table), is pushed down accordingly.
Fields from related objects	• Fields selected from related objects in the hierarchy (see Hierarchy Navigation above) can also be included on the same row in the List Range.
List Ranges for different objects	• These can be created below other ranges or on other worksheets.

List Range Sorting and Grouping

For List ranges, you can set some basic sorting and other settings in the Options tab. The native Excel sorting and grouping options are more flexible in many cases.

To sort and group List Ranges

1. Select **List** from the Type field.
2. Click the **Options** tab.



3. Type a Header Description to display above the label row of the list range.
4. In Sort by Field 1 and Sort by Field 2, select an individual column to sort by.
5. In Group by Field 1 and Group by Field 2, select a column in the repeating range to group by. These fields should be at the leftmost side of the list range.
6. In Group Spacing, select a value to specify the number of rows that should separate the groups.
7. Click Save.

Sorting and Grouping Options have been selected and saved.

Vertical and Horizontal Layouts

Records in a List object can be displayed either vertically or horizontally. In vertical layouts, records are presented as rows with each column being a field. For horizontal layouts, each record is a column and the rows are represented by the fields. The steps are the same a you would use to create a Display Map, but select the Horizontal radio button instead of the Vertical radio button.

Horizontal Layout Restrictions

Records can be retrieved and updated but you cannot add or delete records.

Field Lookups

Field lookups allow designers to create offline and online field-lookups for runtime users. For example, when a user is adding an opportunity, they can lookup the account name. You can create offline lookups that display either a single column or multiple columns.

Prerequisite

Before creating a field lookup, you must have an existing field mapping in a Display or Save Map.

To create an online lookup

A Search and Select Action on the object that corresponds to the lookup value field must exist. For example, if you need an account lookup, there must be a Search and Select Action on the Account object.

1. Click Display Map.
2. Select a display map and click Edit. The Display Map panel opens.
3. In the Mapped Details section of the Display Map pane, click  to open lookup properties for that field.
4. Click the Search And Select Action tab.
5. Select the applicable Search and Select Action.
6. To set Online Lookup as the default lookup method, select the Is Active check box. Either Offline or Online lookup can be active at a given time. To provide an online lookup when the user is online and an offline lookup when the user is offline, create both types of lookups and select the Is Active check box in the online lookup.
7. In the Mapped Record ID field, select the record ID field and click Save.

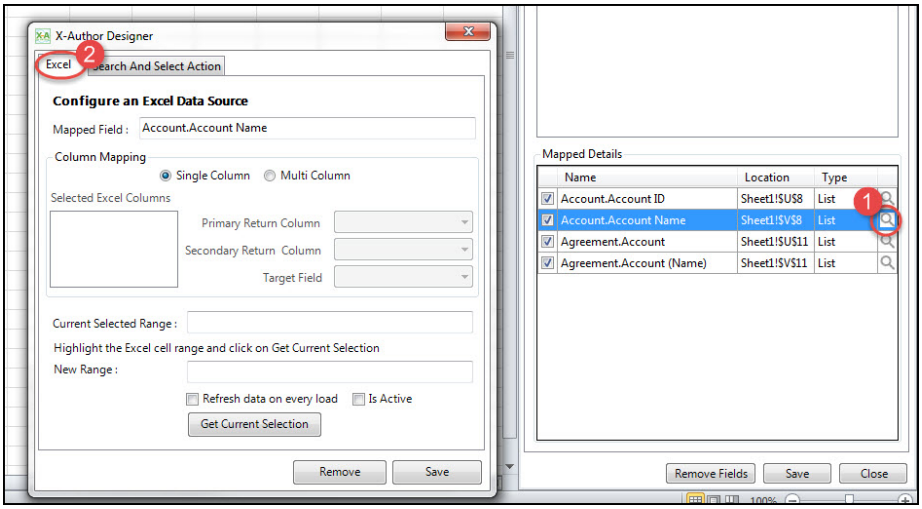
   This ensures that the correct lookup will be returned to the record. To remove an existing Online Lookup, click Remove.
8. Click Save Application.

To create an offline lookup for single column

1. Click **Display Map**.

2. Select a display map and click **Edit**. The Display Map panel opens.

3. In the Mapped Details section of the Display Map pane, click to open lookup properties for that field. The X-Author Designer window opens.



4. Click the **Excel** tab and from the Column Mapping section, select Single Column.

> The single-column option should be used only when the list of values does not contain duplicate lookup values because it returns only the lookup value. If there are duplicates, the first value found in Salesforce is populated in the lookup field, and that may not be the value intended by the user.

5. Enter the range representing the lookup values. This can be done in the following ways.
   - In New Range, type the single column cell range that contains the list of values to be searched and click **Save** to save the data source.
   - Highlight the list values in the worksheet and click **Get Current Selection**.
6. To make offline lookup the default lookup method, select the **Is Active** check box. Either Offline or Online can be active at a given time.
7. To provide an online lookup when the user is online and an offline lookup when the user is offline, create both types of lookups and select the **Is Active** check box of online lookup. When the user is online, the Online Lookup will be active. When the user goes offline, the offline lookup will automatically become active.

> The first time a lookup is initiated by the end user, the lookup function loads data from the worksheet and stores it in the Lookup window for quick results on successive lookups. If you expect changes in lookup data, then refresh the lookup fields every time you load the lookup by selecting the Refresh on Every Data Loadcheck box.

8. Click **Save**.

To create an offline lookup for multiple columns

1. Click **Display Map**.
2. Select a display map and click **Edit**. The Display Map panel opens.
3. In the Mapped Details section of the Display Map pane, click to open lookup properties for that field.
4. Click the **Excel** tab and, under the Column Mapping section, select Multi-Column.

> This is the preferred lookup option for offline lookups because it handles duplicate lookup values correctly.

5. Enter the range representing the lookup values. This can be done in one of the following ways.
   - In New Range, type the multicolumn cell range that has the list of values to be searched and click **Save** to save the data source.
   - Highlight the list values in the worksheet and click Get Current Selection.

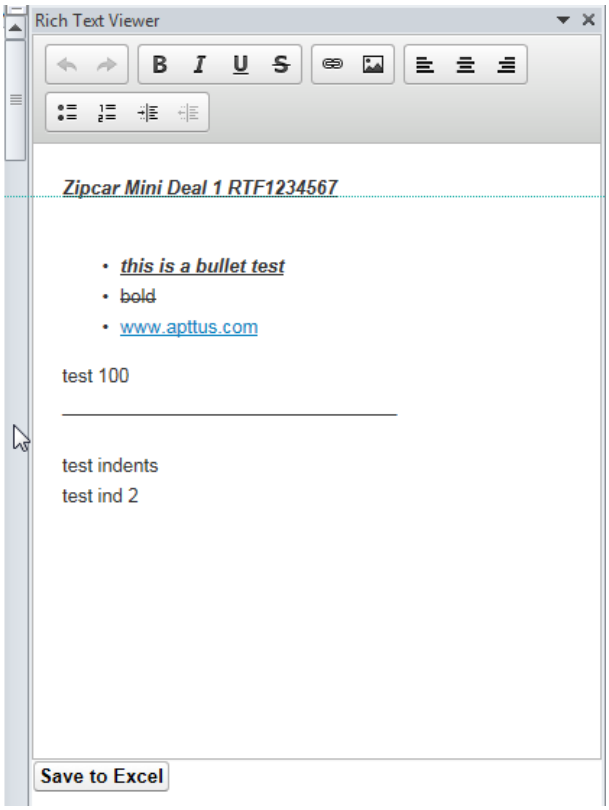
   Your range should always include the record ID of the lookup value (for example, Account ID).
   The column identifiers appear in the display columns.
6. Select the Primary Return Column. This is the column that matches with the lookup value. (For example, Account Name).
7. Select the Secondary Return Column. This should always be the column that has the record ID of the lookup value.
8. Select the Target Field. This should be the record ID field for the lookup name.

9. To make the Offline Lookup the default lookup method, click the **Is Active** check box.
   Note: Either Offline or Online can be active at a given time.
10. To provide an online lookup when the user is online and offline lookup when the user is offline, create both types of lookups and click the **Is Active** check box for online lookup. When the user is online, online lookup will be active. When the user goes offline, the offline lookup will automatically become active.
11. The first time a lookup is initiated by the end user, the Lookup function loads data from the worksheet and stores it in the Lookup window for quick results on successive lookups. If you expect changes in lookup data, then refresh the lookup fields every time you load the lookup by clicking the Refresh on Every Data Load check box.

Rich Text Fields

When a Salesforce field is of type Rich Text, that field can be edited in Excel using a rich-text editor that opens as a new panel.

> **Note**
>
> Rich-text editing is supported only for List objects.



When you are designing an app, you add rich-text fields the same way you would add any field.

When running an app, rich-text fields appear as **Click here to Edit** link.

Rich-Text Editor will not be loaded or is not supported of a rich-text field is added as a save-only field in the repeating group and if a rich-text field is added as an individual retrieved field or as an individually save-only field.

To edit field data when running an app

1. Click **Click here to Edit** to edit a field that already contains data
   Or
   Click **Click here to Add** to edit a field that does not contain data.
   The rich-text editor opens.
2. Edit the field data.
3. Click the **Save to Excel** button to save the changes to your Excel spreadsheet.
4. Save the changes to Salesforce.

Working with Fields and Columns

Moving and Removing Fields

X-Author can easily adjust its map links for when fields need to be relocated, provided these changes are performed correctly. The only way to properly remove a field from a Display or Save Map is through the Remove Fields function in the lower right section of the map. Deleting the cell contents by placing your cursor in the cell and clicking Delete will have no impact and using Excel's native delete row or column functions will corrupt the app.

If you remove a field from a Display Map that is already part of a Save Map, a warning message is displayed that the field will be removed from the Save Map as well. Do not delete a field from the worksheet in any other way, otherwise the name range that governs that field will not get properly removed.

To Remove a Field from a Display Map

1. Open the Display Map.
2. In Mapped Details select the field to be removed.
3. Click Remove Fields.
4. Click Save.
5. Click Close.

To Move a field from one cell to another

1. Close all Display Maps.
2. Place your cursor in the cell to be moved. Right mouse click and select Cut.
3. Place your cursor in the target cell. Make sure there is not already a mapped field in that cell. Right mouse click and select Paste.
4. The label (which may be above or alongside the field) will also need to be moved or just re-entered. Moving the label does not move the data field itself. You can simply drag the label to the new location - it has no impact on the Display Map.
5. Click Save Application. It is this last step that relocates the mapping so **you must save the application after you do any moves**.

Now you can open any Display Map and the adjusted locations will be visible in *Mapped Details*.

Inserting, deleting, and moving columns and rows

To insert or delete Columns or Rows

1. Click Save Application.
2. Close all Display and Save Maps.
3. Insert or delete columns or rows.
4. Click Save Application.

Now you can open any Display Map or Save Map and the adjusted locations will be visible in *Mapped Details*.

To move a column

1. Highlight the column. Right mouse click and select Cut.
2. Highlight the target column. Right mouse click and select Paste.
3. Click Save Application.

Save Maps

- Updates
- Inserts
- Deletes
- Working with Save Maps
- Creating Preloaded Grids

Save Maps are accessed from the Save Map button in the ribbon menu. Save Maps are used to link Salesforce fields to Excel cells for data that is to be saved to Salesforce. X-Author for Excel supports updates, inserts, upserts, and deletes.

Updates

Updates assume that records were retrieved from Salesforce and the data from one or more fields in that record were placed on the worksheet via a Display Map. Using the Display Field function in a Save Map, user modifications to any cell are updated on the record.

Even though a record was retrieved via Salesforce, and the data from one or more fields in that record were placed on the worksheet via a Display Map, you might also want to update a field that was **not** retrieved. In this case, you can use the Other Field function. Since no value is retrieved from Salesforce for this field, it will always be updated to the value in the Excel cell.

Inserts

New records can be created using the Other Field function (see below). To add a record in runtime, a new row must always be added either manually by the user (See Adding Rows/Records to a List) or as a Preloaded Grid (see Preloaded Grids).

Deletes

To delete one or more records from a List Range, the Display Delete Row from Salesforce must be checked in User Menus and there must be a Save Map with at least one field in it.

Working with Save Maps

Unlimited Save Maps	There is no limit to the number of Save Maps.
Unlimited Objects	A Save Map can include fields from any number of objects.
Multiple Worksheets	A Save Map can cross worksheets (but not workbooks).

To create a Save Map

1. Click the Save Map button and click Create. The Save Map pane opens on the right side of Excel.
2. Type a Name for the Save Map. If you are just creating one Save Map, you can just use *Map* or any other name. If you are using multiple Save Maps, then you need a more descriptive name to identify each one.
3. Add fields to the Save Map
    - If you want to save data that has been retrieved via a Display Map and modified by the user, click Add Display Map Fields and select the fields to save. You can use the filter to narrow down the field list, and then click Apply and Save.

- For other cell data (not retrieved) that needs to be saved, you use the Other Field functionality to map the cell to a Salesforce field. Click Add Other Field. Select Individual or List to display the objects you need and then drag the field to the cell. Click Add to Save Map, and then click Save.

4. Click Save Application.

To edit an existing Save Map

1. Click Save Map.
2. From the list of existing Save Maps, select one and click Edit.
3. Make the desired changes and then click Save.

The modified Save Map is saved.

To delete an existing Save Map

> You cannot delete Save Maps that are used in a Save action.

1. Click Save Map.
2. From the list of existing Save Maps, select one and click Delete.

The Save Map is deleted.

Creating Preloaded Grids

There are three ways to add records to Salesforce.

- Click Add Row during runtime.
- Copy and then the X-Author Paste Values or Paste All functions.
- Rows can be added in the Designer that become added rows automatically in runtime without the user having to take any action.

A preloaded grid can only be used to add new records, not for retrieval and update. For example, you might have a six-row-by-three-column grid with three rows of formulas and an empty row between each. The formulas recalculate based on values elsewhere in the workbook. You want each of the three rows to be inserted as Salesforce records each time the “input” values change and the user clicks Save.

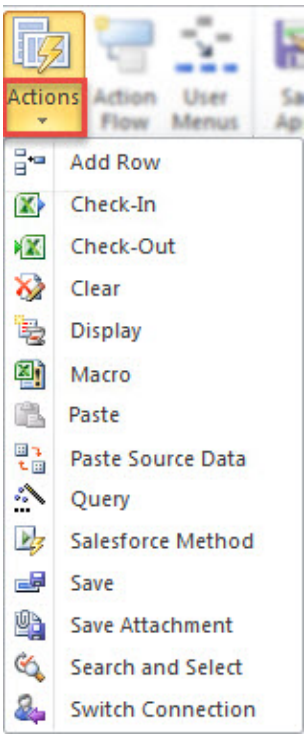
To create a Preloaded Grid

1. In a Save Map, click on Show Preloaded Grid.
2. In the Preloaded Grid window, select the List Object for which you are creating the grid.
3. Enter the number of rows for the Preloaded Grid.
4. Click the check mark when finished. The object will now have an identifier in brackets for the number of rows added. To change the number of rows, enter in a new number in Preloaded Rows.
5. Click Save.

Actions

Actions are used to create a specific user experience. They include being able to query data, display and save it as well being able to manipulate it within the workbook.

The top section of the button provides access to existing Actions. Click the down arrow at the bottom of the button to create new Actions



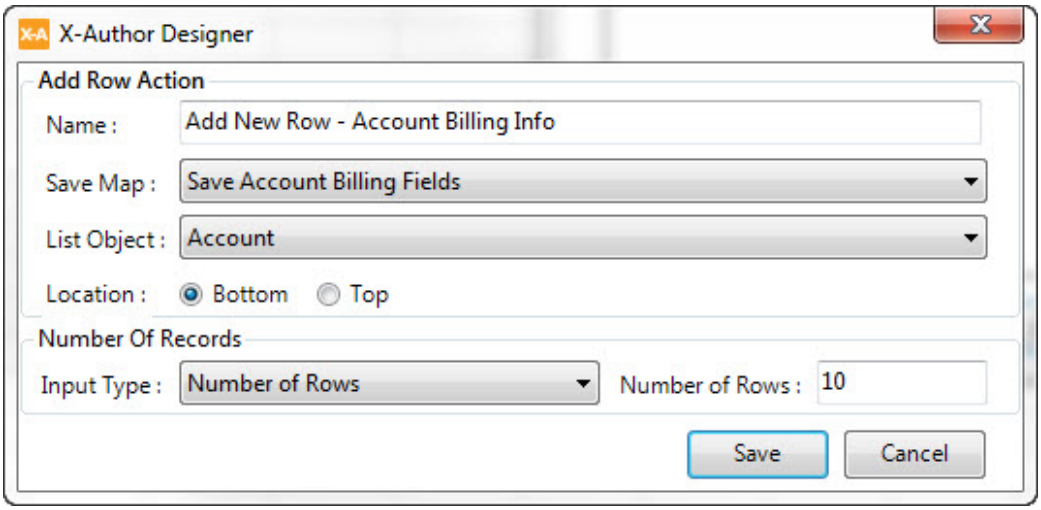
Click on the Action name in the following table for more information on the action.

Action Name	Action Type
Add Row Action	Adds a row to the top or bottom of your worksheet
Check-In	Saves the Excel file to Notes and Attachments in Salesforce and unlocks the file
Check-Out	Opens a previously saved Excel file from Salesforce Notes and Attachments and then locks the file for editing by any other user.
Clear	Clears cell data
Display	Places retrieved data in the worksheet based on a Display Map
Macro	Runs an existing macro

Action Name	Action Type
Paste Action	Pastes data into cells
Query	Queries data from Salesforce
Salesforce Method	Calls a Salesforce Method
Save	Saves worksheet data to Salesforce based on a Save Map
Save Attachment	Saves the entire Excel file or specific worksheets in the Excel file to Notes and Attachments
Search and Select	Allows a user to search for and select data that will then be retrieved into the workbook or that will provide a filter to a subsequent search or query. Retrieves data from Salesforce based on the user record selection.
Switch Connection	Switches between orgs to allow uploading and downloading data across multiple orgs

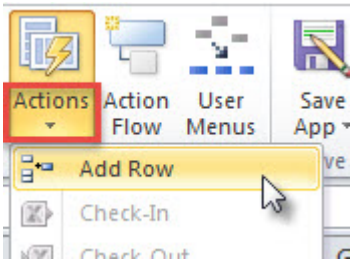
Add Row Action

The Add Row action lets you automatically add rows to the top or bottom of your spreadsheet as part of your flow. You can add a row containing data from a specified set of cells or from another action flow step, or you can add a specified number of empty rows and allow users to insert the data.



To create an Add Row action

1. In the ribbon menu, click the **Actions** drop-down menu and select **Add Row**. The Add Row Action window opens.



2. Complete the fields described in the following table.

Field	Description
Name	Type a name for the new action.
Save Map	Select an existing Save Map to use when adding rows.
List Object	Select the list object that you will be adding rows for.
Location	Specify whether to add the new rows before the existing data or after it.

Field	Description
Input Type	Select the method used to determine how many rows will be added. **Action Flow Step Input** – The number of rows added will be determined by the outcome of a previous step in the action flow. For example, if you have a Search and Select action in the flow, the number of rows added will equal the number of rows selected during that step. **Cell Reference** – Select this option to specify the workbook cell, such as Sheet3!A1, that contains the numerical value of the number of cells you want to add. **Number of Rows** – Select this option to specify a static number of rows to add. **User Input** – Select this option to allow the runtime user to specify the number of rows to add.

3. Click Save.

Check-In Action

This action saves the Excel workbook to the Notes and Attachments of a designated record and then unlocks the workbook and enables it to be checked out by another user.

To create a Check-In action

At least one object identified as *Individual* in Salesforce Objects.

1. Click the lower section of the Actions button and select Check-In.
2. In the Action Name field, type a descriptive name for the action.
3. In the File Name field, type the file name that will be used when the file is checked-in. The user will also be able to provide their own name at runtime.
4. From Object, select an object and click Save. The list of objects will include any Individual Salesforce Objects.

A Check-In action has been created.

Check-Out Action

This action provides the runtime user with a list of all files previously checked into the designated object/record and they can select one of those files to open. This then locks the file for other users until it is checked back in by the current user.

To create a Check-out action

At least one object identified as *Individual* in Salesforce Objects and at least one Check-in action.

1. Click the lower section of the Actions button and select Check-out.
2. In the Action Name field, type a descriptive name for the action.
3. From Object, select an object and click Save. The list of objects will include any object marked as *Individual* in Salesforce Objects.

A Check-out action has been created.

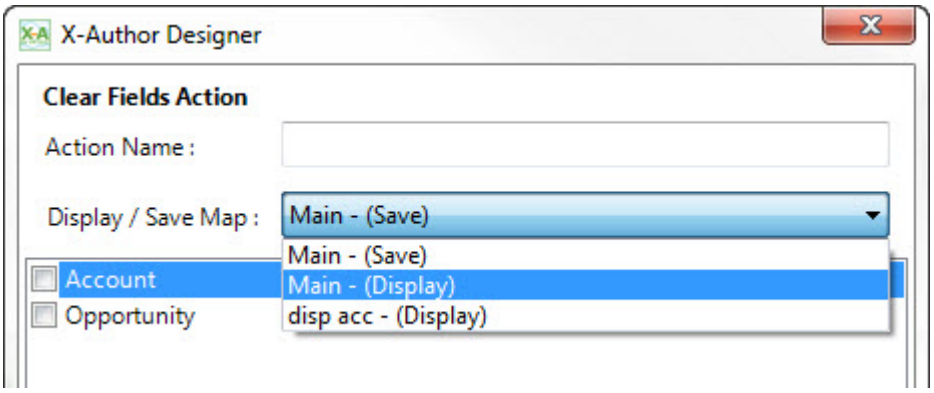
Clear Action

A Clear action clears the cells in the workbook for a specified Save Map. It is used when, for example, you are adding a record to an Individual object using a form and you need to clear the cells after each record has been added so you can add the next one.

To create a Clear action

Clear Actions can only be used to clear *Save Other* fields, meaning those fields that have been added to a Save Map using the Add Other Fields function.

1. Click the down arrow on the Actions button and select Clear from the drop-down menu. The Clear Fields Action window opens.



2. In the Action Name field, type a descriptive name for the action.
3. From the Display/Save Map drop-down menu, select an existing map.
4. Click on the checkbox next to the object to be cleared and it will be displayed in Selected Objects.
5. Click Save. You can select additional Save Maps and objects as necessary.

A Clear action has been created.

Display Action

When data is retrieved from Salesforce, it remains in memory until a Display Action places that data into the workbook based on a Display Map.

To create a Display Action

You must have an existing Display Map.

1. Click the lower section of the Actions button and select Display.
2. In Action Name, type a name. If you are just creating one Display Action, you can use Display. If you have more than one Display action, then you may need a more descriptive name to identify each.
3. From Display Map, select an existing Display Map and click Save.

A Display action is created.

Macro Action

A Macro action executes a macro from the workbook. It can be executed at any time during an Action Flow.

To create a Macro action

You must have an existing Macro in the Excel workbook.

1. Click the lower section of the Actions button and select Macro.
2. In Action Name, type a value that describes which Macro will be executed.
3. From Macro Name, select an existing macro and click Save.

A Macro action has been created.

Paste Action

This action pastes data from other spreadsheets into your workbook.

To create a Paste action

At least one object identified as *Individual* in Salesforce Objects.

1. Click the lower section of the Actions button and select Paste.

A Paste action has been created.

Paste Data Source Action

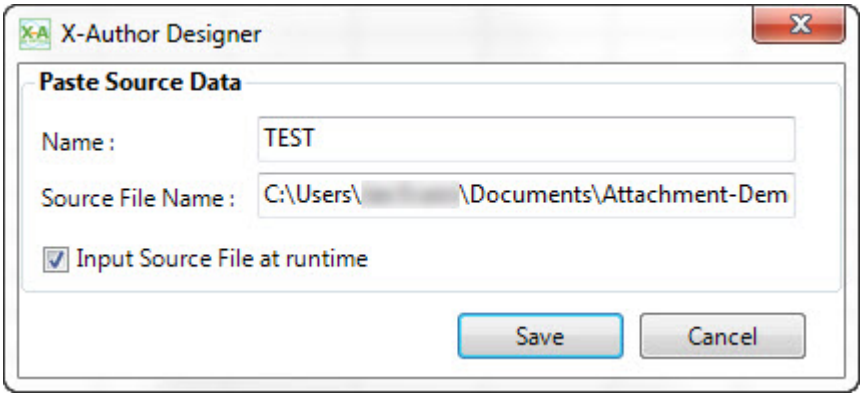
This feature is available only with X-Author editions that include Data Migration. In previous versions of X-Author for Excel, when you used the Source Data function (in the design phase) in combination with Paste with Mapping function at runtime, Paste with Mapping required the runtime user to manually execute the action. In this version, you can create a Paste Source Data action and include that action in the action flow to automatically achieve the same results without user interaction.

Prerequisite

You must be using have licensed an edition that includes data migration.

To create a Paste Data Source action

1. In the ribbon menu, click the **down arrow** on the Actions button and select **Paste Data Source**. The Paste Source Data window opens.



2. In the Name field, type a descriptive name for the action.
3. To automatically paste the data from the source file when the app is started, click the Input Source File at Runtime check box.
4. Click **Save**.

Query Action

The Query action is a point-and-click query builder. This action retrieves data from Salesforce based on predefined parameters and/or user input during App runtime. You can create as many queries as needed and use them more than once across Action Flows.

Within the Query Action, designers can add many types of query filters that are either hard coded or that rely on either user or spreadsheet input at runtime. These filters then can be connected using the Add Filter Logic feature. See the table below for combinations of each of these.

To create a Query action

1. Click the lower section of the Actions button and select Query.
2. Enter an Action Name. Use a descriptive name, e.g. *Query Opportunities*.
3. From Object, select an Object.

4. Type a number in Max Records to specify the maximum number of records to be returned or leaving it blank will return all records unless otherwise restricted by the filters below.

5. Now add query filters as needed by clicking Add Row.

6. In Salesforce Object and Field, click the lookup icon and select a field from the selected object or click on an orange lookup field to navigate to a related object and select a field from there. To select a field from a lookup object, that object and the field must have been selected in Salesforce Objects.

7. Select a Value Type.

   - Input: This is for restricting records from the selected object based on a field from another object. It also implies that an input is required during an Action Flow. For example, retrieve all opportunities for an account that was previously selected by a user in a search and select action. X-Author Designer will automatically suggest matching fields from the related object. Select the correct field from the drop down list.
   - Static: Type a specific value to restrict which records are retrieved. For example, only get records where the geo field equals "West".
   - System Variables: There are three system variables available for use.
     - Current User - Used to restrict records based on the user who is logged in.
     - Current Date - Used to restrict records to the current date.
     - ExportRecordId - This is used for Apps that will be launched from Salesforce. Please refer to that section for usage.
   - User Input: This will generate a user input field at runtime. This option will present the field name to the user at runtime unless you change it as follows:

     Hover over the field name and a pop up will be displayed. Change the field name in the pop up and click the green check mark. To remove a label, click the red cross mark.

   - Cell Reference: This will filter the query results based on the value of a cell.

8. Select an Operator. The following table describes combinations of field types, value types, and operators.

Operator	Equals	Not Equal To	Starts With	Contains	In	Greater Than	Greater or Equal	Less Than	Less or Equal
Datetime									
Value Type									
String									
Static	*	*	*	*	*	*			
Enter Input				*	*				
Cell Reference				*					
Boolean									
Input									
Static	*	*	*	*	*				
Enter Input	*				*				
Cell Reference	*				*				
Lookup									
Input	*	*			*				
Static	*	*			*				
System Variable	*	*			*				
Enter Input	*								
Cell Reference	*								
Date / DateTime									
Static	*	*			*	*	*	*	*
System Variable	*	*			*	*	*	*	*
Enter Input	*	*			*	*	*	*	*
Cell Reference	*	*			*	*	*	*	*
Decimal									
Static	*	*			*	*	*	*	*
System Variable									
Enter Input	*	*			*	*	*	*	*
Cell Reference	*	*			*	*	*	*	*
Double									
Static	*	*			*	*	*	*	*
Enter Input	*	*			*	*	*	*	*
Cell Reference	*	*			*	*	*	*	*
Boolean									
Static	*	*							
Enter Input	*								
Cell Reference	*								

Click to enlarge

9. Click Add Row to create more filter conditions.
10. Click Save.

Filter Logic

This works the same as in Salesforce to implement query filters. As an example, for these filters, find records that match both the first two filters or the third i.e. find records where the amount is greater than “$10 million” AND the close date is earlier than “03/19/2014” OR any opportunity where stage equals “Qualification" Amount > 10000000, Close Data < 03/19/2014, Stage = 'Qualification' You can enter filter line numbers in any order in the Filter Logic text box. For example, 3 OR (1 AND 2) has the same meaning as (1 AND 2) OR 3). See also: https://help.salesforce.com/htviewhelpdoc?id=working_with_advanced_filter_conditions_in_reports_and_list_views.htm&siteLang=en_US
A Query action is created.

Salesforce Method Action

The Salesforce Method action enables you to call Salesforce Apex classes.

To create a Salesforce Method action

1. Click the lower section of the Actions button and select Salesforce Method.
2. Under Action Details, type a Name for the call procedure action.
3. Under Procedure Details, select a Class and type a Method.
4. If the procedure has a return value, select Has return value and select a Return Type object.

5. If the procedure has parameters, select Has params. The Specify Params section is enabled.
6. Type a Param name, and select from the following:
   - Static: Type a value.
   - Object: Enables the Object list. Select an object.
   - Field: Enables the Field list. Select an object field.
   - User Input: This will display an input field to the user at runtime and pass that input to the Salesforce Method.
7. Click Add Param to add more than one parameters.
8. Click Save.

A Salesforce Method action is created.

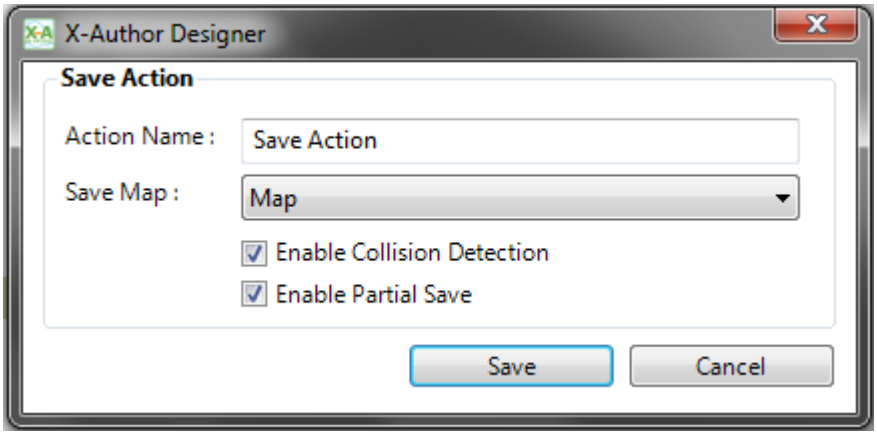
A Save Action pushes Excel workbook data into Salesforce based on a Save Map.

To create a Save Action

You must have an existing Save Map.

1. Click the lower section of the Actions button and select Save.
2. In Action Name, type a name. If you are just creating one Save Action, you can use Save. If you have more than one Save action, then you may need a more descriptive name to identify each.

> Clicking the Enable Collision Detection check box will enforce collision detection during a Save Action. If another user has modified a field in a record subsequent to when that record was retrieved by the current Excel user and the current Excel user is trying to update that same field, then X-Author will prevent that record from being saved.
>
> Clicking the Enable Partial Save check box will allow valid rows to be saved to Salesforce even if other rows have errors or other issues e.g. rows that cannot be saved due to collision detection. Unchecking this box will prevent all rows from being saved even if there is a single error.



3. From Save Map, select an existing save map and click Save.

A Save action is created.

Save Attachment Action

A Save Attachment action saves the entire workbook or specific worksheets to the Notes and Attachments section of an Individual object. A new attachment is inserted each time this action is executed. This action does not provide any file locking. Use a Checkout action if file locking is required.

A Designer can hard code the file name or allow the user to provide the name at runtime. An attachment can be saved either as an Excel or PDF file format.

To create a Save Attachment action

At least one object must be identified as *Individual* in Salesforce objects.

1. Click the lower section of the Actions button and select Save Attachment.
2. In Action Name, type a descriptive name for the action.
3. In **File Name**, type a file name to be used when the file is executed.
   OR
   Click **Use Runtime Filename**, if you want to allow the user to provide the file name at runtime.
4. From Object, select an individual object.
5. In **Format**, select whether to save the attachment as *Excel* or *PDF*.

6. Leave **Custom Sheet Selection** blank to save the entire file as an attachment or select the check box to save just the worksheets into the **Include Sheets** box. If you select the Custom Sheet Selection check box, from the Include Sheets box, you must also select the sheets you want to save.
7. Click Save.

Search and Select Action

This is an interactive way for the end user to search for and select one or more records to retrieve from Salesforce. You can have as many consecutive Search and Select actions as required.

To create a Search and Select action

1. Click the lower section of the Actions button and select Search and Select.
2. On the Options tab, under Search and Select Action, type an Action Name. Use a descriptive name, e.g. *Search Opportunities*.
3. In Object, select an object. In Searchable Fields, select one or more fields that will be displayed to the end user as input fields at runtime
4. In the Result Selection, decide whether the user can select a single or multiple results that are returned to Excel, and select a Result selection. (You can only select Multiple for a List object.)
    - Single
    - Multiple
5. In the Fields to Display in Search Results, select one more fields and use the Sortable field to sort the results. In runtime, the user can also sort by any column when the results are displayed. You can also reorder the result columns using the Arrows to the right of the grid.
6. Add filters as required. Please refer to the Filters section in the Query Action for details on how to create filters.

A Search and Select action is created.

Switch Connection Action

Using the Switch Connection action, you can make your apps automatically disconnect from one org and connect to another while executing an action flow. This is very useful if, for example, you are migrating data between orgs, or need to download and compare data across orgs without

requiring a user to manually switch.

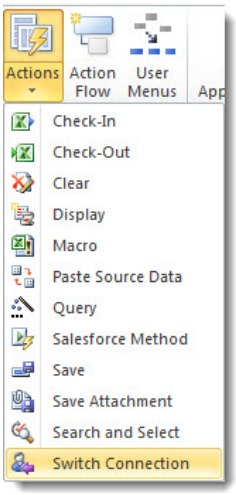
When you design your app and add the Switch Connection action to your action flow, your run-time users are prompted at the beginning of the action flow to add the connection information.

Creating Switch Connection Actions

Switch Connection actions are created through the Actions button on the X-Author Designer ribbon menu.

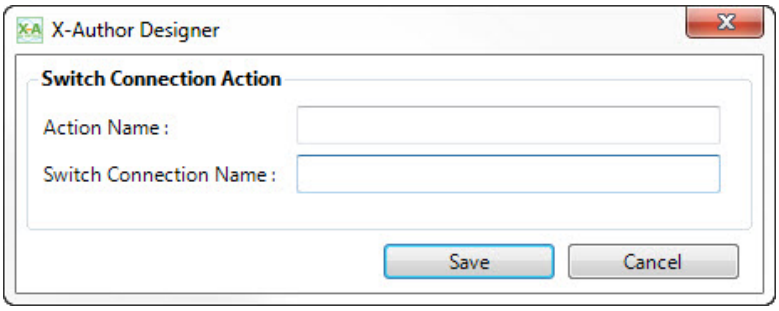
To create a Switch Connection action

1. In the ribbon menu, click **Actions** and select **Switch Connections**.



The Switch Connections Action window opens.

2. Type a name in the Action Name field.



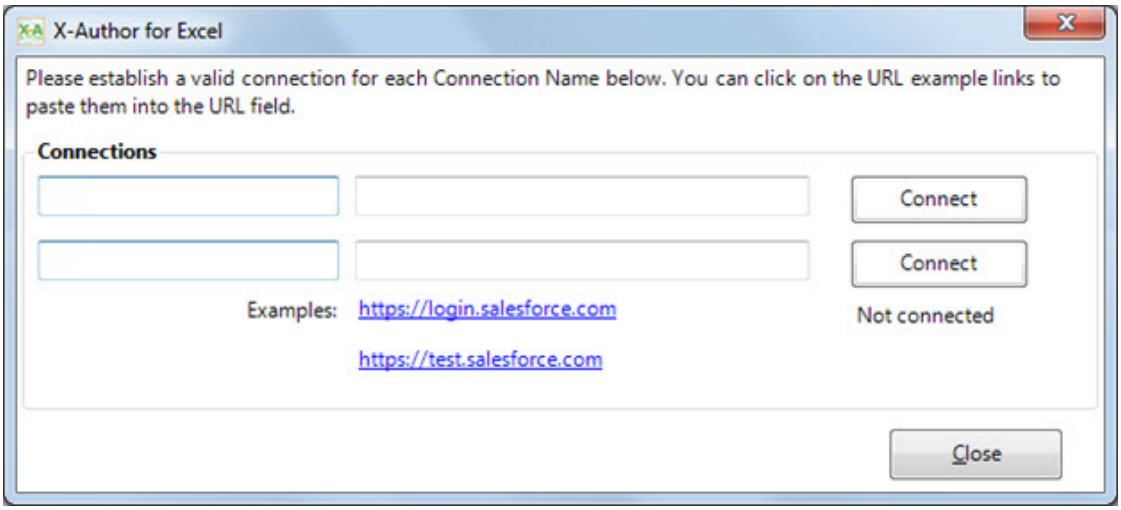
This name displays on the X-Author Designer. For example, this is the name you will see when you design your action flows.

3. Type a name in the Switch Connection name.
   This name displays to the run-time user and should be used to prompt the user to enter the correct connection information.
4. Click Save.

You can now add the Switch Connection action to your action flows.

Using the Switch Connection Action at Runtime

If an app uses the Switch Connection action, the run-time user will be prompted for connection information when the app starts. If multiple connection changes are required, the run-time user will be prompted for all connection information at one time. The following image shows the Connections window in an app that requires two connections.



Enter the connection information in the correct order and clicks **Connect** for each connection. When all connections are completed, click **Close**.

Action Flows

An Action Flow is a sequential set of Actions that is executed when an end user clicks a Designer configured user menu button in runtime. Each Action in an Action Flow can, and in some cases must, take inputs from prior actions and/or provide outputs to subsequent actions.

Examples:

The following table illustrates Action inputs and outputs. Please see the detailed list of examples in the X-Author for Excel: Sample Apps Guide, which covers all Action types.

Action	Input	Output
Check-in	Always because it needs to know on which individual record to store the Excel file. Even though when you create the action itself the individual object is already identified, it's possible you could have used that object in different ways in various action flows so it needs to have the specific output from a prior action as input to this one.	Never
Check-out	Always because it needs to know from which individual record to open the Excel file. Even though when you create the action itself the individual object is already identified, it's possible you could have used that object in different ways in various action flows so it needs to have the specific output from a prior action as input to this one.	Never
Clear Fields	Never. It is tied to a specific save map.	Never
Display	Always. Any object on a Display Map must be represented by an output from a prior Action. Output should be Never.	
Macro	Never	Never
Query	Only if INPUT has been used in the query action filter	Always
Salesforce Method	Optional	Optional
Save	Never	Never
Search and Select	Only if INPUT has been used in the Search and Select action filter	Always

To create an Action Flow

1. Click the Action Flow button and click Create.
2. Enter an Action Flow Name and then click Apply.

> Each Action Flow will ultimately be associated with an end user menu command so use a name that clearly represents this flow. When you click Apply, the Action Flow in the left pane is updated with the name you enter.

3. To add a step in the flow, click Add Step.
4. On the right pane, under Step, type a Step Name (use any name, e.g. *1* or *Search Account*) and then click Apply.
5. Under the Select Action section, select an Action from the list. If this action generates any output, select Persist Data as Output and under Action Output section, type a Name for the output. You can use any name, e.g. *opps*. If this action requires an input, select Provide Input, select one from the Action Input list. Based on the type of Action, X-Author Designer provides input suggestions. See below for more information on when to use inputs and outputs. Repeat steps 3 through 5 to add more than one step. Ensure that you click Apply after each step and notice the hierarchy updated in the left hand pane.
6. To reorder a step in the hierarchy, select the step from the left pane and click Step Up to move the step up in the hierarchy. Click Step Down to move the step down in the hierarchy.
7. After you have created the Action Flow, click Save and don't forget to save your application.

An Action Flow has been created.

To edit or delete an Action Flow

1. Highlight an Action Flow on the Action Flow list.
2. Click Edit or Delete.
3. To delete an Action Flow that has already been associated with a User Menu button, first remove the association in the User Menu.

User Menus

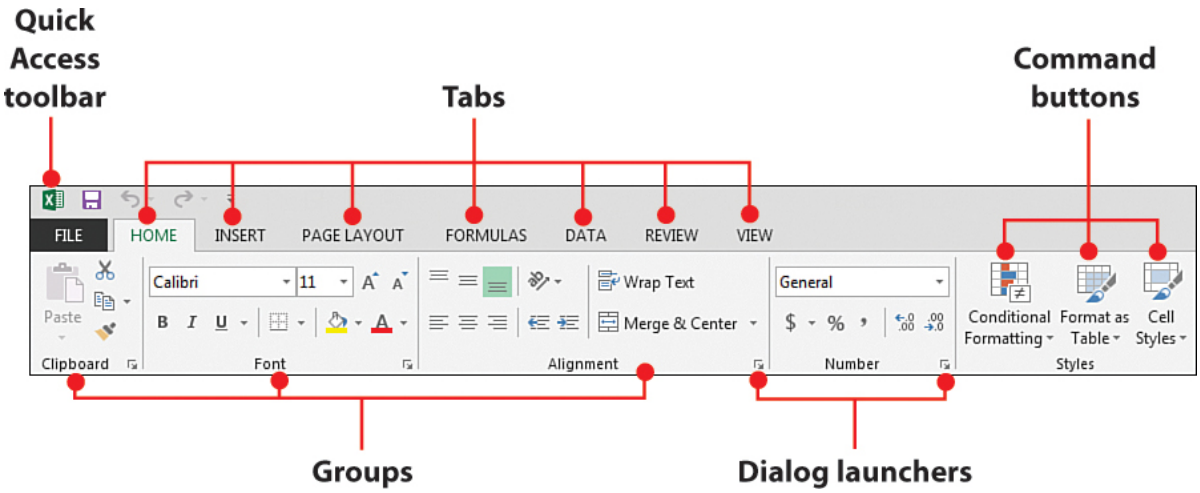
User Menus allow the Designer to create menus that the end user will see in their Excel ribbon when they open an X-Author App in runtime. There are two types of menu items:

- Configurable menu buttons that will execute the Action Flows
- Optional standard buttons that the Designer can use to allow the end user to add rows, delete rows, use special paste, functions, and have access to embedded X-Author Chatter.

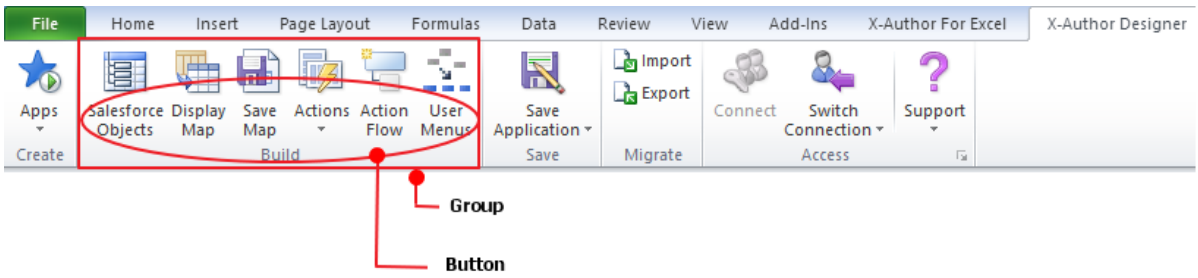
User Menus are divided into two areas:

- **Groups**: These divide the menu buttons into logical areas for the end user. For example, on the X-Author Designer menu, the groups are **Create**, **Build**, **Save**, **Migrate**, and **Access**.
- **Buttons**: These are the menu commands within each menu Group. At least one group and one button within that group are required.

Standard Microsoft Excel Ribbon



X-Author Ribbon



- **Microsoft Icons**
  For icons, X-Author uses the standard Microsoft Office icon list, which can be found here: http://www.microsoft.com/en-us/download/details.aspx?id=21103. Use what is provided below or any other icon from the list. You must enter the icon capitalization exactly as it is described.

To add configurable menu buttons

1. Click the User Menus button.
2. From the Add Groups and Buttons section, click Add Group.
3. Select the *Default_Group* entry in the left pane and from the Properties section, under General, change the Name to the name of the group.
4. To add a button for the group, select the group from the left pane and click Add Button.
5. Under the Properties section, do the following:
   - In Icon, type the name of the Microsoft icon. See the Microsoft Icons below.
   - In **Name**, change the default name to the name of the command.
   - In **Order**, type the order that represents the left to right position where this menu command should appear in the group.
   - In **Tooltip**, type text for a Tooltip.
6. Select an Action Flow to be executed when the end user clicks on this menu button.
7. To activate optional standard menu buttons:
   - To enable the user to add records for any *List Range* in the App, select **Display Add Row**. You will still require a Save Map for this to have any effect.
   - To enable the user to delete List Range records from Salesforce, select Display Delete Row from Salesforce. You will still require a Save Map for this to have any effect.

   Note: Rows are automatically deleted from Salesforce without any further save or other action by the user.

   - To enable the user to use the X-Author Chatter functionality within this App, select Display Chatter.
8. Click Save.

The user menus are created.

Matrix Maps

Matrix maps extend the capabilities of X-Author for Excel well beyond the list format. In lists, one Salesforce record is represented per Excel row or column. Using matrices, a record can be represented by one or more cells by mapping columns and rows (the X and Y axes) of a matrix as well as the data contained in the cell at the intersection of the mapped column and row. Depending on the design, some matrix cells may be updatable by a user. If no value exists in that cell, adding a value can create a new record depending on the matrix design. The following table describes the basic elements of Matrix Maps.

Term	Definition
Matrix	A matrix is any area comprised of one or more "sections."
Section	A section is an area of the matrix that represents a single record. It can include one or many row-column-data combinations. Every matrix has at least one section. Each section can also have its own section filters.
Column mapping	A column mapping is created when you drag a field to the worksheet from a List object. The value in the cell can either be a static field label, a static field value, or a dynamic field value.
Row mapping	A row mapping is created when you drag a field to the worksheet from a List object. The value in the cell can either be a static field label, a static field value, or a dynamic field value.
Data mapping	A data mapping is created when you drag a field to the worksheet from a List object. It must be created **after** the column and row mappings are in place. Each data mapping with its column and row mappings must be its own section or must be added to an existing section.
Matrix filters	Matrix filters are part of the data-getter actions and filter the data as it moves from Salesforce into the matrix. Matrix filters can be displayed above the matrix or elsewhere in the workbook.

- A section is simply any combination of a column and/or a row and the associated data cells, either with or without filters, that represents a single record.

- The simplest section is a combination of a mapped row **or** column and the data cells contained in the row or column. Probably the most common type of section, however, is a combination of a mapped column or columns, a mapped row or rows, and the data cells at the intersection of those column(s) and row(s). A section may also be that same column-and-row-plus-data combination with the addition of filters.
- There can be an unlimited number of sections in a matrix and an unlimited number of matrices in a worksheet.
- The data values can be static or dynamic and can be either field names or field values from one or more objects.
- Filters can be applied to an entire matrix through both the Query action and the Search or Select action, or they can be applied separately for each section.
- Empty rows or columns, with or without formulas, can be added.
- Data is saved via Save Maps which are associated with Sections, not fields.

Examples

The following examples illustrate some of the capabilities of matrices. The first example uses three objects – Budgettrx, GLAccount, and FiscalYear. The objective is to retrieve the Amount field from the Budgettrx object for all Marketing Accounts in the GLAccount object and for each Fiscal Period in Fiscal Year 14.

Departmental Budget

Department	Marketing		
Fiscal Year	2014		
	14-Jan	14-Feb	14-Mar
UK Marketing - Advertising	950.00	1,900.00	800.00
UK Marketing - AR/PR	900.00	1,000.00	1,050.00
UK Marketing - Entertainment	1,000.00	800.00	850.00
UK Marketing - Events		2,100.00	10,000.00
UK Marketing - Field Programs	900.00	950.00	1,150.00
UK Marketing - Mail Campaigns	2,100.00	1,900.00	850.00
UK Marketing - Salaries	1,000.00	1,050.00	2,100.00
UK Marketing - Search	950.00	1,150.00	800.00
UK Marketing - Website	850.00	875.00	950.00

- Each intersecting cell is one record from the BudgetTrx object.
- The data is filtered for the entire matrix to return only Marketing Department data for January 2014 – March 2014.

- Both the X and Y values are dynamically generated based on “Marketing Department” and FY 14 filters.
- There is only one section in the matrix comprised of:
  - Y axis values using the AccountName field values in the GLAccount object.
  - X axis values using the FiscalPeriod field values from the FiscalYear object.
  - The XY intersection, which contains the Amount field from the BudgetTrx object.

The second example is a financial plan from a single object where the columns and rows are static field values and the intersecting cell is the Amount field in that object.

	Forecast Jan	Actual Jan	Variance Jan	Forecast Feb	Actual Feb
Revenue	5,000,000	5,500,000	500,000	6,000,000	6,200,000
Cost of Revenue	3,000,000	3,100,000	100,000	3,600,000	3,655,000
Gross Margin	2,000,000	2,400,000	400,000	2,400,000	2,545,000
% of Revenue	*40.0%*	*43.6%*	*3.6%*	*40.0%*	*41.0%*
			0		
R&D Expense	400,200	485,000	84,800	549,800	539,000
% Revenue	*8.0%*	*8.8%*	0.81%	*9.2%*	*8.7%*

This matrix has multiple Sections because, even though the column, row, and intersecting cells have the same field mappings throughout, the forecast and the actual records differ based on a field that is not displayed on the matrix: the Type field. Therefore there are two sections and they are distinguished by filtering the data for each section using the section filters.

Creating a Matrix

There is no precise order in which to create a matrix. However, following these steps in order will provide better results.

1. Add your Salesforce Objects.
2. Decide how you are going to lay out the matrix, including formulas and empty rows and columns.
3. Create your sections and add your empty rows and columns as you go.
4. If any cells are static, add the cell values at anytime.

5. Extend or copy your sections to fill out the matrix using the Extend function.
6. Add Section Filters.
7. Add Display Filters.
8. Add Save Maps.
9. Add actions, action flows, and ribbon menus.

Creating and Extending Sections

Creating Sections

To create sections

You must have at least one object selected in *Salesforce Objects*.

1. Click Matrix Map and click Create.

   The Matrix Map pane opens on the right side of Excel.

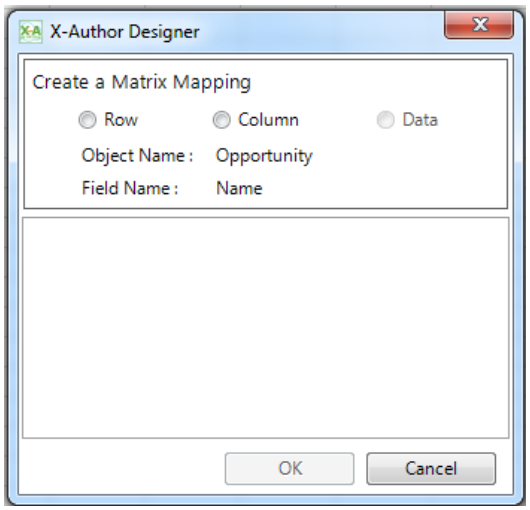
2. Type a name for the Matrix Map.

   If you are creating one Matrix Map, you can use *Map* or any other name. If you are using multiple Matrix Maps, then you need a more descriptive name to identify each one.

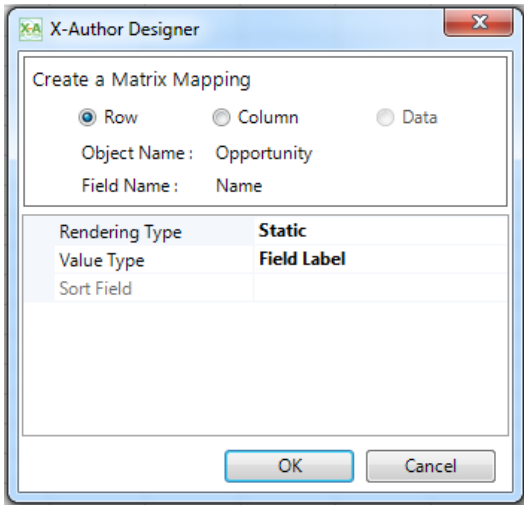
   Each object and its fields selected in Salesforce objects will be available for use in the Matrix Map. Objects are listed based on the object behavior from Salesforce Objects ("Ind" and "List"). List fields are displayed by default. You can switch to Individual fields using the Type list.

3. Create a Row and Column Mapping for each section. *Row and Column Mapping*

   a. Drag a List field onto the worksheet. When you release the mouse, a pop up window appears.



b. Click the Row or Column radio button. The Rendering Type, Value Type, and Sort options are displayed. All of these work together to determine how row or column values are populated.



c. Select a Rendering Type.

- **Static**: This requires you to enter the cell value manually or X-Author uses the Field Label.
- **Dynamic**: Field data is retrieved automatically based on the data getter actions.

d. Select a Value Type.

If you select the dynamic rendering type, this option is unavailable.
If you select Static from Rendering Type, this option becomes available with the following options.

- Field Label: Sometimes, the row or column label and the data value will use the same field mapping. This occurs when you have multiple fields from the same record in the section. This is a placeholder for the Designer to use the Field Name as the row or column label. It is the one instance where the intersecting value is determined from just the column or row mapping plus the data value. For example, if, for a single opportunity, you want to display three fields—amount, volume, and average selling price—then, in the rows, the field names would be described one on top of the other, and, in the data, you would drag the field and the value of that field would be rendered.
- Field Value: Enter the value of the field to be used in determining the data value. If you look at the previous example, the field value would be either Revenue or Cost of Revenue.

e. From Sort Field, select an option from the list of fields.

   This applies only when you select the dynamic rendering type. Usually the field selected from the drop-down is the same field that is being mapped. In some cases, however, you might select a different field. In the example above, there are 12 periods, but if you choose the *period name*, they will be sorted in ascending order, which makes no sense because they should be sorted based on a sequence of 1 to 12. Here you would select *sequence* as the sort field.

f. Click OK to complete the settings for that field.

4. Create Data Mappings for each section. Make sure that the cell you are mapping to is an intersection of a row and column mapping.

   a. Once you have created column and row mappings, you can drag and drop the field that represents the intersection. When you drop the field, the pop up window will appear and, in this case, the radio button will automatically default to Data. It uses the closest row and column combination.

   b. Enter a Section Name for the data or select an existing section name from the drop-down menu.

   c. Click OK to complete the settings for that field. If you added a new section, clicking the Sections tab in the right pane now lists the section you just created.

5. Click Save and Close.

6. Click Save Application.

Extending Sections

Usually, a section will not only be used for a single cell (with its column and row mappings), but will be used across a range of cells. To copy the properties of one section to other cells, designers can use the Extend Section function (similar to the Excel format paintbrush function).

To extend a section

1. Click the Sections tab.
2. Select a section. The cells that comprise that section will be highlighted in the worksheet. Make sure your cursor is in the worksheet data cell you want to copy.
3. Select the remainder of the range to extend to, including the data cell.
4. Click Extend Section in the pane to the right. The highlighted cells will expand and their cells will be mapped on the Fields tab.

The selected section is extended.

Creating Records

Depending on the type of matrix you create, a new record may be created when a user enters a value in a mapped data cell that was previously blank.

Because of all the possible configuration variations, there is no *absolute* rule for record creation, but there are some guidelines, assuming all required fields are ascertainable from the matrix, matrix filters, and section filters.

Records will usually be able to be created when the following criteria are met.

- Both the row and column mappings are lookups to other objects from the mapped data object.
- Column, row, or data mappings are all from the same object, and there are no other fields on the matrix.

X-Author will use the following mappings and filters to create a new record:

- Section Row
- Section Column
- Section Data
- Static values identified as *Equal To* in the Section Filter
- Individual fields used as Matrix Filters

Creating Section Filters

There are two objectives for adding filters to a section.

- To filter the data for that section.
- To serve as the field value when a record is added in that section.

To add a section filter

You must have an exisitng Section.

1. Click Matrix Map.
2. Select the desired Matrix Map and click Edit.
3. Click the Sections tab.
4. Click the icon next to a Section to open the filters window.
5. Add additional rows as needed. Refer to the Query Action section for information on adding rows.
6. Click Save.
7. Click Save and click Close.
8. Click Save Application.

Matrix Filters

Matrix data is filtered based on the Query action and the Search or Select action. To display any of these filters, the fields must be mapped outside the matrix (usually above the matrix).

Displaying individual field filters

If you are using an individual field from another object as a filter, then use the Matrix Map Fields tab to drag that field onto the worksheet. Matrix Map Individual fields have slightly different behavior than Display Map fields because they cannot be updated. They are there for display purposes only.

To display a matrix filter

1. Click the Fields tab.
2. Select Individual as the Type.

3. Select a field and drag it onto the worksheet outside the matrix.
4. Click Save.
5. Click Save Application.

A matrix filter is displayed.

Working with Mapped Fields

Editing Mapped Fields

The Rendering Type, Field Type, and Sort Value can be changed by clicking the button in the Edit column next to the field.

Viewing Sections

To see section layout, highlight a section in the Sections tab and the rows, columns, and data mappings that comprise the section will be outlined.

Extending Sections

Usually, a section will not only be used for a single cell (with its column and row mappings), but will be used across a range of cells. To copy the properties of one section to other cells, designers can use the Extend Section function (similar to the Excel format paintbrush function).

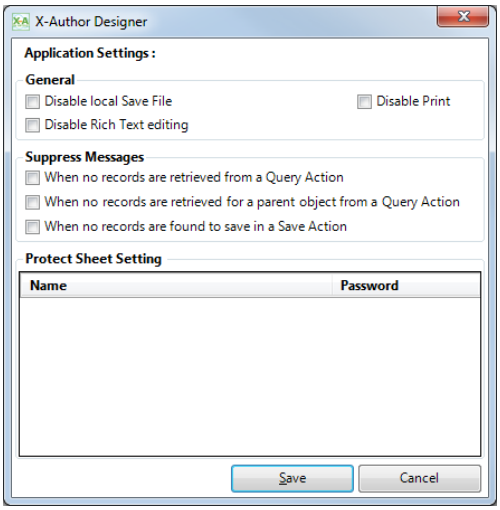
To extend a section

1. Click the Sections tab.
2. Select a section. The cells that comprise that section will be highlighted in the worksheet. Make sure your cursor is in the worksheet data cell you want to copy.
3. Select the remainder of the range to extend to, including the data cell.
4. Click Extend Section in the pane to the right. The highlighted cells will expand and their cells will be mapped on the Fields tab.

The selected section is extended.

Application Settings

The following options are used to affect the runtime user experience, use protected worksheets, and secure Excel file saving and printing.



Disabling the Save and Print Menus

To prevent a runtime user from using the Save, Save As, or Save and Send sub-menus of an Excel File menu, click Disable Local Save File.

To prevent a runtime user from using Microsoft Excel's Print menu, click Disable Print.

Disabling Rich Text editing

To prevent a runtime user from adding and editing rich text data, click Disable Rich Text editing. Use this option when you want the rich-text field HTML to be populated in the cells rather than the editing links that allow the user to add or edit rich text. For more information about rich text editing, refer to Rich Text Fields.

Suppressing Messages

Select these options to suppress messages for the specific conditions described.

Using Protected Worksheets

All Excel sheet protections work with X-Author for Excel provided the password for each protected sheet has been entered. Any worksheets that have been protected will be displayed in the Protect Sheet Setting window. Enter the password for each sheet and click Save.

Only the default Excel options for Select Locked Cells and Select Unlocked Cells are supported. X-Author for Excel will not allow the addition of rows or columns in a protected sheet.

Saving an Application

After you have created or edited an application, you save the Application in Salesforce.com using the Save Application menu function.

When you save an application, a series of validations are executed to ensure that required components have been included in your App. It then saves the changes you have made to the Application, e.g., including an ID field in a repeating Display Map where you also have a Save Map.

The Save Application button is divided into two parts.

Click the top half of the Save Application button to save the Application. This updates Salesforce.com with the updated template and configuration.

Always save your Application after you have completed work in any menu function, e.g., Save Map or Display Map.

Clicking on the bottom half of the Save Application button to *Save As* or *Clone* the Application. X-Author suggests a new name but you can type in a name. This creates a duplicate of the App with the new name.

Cloning an Application

This function copies an Application with an option to save it with a new name.

Clicking on the bottom section of the Save Application button provides an option to Save As or Clone the App. X-Author suggests a new name but you can type in any name. This creates a duplicate of the name with the new name.

Import and Export

X-Author for Excel Apps can be easily migrated from one Salesforce org to another provided that the Salesforce objects being used in the exporting org match those in the importing org.

The Import function imports a previously exported App into the Salesforce org to which you are connected.

To import an app

A previously exported X-Author for Excel .app file.

1. Click Import.
2. To browse for the Application to import, click Browse App.
3. From the X-Author Application Import dialog, select the file with a *.app* extension and click Open.
4. Under App Details, in Import App Name, change the name of the App, if required, and click Import.

The App is imported into your current Salesforce org.

The Export function saves an App as a .app file. This file can then be used by the Import function to add that App to a different Salesforce org.

To export an app

1. Click the Export button.
2. From the X-Author Application Export dialog, choose a location to save the App, type a File Name and click Save.

The App is saved to a file with a .app extension.

Source Data

Data should be rendered from the source org into an App. The App should be saved on the local drive. The App will be later referred to while extracting data for the target org.
To use Source Data to migrate data

1. Open the source App in X-Author Designer and make a note of the starting cell location for each range of data that needs to be copied.
2. Click File > Save As and save the file to your local drive. Use any file name.
3. Open the target App in X-Author Designer and click Source Data on the menu.

	Source Sheet	Source Range	Target Sheet	Target Range
	Opportunities	A3	Sheet1	A4
	Quotes	A3	Sheet2	A4
	Products	A3	Sheet3	A4
*				

4. Select Source Excel Sheet, locate the file above and click Open.
5. Use the drop down arrow under Source Sheet to select a worksheet.
6. In the Source Range enter in the starting range cell location for that worksheet.
7. Use the drop down arrow under Target Sheet to select a worksheet.
8. In the Target Range enter in the starting range cell location for that worksheet.
9. Repeat these steps for each of the other worksheets.

The starting cell in the Target Range must be one row below where you have your fields mapped. So if you have all fields mapped in A3:J3 for example, the Target Range would be A4.

The Paste from mapping function in runtime will copy all data in the list range from the source based on the source range starting location. For example, if you have a data range from A3:J3, it will copy all cells in that range into the target App.

Deleting an Application

You can not delete an App from within X-Author for Excel. You must login to your Salesforce org using a web browser. Navigate to the All Tabs area of your Salesforce org by clicking on the plus sign on the main objects bar. You must have sufficient administrator credentials to perform this task.

To delete an Application

Log in to your Salesforce org using a standard web browser.

1. From the Force.com App Menu, select Apttus X-Author for Excel.
2. Click on the Apps tab in the menu bar.
3. Click on Go next to the X-Author for Excel picklist. This action will display all of the X-Author Designer-created Apps.
4. Click Del next to the App name to delete the App, and then click OK.

You have successfully delete an App from the linked X-Author for Excel Salesforce org.

File Attachments

You can now upload and download file attachments using X-Author for Excel. For example, if you create an app that uses Salesforce objects that contain attachments, such as the Opportunity object, you can now download opportunity-related attachments from Salesforce and upload attachments to Salesforce using X-Author for Excel. All you need to do is add the attachments field to the Salesforce Objects, the Display Map, and–if you are uploading attachments–the Save Map in X-Author Designer. This field can be used in Display Map and Save Map.

Downloaded files are located in **%AppData%\Local\Temp\X-Author Attachments\[Object_ID]**, where %AppData% is the location of the logged-in user's AppData directory in Windows (for example, C:\Users\bdwyer\AppData) and Object_ID is a mashup of the name and the ID of the object the attachment is associated with, such as an Opportunity. You can upload files from anywhere on your workstation, though. To upload a file, you need to provide the full path to the document and the document name as shown in the following example. If you are uploading multiple attachments, separate them with a pipe (|) as shown in the following example.

	A	B	C	D	E	F
1	Opportunity List					
2						
3	Opportunity List					
4	Opportunity ID	Name	Stage	Close Date	Closed	Notes & Attachments
6	006i000000LoHFKAA3	Acme - 1,200 Widgets 11	Prospectin	5/11/2015	FALSE	C:\Users\abaum\Contracts\Acme05112015.docx|C:\Users\abaum\Contracts\Acme05112015.pdf
7	006i000000LoHSaAAN	Acme - 1,200 Widgets 11	Prospectin	4/11/2015	FALSE	C:\Users\abaum\Contracts\Acme04112015.docx|C:\Users\abaum\Contracts\Acme04112015.pdf
8	006i000000LoHLxAAN	Acme - 1,200 Widgets 11	Prospectin	3/11/2015	FALSE	
9	006i000000LoHFLAA3	Acme - 1,200 Widgets 14	Prospectin	2/11/2015	FALSE	
10	006i000000LoHSbAAN	Acme - 1,200 Widgets 14	Prospectin	5/11/2015	FALSE	
11	006i000000LoHLyAAN	Acme - 1,200 Widgets 14	Prospectin	5/11/2015	FALSE	

In this example, the files will be uploaded to the opportunity when you save the data to Salesforce.

Activating and Assigning Apps

Once you have created and app, you need to activate the app in Salesforce and assign it to runtime users.

Activating Apps

After you create an app in X-Author for Excel, the app must be activated in Salesforce.

To activate apps

1. Login to Salesforce.

2. In the Force.com app menu, click **X-Author for Excel.**
3. Click **Apps.**
4. Click **Edit** next to the newly created app.
5. Click the **Activate** check box and click **Save**.

The app has been activated. Now you can assign it to users.

Assigning Apps to Runtime Users

Now that you've activated the app, you need to assign it to runtime users. Assigning the app allows specific users or users with specific profiles to access the app in the X-Author for Excel ribbon menu in Excel.

To assign apps to runtime users

1. Login to Salesforce.
2. In the Force.com app menu, click **X-Author for Excel.**
3. Click **Apps.**
4. Click **Assign** next to the newly created app.
5. Check the boxes for the appropriate users or profiles.
6. Click **Save**.

Launching an X-Author for Excel App from Salesforce

From a button on any page in Salesforce, users can launch an App that opens in Excel and automatically retrieves the data from Salesforce. The flow is:

1. A button is clicked on any Salesforce page (that can accommodate a button).
2. A message box pops up to warn the user that they are leaving Salesforce and to whether they would like to continue.
3. A new instance of Excel is opened and connected to Salesforce without any additional login required by the user.
4. App id and optionally parent id are passed as parameters to the App.
5. The App opens in Excel and executes the Action Flow that has been flagged as auto-execute.

There are three steps required for implementing this feature.

1. Adjust your App slightly to make it launch-able from Salesforce.
2. Add an Edit in Excel setting in Salesforce Setup to identify the App that should be launched from Salesforce.
3. Add a button in Salesforce.

Making an App Launch-Ready from Salesforce

There are only two tiny differences from a regular App: 1, identifying which Action Flow should be executed at launch and 2, in the case of a parent-child App, providing an external id parameter.

Identify the Action Flow to be Executed on Launch

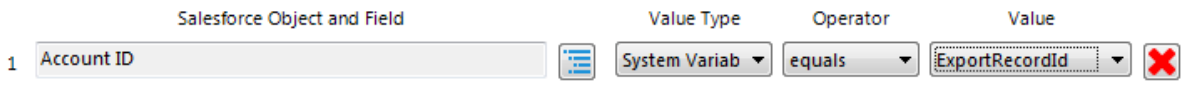
1. Click Action Flow from the menu.
2. Select an Action Flow for auto-execution by checking the box in the Auto-Execute column. Only one Action Flow can be selected.

Adding an External ID to the Query

If this is an App that has no parent, such as a list of opportunities, then make sure the query has the appropriate hard coded filters to retrieve the records you want.

If this is a parent/child App, then the query requires a filter to identify the parent field. For example, if the button is on the Opportunity related list for Accounts, then the Account Id is the parent id that is passed as a parameter.

1. Open the query that is retrieving the data.
2. Add a filter as follows - this example uses Account/Opportunity but you should select the parent id field in the object that looks up to the parent.

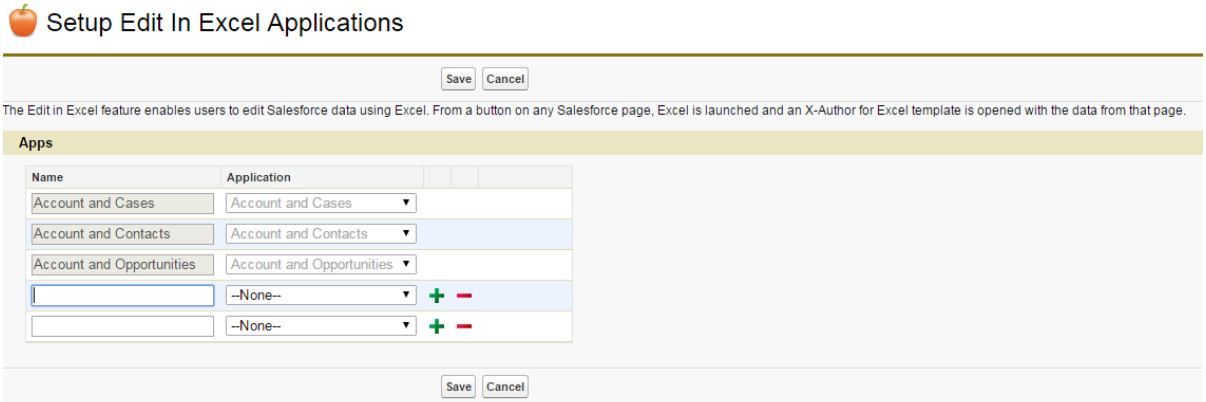


3. Click Save and Save the Application.

Setting Up an Edit in Excel App in Salesforce

1. Switch to the X-Author for Excel application from the Force.com App Menu or locate *Apps* using the + on the menu.

2. Click on Edit in Excel Setup.
3. You will see a list of example Apps that come with the X-Author for Excel managed package.
4. Click + to add a new row.



5. In *Name* enter in a brief description, e.g. if launching Contacts from the Account page, you could use *Account - Contacts*.
6. From the Application dropdown, select the App that will be launched.
7. Click Save.

Creating Buttons in Salesforce

There is no limit to the number of different Apps that can be launched from Salesforce. It could be a list of opportunities, or a list of opportunities or contacts for an account. You can place buttons anywhere Salesforce allows them to be located.

X-Author for Excel provides a few sample buttons. To see an example, search for buttons from Salesforce Setup, then navigate to the Opportunity buttons, and locate the Edit in Excel button.

X-Author for Excel Runtime Functions

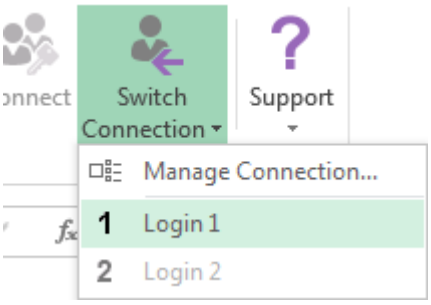
The following runtime fuctions are available for X-Author for Excel apps.

- Switch Connection
- Apps
- Adding Rows or Records to a List
- Adding Rows and Columns to a Matrix
- Paste from Mapping
- Deleting Records from Salesforce
- Chatter
- Rich Text

Switch Connection

To connect to Salesforce from X-Author for Excel at Runtime

1. Click Connect and the Manage Connections window is displayed.
2. Under the Create a new connection section and in the Name field, enter the Salesforce.com user name.



3. In the Login URL, add the URL. In many cases, the URL is login.salesforce.com for production orgs and test.salesforce.com for sandboxes. You can click the shortcut for either of these in Examples. For a custom URL, enter that in.

4. Click Authorize.

The X-Author for Excel ribbon options are enabled.

Open an App.

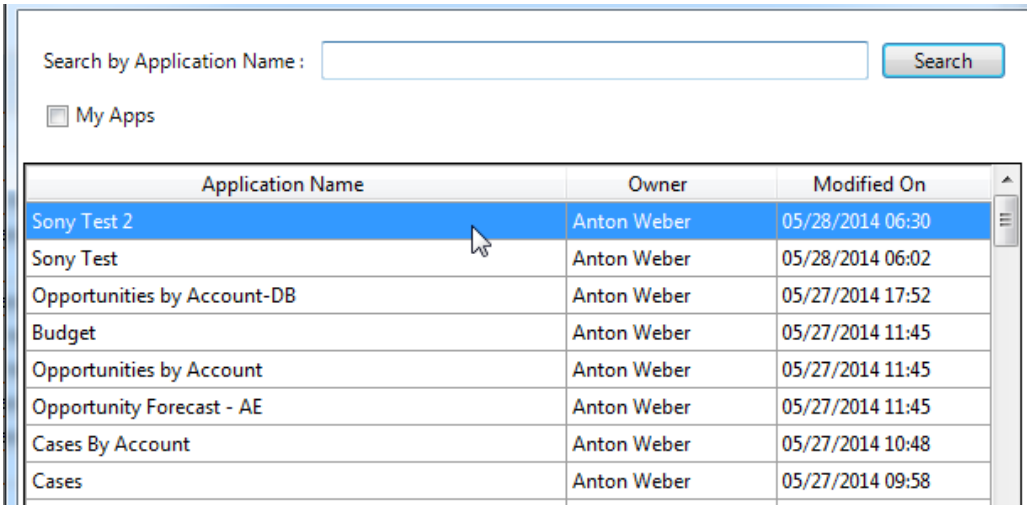
Apps

To be able to open an App, you must be connected to your Salesforce org.

To open an app

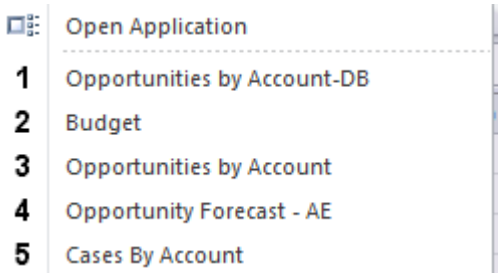
1. Click on Apps and Open Application to view the available Apps to open.

2. Either scroll down to find an App or enter in the App name in Search by Application Name and click Search. Then either double-click on the App or click Select.



> The My Apps check box is an App designer option only for testing their Apps.

3. You can also select an App from the shortcut list.



Once the App loads into Excel, the menu could include buttons specifically designed for that App, e.g. Retrieve Opportunities, as well as one or more of the X-Author for Excel standard buttons: *Add Row*, *Paste*, *Delete Row(s) from Salesforce*, and *Chatter*.

Adding Rows or Records to a List

New rows or records can be added to a list in Excel through one of two ways: Either by using the Add Row button or by using one of the Paste functions from the Paste dropdown menu after you have copied spreadsheet data.

To add a row or record

1. Click on the top part of Add Row to insert a single row or record.
2. Click on the bottom part of Add Row to insert more than one row.

There are three variations of the Paste function: Paste Values Only, Paste All, and Paste from Mapping.

If you are copying data from another worksheet (same or different workbook), Paste eliminates the need to use the Add Rows function. It automatically adds the exact number of rows and pastes the data into those rows simultaneously.

1. Use the standard Excel copy function or Ctrl+C to copy data from the source.
2. Click Paste > Paste Values Only to add rows and paste the data values.
3. Click Paste > Paste All to add rows and paste the data including formulas.

Adding Rows and Columns to a Matrix

Using the Add Matrix Row and Add Matrix Column buttons, you can add more data to your matrix. However, there are specific rules you need to follow. The only time you can add a column or a row is when that row or column is defined with a Rendering Type of Dynamic and a Field Type of Field Value. If either the Rendering Type is set to Static, or the Field Type is set to Field Label, or both, the you **cannot** add the row or column to the matrix at runtime. Click the link after the following image to see all the possible combinations and which combinations allow you to add rows and columns to matrices.

See all possible matrix combinations and whether or not you can add rows and columns

Matrix Configuration		Add Row?	Add Column?
	Column: Static - Field Label	**NO**	**NO**
Row: Static - Field Label	#########Data########		
	Column: Static - Field Value	**NO**	**NO**
Row: Static - Field Label	#########Data########		
	Column: Dynamic - Field Value	**NO**	**YES**
Row: Static - Field Label	#########Data########		
	Column: Static - Field Label	**NO**	**NO**

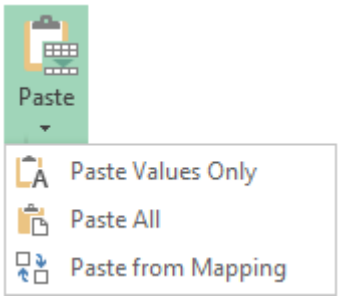
Matrix Configuration	Add Row?	Add Column?
Row: Static - Field Value ########Data########		
Column: Static - Field Value **Row: Static - Field Value** ########Data########	**NO**	**NO**
Column: Dynamic - Field Value **Row: Static - Field Value** ########Data########	**NO**	**YES**
Column: Static - Field Label **Row: Dynamic - Field Value** ########Data########	**YES**	**NO**
Column: Static - Field Value **Row: Dynamic - Field Value** ########Data########	**YES**	**NO**
Column: Dynamic - Field Value **Row: Dynamic - Field Value** ########Data########	**YES**	**YES**

Paste from Mapping

This function is used during data migration between Salesforce orgs and is specifically for copying data between source and target workbooks. See the Data Migration section for more details.

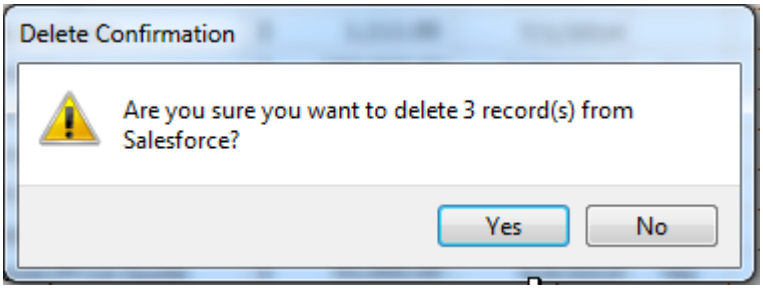
1. Open the *Target App*.
2. Click Paste > Paste from Mapping.
3. Select the source workbook file that contains the data to be copied and click Open.

The data will be transferred from the source file to the Target App.



Deleting Records from Salesforce

To delete a row or multiple rows, select the row or consecutive rows and click Delete Row(s) from Salesforce. X-Author for Excel will then ask you to confirm deletion of these rows. Once you click Yes, the records will be removed from Salesforce without any further save action or other user action. The rows will also be removed from the worksheet.



Chatter

This option opens a Salesforce Chatter feed in a separate window pane to the right of your Excel workbook. It provides both the Chatter feed and various options to share all or parts of your workbook directly into Chatter feeds. The X-Author for Chatter functionality requires both a Salesforce Chatter license that provides API access and an X-Author for Chatter license. For more detailed instructions, please refer to the X-Author for Chatter guide.

Rich Text

If a rich text field has been included in a list, it will display as either *Click here to Add* or *Click here to Edit* depending on whether that field already has data or not. When you click on a rich text field, it will launch a side window that can be used to add or edit rich text. When you have completed the rich text modifications for a record, click Save to Excel. When you are done with all other changes to this or other records, then use the menu button provided in the template to save the data to Salesforce.

FAQs

This section contains a list of frequently asked questions.

How can I access the Product Error Log?

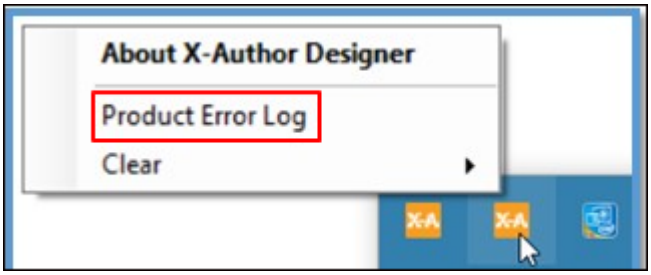
There are two X-Author for Excel icons in the system tray when X-Author for Excel is running. You can access Product Error Logs of X-Author for Excel and X-Author Designer.

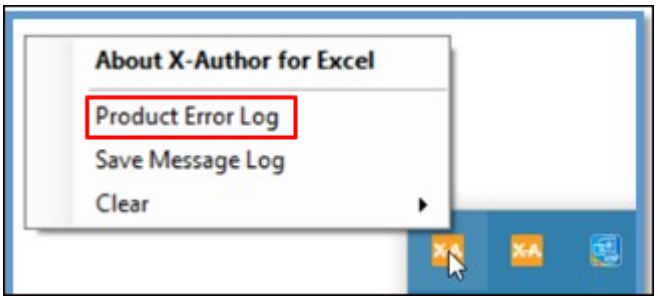
Product Error Log of X-Author for Excel captures error messages while running the app.

Product Error Log of X-Author for Designer captures error messages while designing the app.

To access the Product Error Log

1. Right click on icon and select **Product Error Log**.





How can I access the Save Message Log?

You can access Save Message Logs of X-Author for Excel and X-Author Designer.

Save Message Log of X-Author for Excel captures error messages while running the app.

To access the Save Message Log

1. Right click on  icon and select **Save Message Log**.