

# **APTTUS**

---

## **X-Author for Excel March 2017 Designer and Runtime Guide**

31 March 2017



# Table of Contents

<b>Overview</b>	<b>9</b>
<b>About this Guide</b>	<b>9</b>
<b>Installing X-Author for Excel</b>	<b>11</b>
<b>Minimum Packages Required by Role</b>	<b>12</b>
<b>Installing the X-Author Add-in</b>	<b>13</b>
Installation Requirements	13
Showing Disabled X-Author Add-in Tabs	15
Enabling Proxy Authentication	16
Multi-Language Support	18
<b>Getting Started with X-Author for Excel</b>	<b>20</b>
<b>Required Licenses</b>	<b>20</b>
<b>Setting Salesforce Permissions</b>	<b>20</b>
To create a permission set for Runtime users	21
To create a permission set for Designer users	22
<b>Assigning Field-Level Permissions</b>	<b>25</b>
To set the field-level permissions	25
<b>Completing Designer Setup</b>	<b>25</b>
To add the Assign button	25
To add the Edit in Excel Setup button	26
<b>Connecting to Salesforce from Excel</b>	<b>27</b>
Connecting to Salesforce from Excel	28
Adding More Connections	29

Switching Salesforce Connections	30
Revoking Connections	30
<b>Using the X-Author Designer Ribbon</b>	<b>31</b>
<b>Designing Apps</b>	<b>34</b>
<b>About Apps and Where They Are Stored</b>	<b>34</b>
<b>Choosing the Correct App Type</b>	<b>34</b>
To create an app	35
<b>Creating Apps with Quick Apps</b>	<b>37</b>
To create a Single Object app	38
To create a Parent Child app	46
Customizing the Quick App Color Scheme	56
Example: Quick App for Opportunities by Account	58
<b>Adding Salesforce Objects to your App</b>	<b>78</b>
Object Behavior	78
Object Combinations	78
<b>Working with Display Maps</b>	<b>86</b>
Prerequisites	87
Display Maps and Object Behavior	87
Grouping and Sorting List Ranges	97
Moving and Removing Mapped Fields	103
<b>Working with Save Maps</b>	<b>106</b>
Saving Data to Salesforce	106
Working with Save Maps	107
Creating Preloaded Grids	115
<b>Customizing the App Experience</b>	<b>115</b>
Display Action	117
Query Action	119
Search and Select Action	126
Save Action	129

Clear Action	131
Delete Action	133
Add Row Action	137
Check-In Action	140
Check-Out Action	142
Paste Action	143
Paste Data Source Action	145
Switch Connection Action	147
Macro Action	148
Salesforce Method Action	150
Save Attachment Action	152
Dataset Action	155
<b>Executing Actions</b>	<b>159</b>
To create an Action Flow	159
Action Flow Check boxes	161
Suppress Message Validations	162
Action Flow Required Inputs and Outputs	163
<b>Creating User Menus</b>	<b>165</b>
To add configurable menu buttons	166
<b>Saving or Cloning an Application</b>	<b>167</b>
<b>Deleting an Application</b>	<b>168</b>
To delete an Application	168
<b>Example: Opportunities by Account App</b>	<b>168</b>
App Objectives	168
App Components	169
<b>Example: Opportunity Forecast Application</b>	<b>175</b>
App Objectives	175
App Components	176
Creating the Opportunity Forecast Application	176
<b>Previewing Apps</b>	<b>184</b>

<b>To preview an app</b>	<b>184</b>
<b>Example of previewing an app</b>	<b>185</b>
 <b>Activating and Assigning Apps</b>	 <b>189</b>
<b>About Activating Apps</b>	<b>189</b>
To activate apps	189
<b>Assigning Apps to Runtime Users</b>	<b>189</b>
To assign apps to runtime users	190
 <b>Managing Apps</b>	 <b>191</b>
<b>Application Settings</b>	<b>191</b>
Disabling the Save and Print Menus	192
Disabling Rich Text editing	192
Ignoring Picklist Validation on Save	192
Suppressing Messages	192
Using Protected Worksheets	193
Max Attachment Size Limits	194
Color for Unsaved Records	195
<b>Importing and Exporting an Application</b>	<b>195</b>
To import an app	195
To export an app	196
<b>Migrating Data Between Salesforce Orgs</b>	<b>197</b>
To use Source Data to migrate data	197
Using the Data Migration Wizard	198
 <b>Using Advanced App Design</b>	 <b>224</b>
<b>Creating External Actions</b>	<b>224</b>
Prerequisites for creating Actions	224

<b>Creating Look Aheads</b>	<b>227</b>
To create an online lookup	228
To create an online lookup where new rows are added for each record selected in the lookup	229
To create an offline lookup for single column	231
To create an offline lookup for multiple columns	233
<b>Mapping Records into Columns and Rows</b>	<b>235</b>
Matrix Map Examples	236
Creating a Matrix	238
<b>Uploading or Downloading File Attachments</b>	<b>250</b>
 <b>Launching an X-Author for Excel App from Salesforce</b>	 <b>252</b>
<b>Making an App Launch-Ready from Salesforce</b>	<b>253</b>
Identify the Action Flow to be Executed on Launch	253
<b>Adding an External ID to the Query</b>	<b>253</b>
To add an External ID to a query	253
Setting Up an Edit in Excel App in Salesforce	254
<b>Creating Buttons in Salesforce</b>	<b>255</b>
To create a button on an object	255
To add a button in a Salesforce page	256
 <b>X-Author for Excel Runtime Functions</b>	 <b>258</b>
<b>Switch Connection</b>	<b>258</b>
To connect to Salesforce from X-Author for Excel at Runtime	258
<b>Opening Apps</b>	<b>259</b>
To open an app	259
<b>Adding Rows or Records to a List</b>	<b>260</b>
To add a row or record	260
<b>Adding Rows and Columns to a Matrix</b>	<b>261</b>

<b>Paste from Mapping</b>	<b>263</b>
<b>Deleting Records from Salesforce</b>	<b>264</b>
<b>Adding and Deleting Records</b>	<b>264</b>
Limitations of the Add Rows and Delete Rows functions	265
<b>Chatter</b>	<b>265</b>
<b>Rich Text</b>	<b>265</b>
<b>Configuring Presto Apps</b>	<b>267</b>
<b>Configuring a Presto App</b>	<b>268</b>
<b>Apttus Copyright Disclaimer</b>	<b>271</b>



# Overview

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.** X-Author lets you use Microsoft Excel natively as a user interface (UI) for tasks that 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, and make use of powerful query capabilities.

**For traditional stand-alone spreadsheets.** In many companies, there are hundreds of thousands of spreadsheets. In these companies, people route spreadsheets through email and shared drives. This means you do not always know who has the latest version. If worksheets are cut and pasted 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. Because X-Author centralizes the Microsoft Excel templates in Salesforce, you can create and enforce security and version control over the data even though it is 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. If you still need the actual Microsoft Excel file for audit purposes—or perhaps because it contains more data than you want to store in Salesforce—X-Author offers file check-in and check-out capabilities to provide centralized control.

## About this Guide

This guide is for X-Author for Excel designer users who are responsible for developing apps for runtime users. You can begin at the "Getting Started" section and work your way through the guide to learn how to build a complete, well-designed app or jump to any section in the guide for help on a specific topic.

Included in the X-Author for Excel Designer guide:

- Information for Admins on installation, licensing and permissions for both Runtime and Designer users of X-Author for Excel.
- An overview of Apps and how to get started creating your first app.
- Step-by-step instruction for building Quick Apps, including examples.
- Detailed instructions on how to create apps quickly and easily using X-Author for Excel Designer.
- Advanced features for creating complex apps, like Matrix Maps and External Actions.
- Detailed use cases for each part of the app design process.
- Information on migrating your applications from one Salesforce org to another, including how to use the Data Migration Wizard to create automated migration apps.
- How to launch apps directly from Salesforce.
- How to configure apps for Presto.

Continue to the next section for installation details or skip ahead to [Getting Started with X-Author for Excel](#).

# Installing X-Author for Excel

The following table summarizes the components required to install and begin working with X-Author for Excel. Refer to the most recent *X-Author for Excel Release Notes* for the current versions.

Component	Type	Notes
Apttus X-Author for Excel (Apttus_XApps / Runtime).	Salesforce Managed Package	<b>This package must be installed first.</b>
Apttus X-Author Designer for Excel (Apttus_XAppsDS / Designer).	Salesforce Managed Package	
X-Author for Excel Add-in: Designer + Runtime	Microsoft Excel Add-in	This add-in allows you to work with the X-Author Excel and X-Author Designer tabs in Microsoft Excel.
X-Author for Excel Add-in: Runtime Only	Microsoft Excel Add-in	This add-in allows you to work with the X-Author Excel tab in Microsoft Excel (Runtime users).
<b>The following packages are optional and may be required in addition to the above components based on your license or licenses.</b>		
Apttus X-Author for Excel (Admin)	Salesforce Managed Package	Required for Power Admin licenses or the legacy Data Migration product.

Component	Type	Notes
Apttus Document Generation	Salesforce Managed Package	Required for X-Author Presto licenses. <b>This package must be installed first.</b>
Apttus X-Author for Excel Document Generation	Salesforce Managed Package	Required for X-Author Presto licenses.

## Minimum Packages Required by Role

The following table describes the minimum Salesforce package licenses required for specific X-Author roles. For detailed information on required licenses and permissions, refer to [Getting Started with X-Author for Excel](#).

Role	Salesforce Packages Licensed
Runtime User	X-Author for Excel
Designer	X-Author for Excel, X-Author Designer for Excel
Power Admin	X-Author for Excel, X-Author Designer for Excel, X-Author for Excel (Power Admin)
Presto User	X-Author for Excel, Document Generation, X-Author for Excel Document Generation

## Installing the X-Author Add-in

To design apps, you need to install the *X-Author Designer + Runtime Add-in* for Microsoft Excel. The installation process creates a folder on your computer that contains the files required to use both the X-Author Designer and the X-Author for Excel tabs in Excel. Depending on the version of Windows you are using, the folder is created either in your Program Files or Program Files (x86) directory.



If you are installing the X-Author Add-in for a runtime user, you only need the *X-Author for Excel: Runtime Only* version.

Refer to the most recent *X-Author for Excel Release Notes* for the correct version of the add-in to install.

## Installation Requirements

Your computer must meet the minimum requirements listed below.



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 environment.

Component	Requirement
Microsoft	Microsoft Excel 2010 (32 bit & 64 bit), Microsoft Excel 2013 (32 bit & 64 bit), Microsoft Excel 2016 (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

Component	Requirement
Salesforce	Any Salesforce edition that provides API support for the Salesforce objects that will be accessed in an X-Author for Excel App.

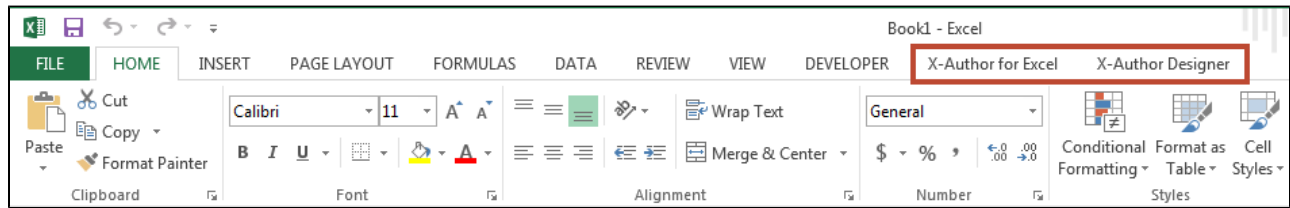
Before you install the X-Author add-in:

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

## To Install the X-Author Add-in

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. After you log in to Windows, the installer automatically resumes.
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 installation 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.

Designers: The **X-Author Designer** and **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.



Runtime Users: The **X-Author for Excel** ribbon tab should be available in Excel.

## Showing Disabled X-Author Add-in Tabs

X-Author for Excel provides the utility *Show X-Author Tabs.exe* to re-enable the X-Author add-in in Microsoft Excel after errors have disabled it. This ensures that the add-in will always be enabled when opening Excel after the utility is run. When you install X-Author for Excel, the diagnostic utility is installed in the same directory. A shortcut for *Show X-Author Tabs.exe* is also created on the desktop.



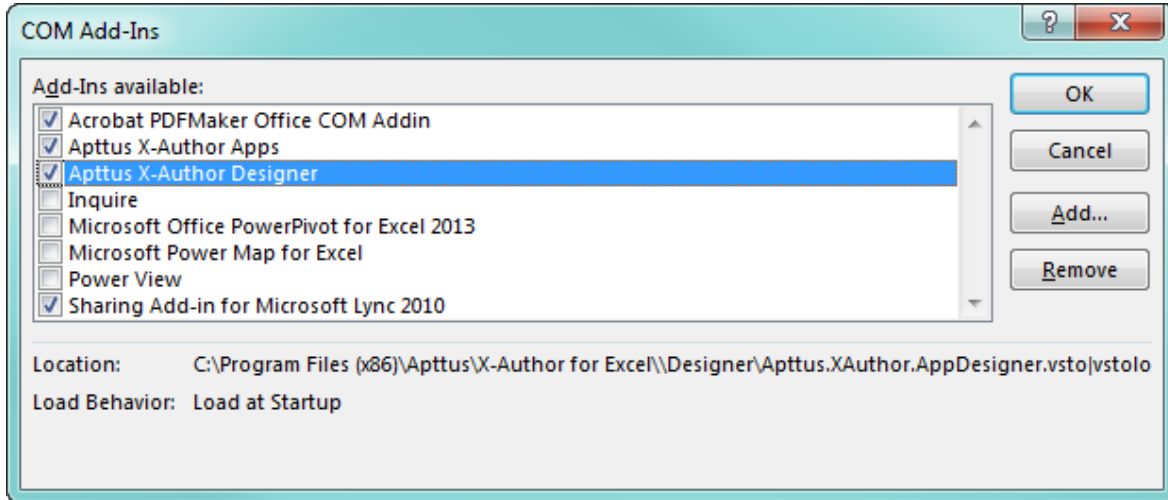
You must to have administrator rights in the system where you want to run the diagnostic tool.

## Registering the X-Author Add-in

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-In

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. Select the **Apttus X-Author Designer** and **X-Author for Excel** check boxes and click **OK**.



The add-ins are now available in the Excel ribbon. You may need to restart Excel for the tabs to display as expected.

## Enabling Proxy Authentication

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

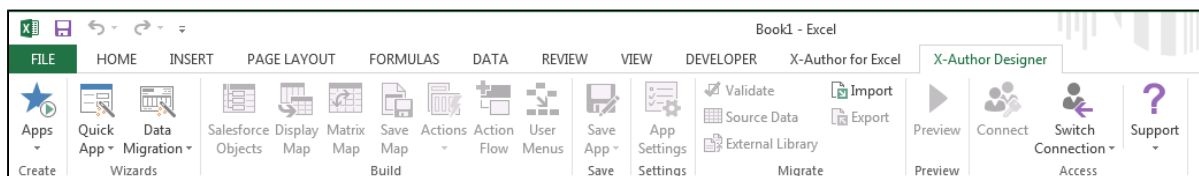
### Microsoft Internet Explorer Proxy Settings

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.

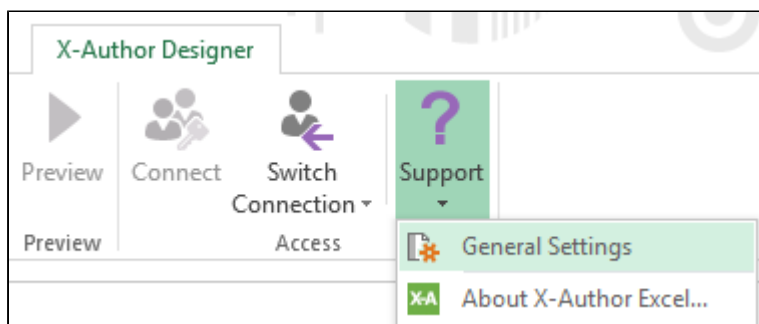
## To enable X-Author for proxy authentication

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





- From the Access section of the ribbon, click on the arrow under the Support icon and choose **General Settings**



- Under the "Proxy Settings" section, choose one of the options described in the following table.

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 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. <i>This is the default setting.</i>
Manual proxy configuration	Click this radio button to enter 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 in the <b>Proxy Address to Use</b> and <b>Port</b> fields, respectively. If you are required to provide authentication information, click the <b>Authenticated</b> check box and enter a user name and password in the provided fields.

4. Click **Apply** to apply settings and **Close** to return to X-Author.

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

## Multi-Language Support

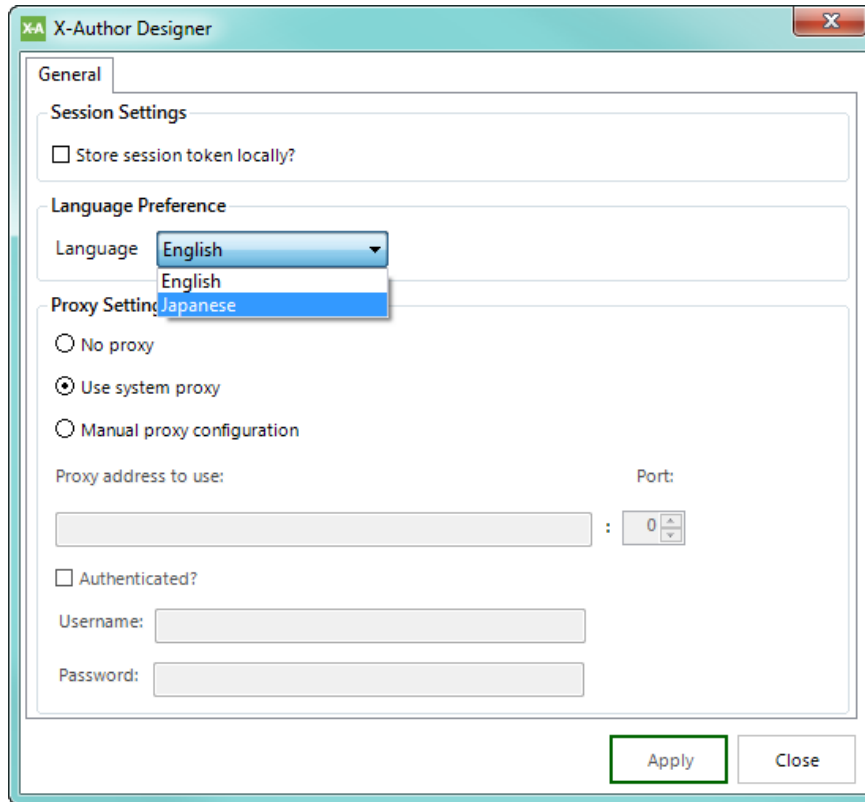
In addition to English, X-Author for Excel is also available in the Japanese (Nihongo) language. X-Author application support is currently provided for:

- Validation Messages
- Captions
- Labels
- Error Messages and warnings

Actual data remains as in your Salesforce org.

You can choose the language to use in X-Author for Excel.

Go to **Support > General Settings** and set the language preference as English or Japanese. After changing this option, restart Microsoft Excel.



# Getting Started with X-Author for Excel

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

- [Required Licenses](#)
- [Setting Salesforce Permissions](#)
- [Assigning Field-Level Permissions](#)

## Required Licenses

App creation requires a combination of licenses and user assignment:

### **Apttus Licenses**

- One or more X-Author for Excel product editions (for example, X-Author Enterprise or X-Author Power Admin).

### **User Assignment:**

- X-Author Designer Package—Assignment to this package is required for app designers.
- X-Author for Excel Package —Assignment to this package is required for designer and runtime users who can run any app that has been both activated and assigned to them by a designer.


### **Salesforce Licenses**

- To access Salesforce.com from Microsoft Excel using any X-Author for Excel edition, a user must have an API-enabled Salesforce.com license to the objects they need to access.

## Setting Salesforce Permissions

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.

To provide access to users, Apttus recommends that administrators create new permission sets for both Runtime and Designer and assign them to the appropriate users.

 As an alternative, you can assign the below permissions to user profiles rather than creating permission sets if it is more convenient.

## To create a permission set for Runtime users

Follow the standard Salesforce steps for creating a permission set and configure the following settings and permissions. After creating the permission set, assign it to one or more **Runtime** users in your org.

Runtime Users																	
Custom App Settings	Apttus X-Author for Excel: Visible																
Tab Settings	X-Author for Excel: Default on																
Custom object permissions	<ul style="list-style-type: none"> <li>Apps: Read</li> </ul> <p><b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; Apps</b></p> <table border="1"> <thead> <tr> <th colspan="2">Object Permissions</th></tr> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input type="checkbox"/></td></tr> <tr> <td>View All</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table>	Object Permissions		Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Delete	<input type="checkbox"/>	View All	<input checked="" type="checkbox"/>	Modify All	<input type="checkbox"/>
Object Permissions																	
Permission Name	Enabled																
Read	<input checked="" type="checkbox"/>																
Create	<input type="checkbox"/>																
Edit	<input type="checkbox"/>																
Delete	<input type="checkbox"/>																
View All	<input checked="" type="checkbox"/>																
Modify All	<input type="checkbox"/>																

Runtime Users																																	
	<ul style="list-style-type: none"> <li>App Assignments: Read  <b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; App Assignments</b> <table border="1"> <thead> <tr> <th colspan="2">Object Permissions</th></tr> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input type="checkbox"/></td></tr> <tr> <td>View All</td><td><input type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table> </li> <li>App Files: Read / Create / Edit / Delete  <b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; App Files</b>  <b>Only required for Check-in/Check-out</b> <table border="1"> <thead> <tr> <th colspan="2">Object Permissions</th></tr> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>View All</td><td><input type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table> </li> </ul>	Object Permissions		Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Delete	<input type="checkbox"/>	View All	<input type="checkbox"/>	Modify All	<input type="checkbox"/>	Object Permissions		Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input checked="" type="checkbox"/>	Edit	<input checked="" type="checkbox"/>	Delete	<input checked="" type="checkbox"/>	View All	<input type="checkbox"/>	Modify All	<input type="checkbox"/>
Object Permissions																																	
Permission Name	Enabled																																
Read	<input checked="" type="checkbox"/>																																
Create	<input type="checkbox"/>																																
Edit	<input type="checkbox"/>																																
Delete	<input type="checkbox"/>																																
View All	<input type="checkbox"/>																																
Modify All	<input type="checkbox"/>																																
Object Permissions																																	
Permission Name	Enabled																																
Read	<input checked="" type="checkbox"/>																																
Create	<input checked="" type="checkbox"/>																																
Edit	<input checked="" type="checkbox"/>																																
Delete	<input checked="" type="checkbox"/>																																
View All	<input type="checkbox"/>																																
Modify All	<input type="checkbox"/>																																
Custom field permissions	<ul style="list-style-type: none"> <li>Apps: Read (All fields)</li> <li>App Assignments: Read (All fields)</li> <li>App Files: Read / Edit (All fields)</li> </ul>																																
Enabled Apex Class Access	Apttus_XApps.AppBuilderWS																																
Enabled Visualforce Page Access	<ul style="list-style-type: none"> <li>Apttus_XApps.SearchSelectAction</li> <li>Apttus_XApps.Landing</li> </ul>																																

## To create a permission set for Designer users

Follow the standard Salesforce steps for creating a permission set and configure the following settings and permissions. After creating the permission set, assign it to one or more **Designer** users in your org.

Designer Users																																	
Custom App Settings	Apttus X-Author for Excel: Visible																																
Tab Settings	<ul style="list-style-type: none"> <li>• X-Author for Excel: Default on</li> <li>• X-Author Designer: Default on</li> <li>• Apps: Default on (Visible &amp; Available)</li> <li>• App Licenses: Default on (Visible &amp; Available)</li> </ul>																																
Custom object permissions	<ul style="list-style-type: none"> <li>• Apps: Read / Create / Edit / Delete  <b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; Apps</b> <table border="1"> <thead> <tr> <th colspan="2">Object Permissions</th></tr> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>View All</td><td><input type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table> </li> <li>• App Assignments: Read / Create / Edit / Delete  <b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; App Assignments</b> <table border="1"> <thead> <tr> <th colspan="2">Object Permissions</th></tr> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>View All</td><td><input type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table> </li> </ul>	Object Permissions		Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input checked="" type="checkbox"/>	Edit	<input checked="" type="checkbox"/>	Delete	<input checked="" type="checkbox"/>	View All	<input type="checkbox"/>	Modify All	<input type="checkbox"/>	Object Permissions		Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input checked="" type="checkbox"/>	Edit	<input checked="" type="checkbox"/>	Delete	<input checked="" type="checkbox"/>	View All	<input type="checkbox"/>	Modify All	<input type="checkbox"/>
Object Permissions																																	
Permission Name	Enabled																																
Read	<input checked="" type="checkbox"/>																																
Create	<input checked="" type="checkbox"/>																																
Edit	<input checked="" type="checkbox"/>																																
Delete	<input checked="" type="checkbox"/>																																
View All	<input type="checkbox"/>																																
Modify All	<input type="checkbox"/>																																
Object Permissions																																	
Permission Name	Enabled																																
Read	<input checked="" type="checkbox"/>																																
Create	<input checked="" type="checkbox"/>																																
Edit	<input checked="" type="checkbox"/>																																
Delete	<input checked="" type="checkbox"/>																																
View All	<input type="checkbox"/>																																
Modify All	<input type="checkbox"/>																																

Designer Users															
	<ul style="list-style-type: none"> <li>App Files: Read / Create / Edit / Delete</li> </ul> <p><b>Permission Sets &gt; Permission Set Overview &gt; Object Settings &gt; App Files</b></p> <table border="1"> <caption>Object Permissions</caption> <thead> <tr> <th>Permission Name</th><th>Enabled</th></tr> </thead> <tbody> <tr> <td>Read</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Create</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Edit</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>Delete</td><td><input checked="" type="checkbox"/></td></tr> <tr> <td>View All</td><td><input type="checkbox"/></td></tr> <tr> <td>Modify All</td><td><input type="checkbox"/></td></tr> </tbody> </table>	Permission Name	Enabled	Read	<input checked="" type="checkbox"/>	Create	<input checked="" type="checkbox"/>	Edit	<input checked="" type="checkbox"/>	Delete	<input checked="" type="checkbox"/>	View All	<input type="checkbox"/>	Modify All	<input type="checkbox"/>
Permission Name	Enabled														
Read	<input checked="" type="checkbox"/>														
Create	<input checked="" type="checkbox"/>														
Edit	<input checked="" type="checkbox"/>														
Delete	<input checked="" type="checkbox"/>														
View All	<input type="checkbox"/>														
Modify All	<input type="checkbox"/>														
Custom field permissions	<ul style="list-style-type: none"> <li>Apps: Read / Edit (All fields)</li> <li>App Assignments: Read / Edit (All fields)</li> <li>App Files: Read / Edit (All fields)</li> </ul>														
Enabled Apex Class Access	Apttus_XApps.AppBuilderWS														
Enabled Visualforce Page Access	<ul style="list-style-type: none"> <li>Apttus_XApps.SearchSelectAction</li> <li>Apttus_XApps.Landing</li> <li>Apttus_XApps.LicenseManager</li> <li>Apttus_XAppsDS.DesignerInfo</li> <li>Apttus_XAppsDS.AppAssignmentOverview</li> <li>Apttus_XAppsDS.AppAssignmentUsers</li> <li>Apttus_XAppsDS.AppAssignmentProfiles</li> </ul>														



## Assigning Field-Level Permissions

The **Activated** and **Uniqueld** 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

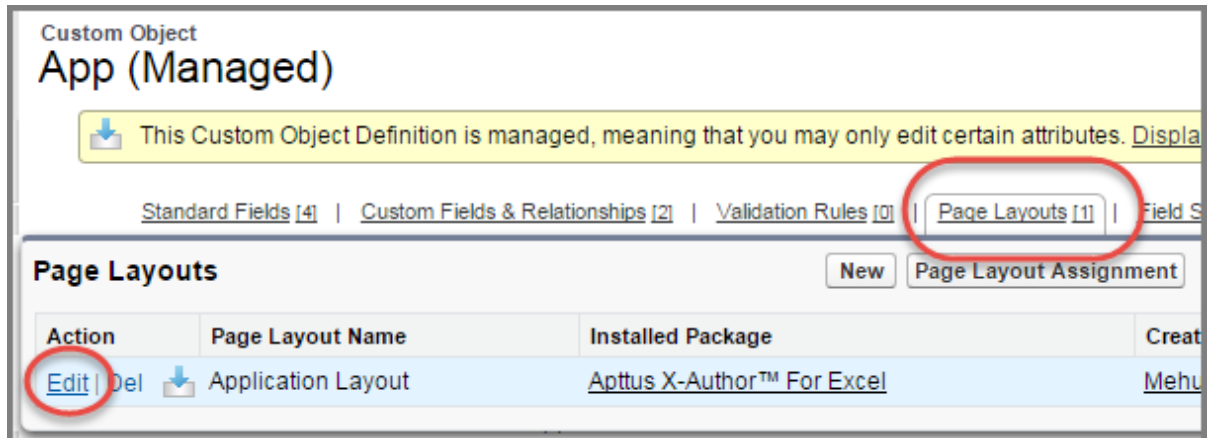
1. In Salesforce App Setup, go to **Create > Objects > App**.
2. Click **Custom Fields and Relationships**.
3. Click **Activated**.
4. Click **Set Field Level Security**.
5. Make sure that the **Visible** check box is selected for every profile that requires app access.
6. Repeat steps 3 – 5 for the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

## Completing Designer Setup

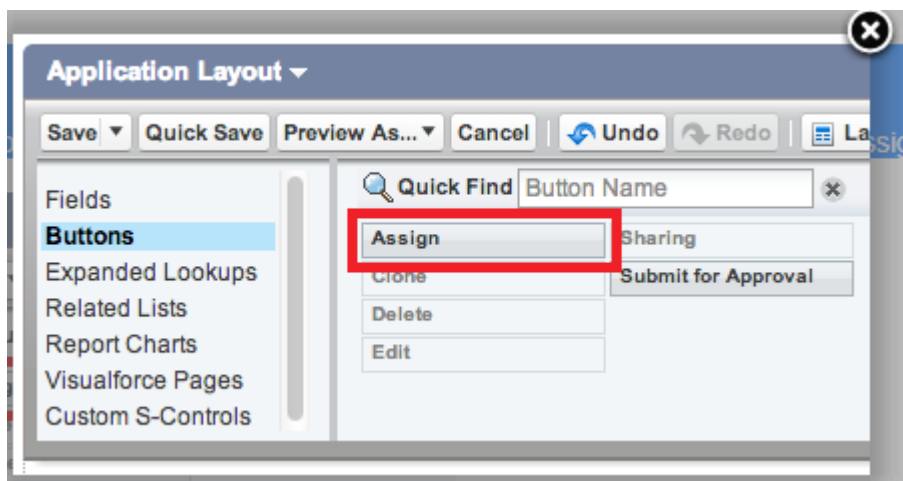
To permit Designers to assign apps they have created to users and use the Launch from Salesforce feature, a Salesforce admin must add two buttons to the App Page Layout: the Assign button and the Excel Setup button.

### 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 is not in the displayed as shown in the following figure, you will need to add it for designers.



**i** Add the assign button to the layout that leads you to the user profile page.

### To add the Edit in Excel Setup button

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.

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
<a href="#">Edit</a>	Search Results	Application Name	
<a href="#">Edit</a>	Lookup Dialogs	Application Name	N/A
<a href="#">Edit</a>	Lookup Phone Dialogs	Application Name	N/A
<a href="#">Edit</a>	Apps Tab	Application Name	N/A
<a href="#">Edit</a>	Apps List View	N/A	New,Accept,Change Owner,Edit In Excel Setup
<a href="#">Edit</a>	Search Filter Fields		N/A

3. Click Save.



- For step-by-step instructions on launching an app from Salesforce, refer to [Launching an X-Author for Excel App from Salesforce](#).
- For step-by-step instructions on activating and assigning apps, refer to [Activating and Assigning Apps](#).

## 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.

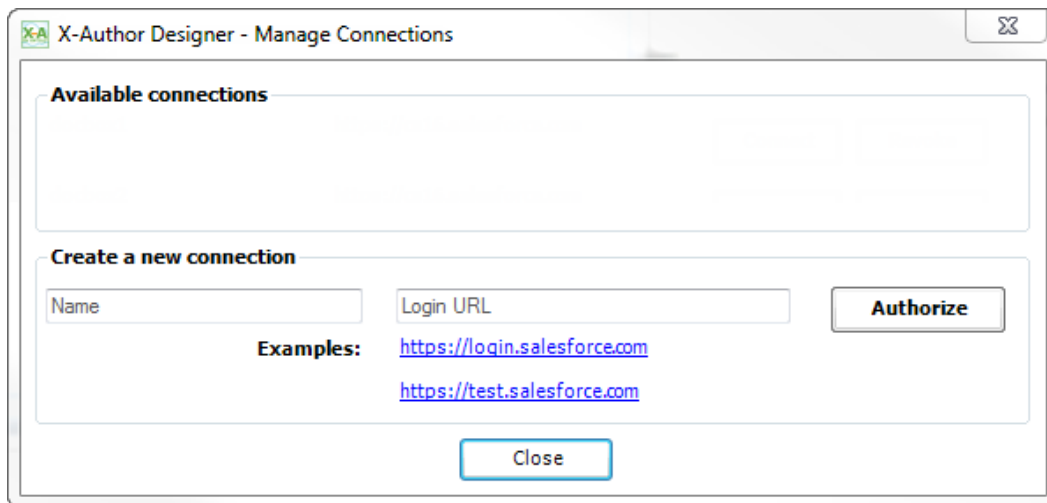
- [Connecting to Salesforce from Excel](#)
- [Adding More Connections](#)
- [Switching Salesforce Connections](#)
- [Revoking Connections](#)

## 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 have 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 *Switching 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](https://login.salesforce.com) for production orgs and [test.salesforce.com](https://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.

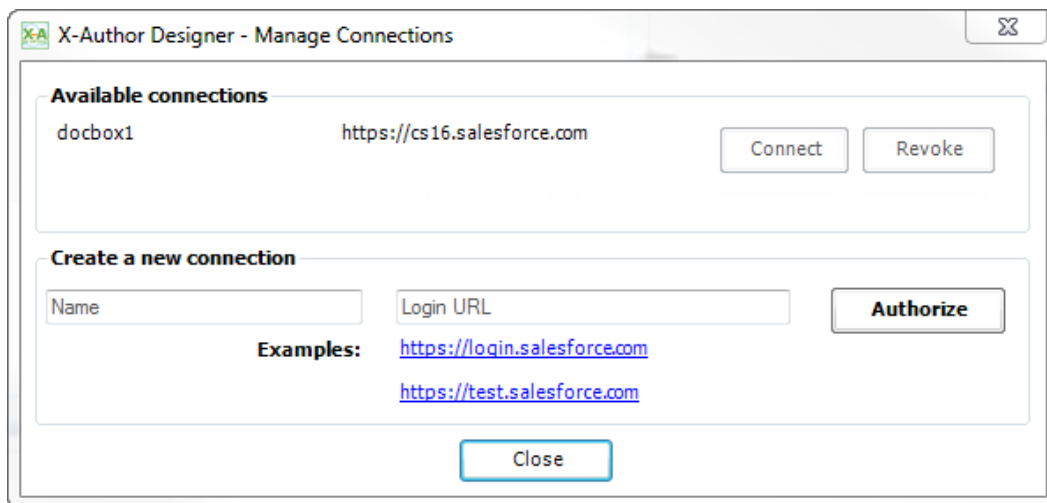
The X-Author Designer ribbon is fully enabled. Now you can start designing an app or using existing apps.

## Adding More Connections

If you use X-Author for Excel with more than one Salesforce org, you can connect to multiple instances of Salesforce.

### 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.

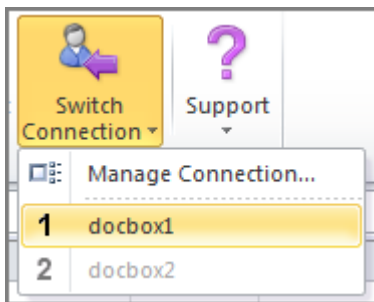
The X-Author Designer ribbon is fully enabled. Now you can start designing an app or using existing apps.

## 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 to a different 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. After a connection is revoked, you will no longer see that connection as an option when switching or managing connections.

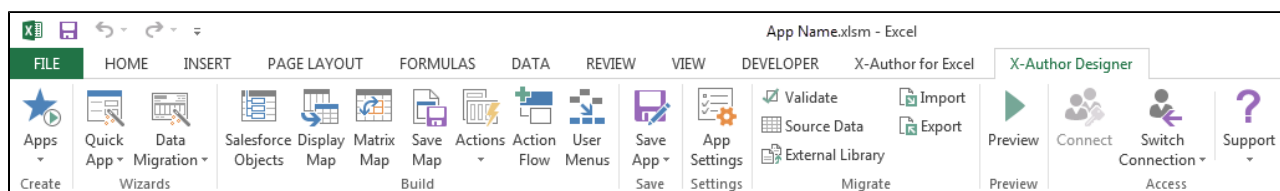
## To revoke a Salesforce connection

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 the connection you want to remove. The Revoke Access confirmation window opens.
4. Click Yes to confirm the action.

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

## Using the X-Author Designer Ribbon

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
<a href="#">Apps</a>	Click this button to create a new app, change the app type, or open an existing app. (Click the Quick App button to instead create a simple single-object app or a parent-child-type app.)
<a href="#">Data Migration</a>	Click this button to create new app using the Data Migration Wizard, or edit an existing Data Migration app.
<a href="#">Salesforce Objects</a>	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.)

Component	Description
Display Map	Click this button to define the associations between Excel spreadsheet cells and Salesforce fields for the Salesforce data to be retrieved. When data is retrieved from Salesforce, using one of the X-Author data invoking 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 map. 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 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.
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.
Save App	Click this button to save or clone your application.
App Settings	Click this button to restrict runtime functions (such as saving, printing, rich-text editing) and password-protect worksheets.



Component	Description
Import	<p>Click this button to import a previously exported app from a different Salesforce org.</p> <p><b>Note:</b> All Salesforce objects used in the source org must also exist in the destination org.</p>
Export	<p>Click this button to export an app from one Salesforce org for the purpose of importing it to another.</p>
Validate	<p>Click this button to validate an app after any import or data migration is complete.</p>
Source Data	<p>Click this button to easily migrate data from one Salesforce org to another using X-Author for Excel.</p>
Preview	<p>Click this button to preview your app in Runtime mode.</p>
Connect	<p>Click this button to connect to your Salesforce org.</p>
Switch Connection	<p>Click this button to to switch your connection to another org, add additional connections, or to revoke a previously configured connection.</p>
External Library	<p>As an application designer, you can choose to create additional custom actions apart from the ones provided in the X-Author for Excel executable packages. For example, you can create a custom action to populate missing External IDs for the Custom Objects in your org.</p>

# Designing Apps

## About Apps and Where They Are Stored

An X-Author for Excel application (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, 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 in which the Microsoft Excel spreadsheet template, an XML file, and other configuration data are stored in the Notes and Attachments. (These files should not be edited without using the X-Author Designer 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.

## Choosing the Correct App Type

Before you begin designing an app, you need to consider what type of app you want to create. This will usually be dictated by the type of license you have.

Apps can be created in two ways:

- Use the fully featured **Designer**. When you use the designer, each of the components must be configured manually (for example, Display Map and Save Map).

- Create a **Quick app**. A Quick app is a 1 – 2 minute wizard experience where a designer can create a simple single object list on its own (for example, Opportunities) or a list filtered by a parent object (for example, Opportunities by Account).

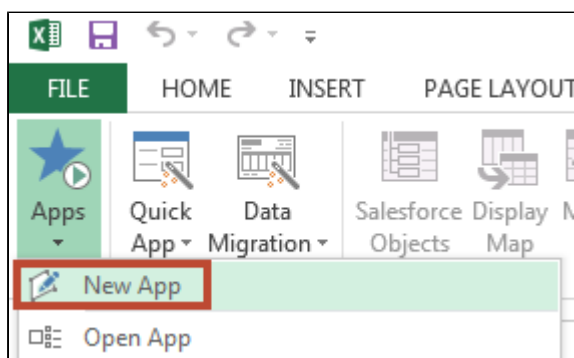
To learn how to create a Quick app, refer to [Creating apps with Quick apps](#).

### **Note**

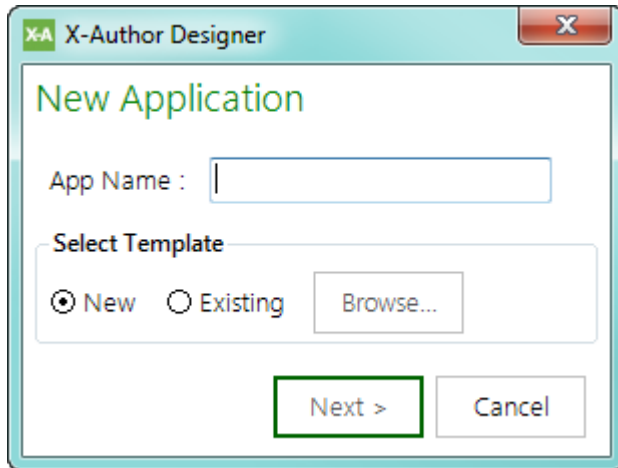
A third type of app can be created using the Data Migration Wizard. This app is specifically designed for migrating data between two organizations. For information on creating a Data Migration app, refer to [Using the Data Migration Wizard](#).

## To create an app

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

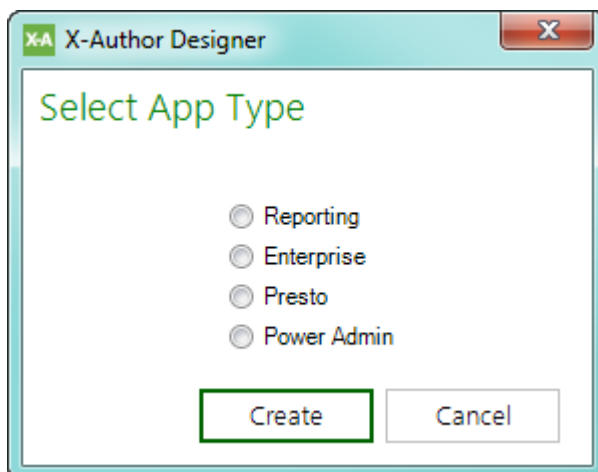


2. Type the app name in the app Name field. This name will display as the window title in the Excel worksheet at runtime.



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

3. To use a new Excel template, click the **New** radio button. Although most existing templates can be used, it is sometimes easier to start with a clean workbook unless users are very familiar with the existing file and change management is a concern.
4. To create an app from an existing Excel file, click the **Existing** radio button and then click Browse to select the template file.
5. Click Next.



6. Select the app edition based on your license and requirements:

- Reporting – choose this app type for *data retrieval only*.
- Enterprise – choose this app type for bi-directional movement of data between Excel and Salesforce. **This is the standard app type.**
- Presto – choose this app type for use cases requiring remote execution of an app. The runtime user does not open Excel to work with this type of app.
- Power Admin – choose this app for use cases requiring data migration between orgs. This app type enables features to aid in the data migration process.

7. Click **Create**.

8. In the ribbon menu, click **Save App**.

You have created an app shell and you are ready to add the various Designer components.

## Creating Apps with 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 [Getting Started with X-Author for Excel](#) and [Connecting to Salesforce from Excel](#).

Use the **Quick App** wizard when you need to create a simple single object list on its own (for example, Opportunities) or a list filtered by a parent object (for example, Opportunities by Account).

The Quick App produces a complete, ready to run app that includes:

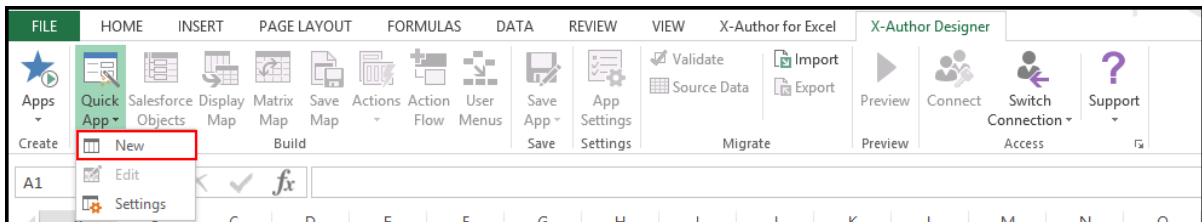
- Two objects (maximum).
- One Display Map, one Save Map, two Action Flows (Data Retrieve and Save), and two User menus (Retrieve and Save). No additional components are available. The Quick App functionality equates to the X-Author Essentials license. A Quick App can be edited using the Quick App editing function but if a Quick App is opened and customized using the full Designer, it can no longer be an X-Author Essentials app.

Follow the step-by-step instructions in the next two sections to create either a single-object or parent-child Quick App.

- [To create a Single Object app](#)
- [To create a Parent Child 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.

**Build a Quick App**  
Provide App name and select App type.

App Name :

Select App Type :

<input checked="" type="radio"/>  <b>Single Object List</b>  Allows for retrieval, editing, and saving of Salesforce data for a list of records for one object based on user selected fields.	<input type="radio"/>  <b>Parent Child</b>  Allows for retrieval, editing, and saving of Salesforce data for both a parent record and a list of child object records based on user selected fields.
--	--

< Back **Next >** Cancel

- Click the Object drop-down menu and select a Salesforce object. The Select Fields section will display the fields associated with the object.
- Click the check box next to the fields you want to include in your app. ( Notice that the two primary fields of the object are selected by default. )



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

- Click **Next**.

**Build a Quick App**  
Select an Object and Fields.

Object : Account

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	Id	Account ID	String
<input checked="" type="checkbox"/>	Name	Account Name	String
<input type="checkbox"/>	Description	Account Description	Textarea
<input type="checkbox"/>	Fax	Account Fax	String
<input checked="" type="checkbox"/>	Phone	Account Phone	String
<input checked="" type="checkbox"/>	AccountSource	Account Source	Picklist
<input checked="" type="checkbox"/>	Type	Account Type	Picklist
<input type="checkbox"/>	Apttus_Config2__Active__c	Active	Picklist
<input type="checkbox"/>	AnnualRevenue	Annual Revenue	Decimal
<input type="checkbox"/>	Attachment	Attachment	Attachment
<input type="checkbox"/>	BillingAddress	Billing Address	NotSupported

< Back Next > Cancel

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

7. Select from the options in the following table.

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



Column	Description
Display Fields	
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.
<b>Grid Columns</b>	
Display	Click this check box to include this field's data in 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. <b>Note:</b> Fields that are not Display fields cannot be Save fields.
Search Fields	Click this check box to add a runtime search filter for this field. These filters will be used to limit the type of records returned when the runtime 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. <b>This field appears only when the Data Criteria is set to User Selection.</b> <b>Note:</b> You must select at least one field, and ID fields are <b>not</b> searchable.
Result Fields	Click this check box to display this field in the search results when retrieving and filtering data from Salesforce. <b>This field appears only when the Data Criteria is set to User Selection.</b> <b>Note:</b> You must select at least one field and ID fields <b>cannot</b> be used.

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



### Selecting Field Options with *Query* Data Criteria

#### Build a Quick App

Select Field Options.

Object :

Data Criteria :

☒ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save
Account ID	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	2	<input type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>
Account Phone	<input checked="" type="checkbox"/>	4	<input type="checkbox"/>
Account Source	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>

### Selecting Field Options with *User Selection* Data Criteria

**Build a Quick App**  
Select Field Options.

Object :


Data Criteria :


Select All  
☒ Display Fields  
☒ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Account Phone	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Account Source	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

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

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

Field	Description
Salesforce Object and Field	Click the lookup icon  to select the object and field that you want the filter to apply to. <b>Note:</b> You cannot type the field name in the text field.
Value Type	Select the value type from the drop-down menu. <b>The value types change based on the Salesforce field or object selected.</b> <ul style="list-style-type: none"> <li><b>Input</b>—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.</li> </ul>

Field	Description
	<ul style="list-style-type: none"> <li>• <b>Static</b>—Select this value to prevent users from changing the Value field at runtime.</li> <li>• <b>System Variable</b>—Select this value to specify a system variable, such as <code>CurrentUser</code> or <code>CurrentDate</code>, as the filter value.</li> <li>• <b>User Input</b>—Select this value to allow users to change the Value field at runtime.</li> </ul>
Operator	<p>Select a relational operator from the drop-down menu.</p> <ul style="list-style-type: none"> <li>• <b>Equals</b>—The field value is equal to the specified value.</li> <li>• <b>Not equal to</b>—The field value is <b>not</b> equal to the specified value.</li> <li>• <b>Starts with</b>—The field value begins with the specified value.</li> <li>• <b>Contains</b>—The field value contains the specified value string.</li> <li>• <b>Not contains</b>—The field does not contain the specified value string.</li> <li>• <b>In</b>—The field value is in the specified list of hard-return-separated values.</li> <li>• <b>Not In</b>—The field value is in the specified list of hard-return-separated values.</li> </ul>
Value	Type the comparison value in this field or select the value from the drop-down menu. <b>Your options will vary depending on the type of field you are filtering on.</b>
	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 <a href="#">remove all filter rows</a> .
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 <b>all</b> of your filter conditions are met or you might want to display results when <b>any</b> 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.

9. Click **Save**.
10. Click **Next**.
11. On Provide App Settings page, provide requisite information as described in the 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).
Enable Save for Google Sheets	Select this check box to allow users to open this X-Author app from within the Google Sheets 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.
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.

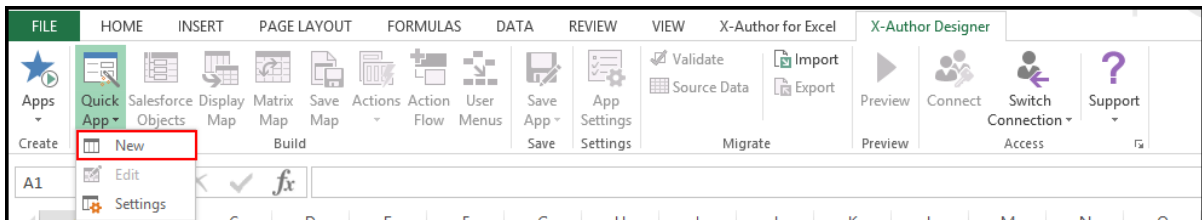
Field	Description
Save Data Button Label	Type the text you want to appear on the ribbon menu button used for saving data to the Salesforce database.
Enable Add Row	Select this check box to allow users to add rows to the Excel worksheet. This option could be used to add records to the workbook so those records can be uploaded to Salesforce.
Enable Delete Row	Select this check box to allow users to delete rows from the Excel worksheet.

- 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 create a Parent Child app

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



The Build a Quick App window opens.

- 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.


- Click the Parent Child radio button and then click **Next**. The Build a Quick App page opens.

**Build a Quick App**  
Provide App name and select App type.

App Name :

Select App Type :

☐  **Single Object List**  
Allows for retrieval, editing, and saving of Salesforce data for a list of records for one object based on user selected fields.

☒  **Parent Child**  
Allows for retrieval, editing, and saving of Salesforce data for both a parent record and a list of child object records based on user selected fields.

< Back   **Next >**   Cancel

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 two primary fields of the object 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.

**Build a Quick App**  
Select Field Options.

Parent Object :

Data Criteria :

☒ Display Fields  
☒ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Phone	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Annual Revenue	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Source	<input checked="" type="checkbox"/>	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Attachment	<input checked="" type="checkbox"/>	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

7. Select the Child Object and the fields that you want to include in your app.



**Build a Quick App**  
Select an Object and Fields.

Child Object: Opportunity (AccountId) ▼

Search Fields:  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	AccountId	Account	Lookup
<input checked="" type="checkbox"/>	Name	Name	String
<input checked="" type="checkbox"/>	Id	Opportunity ID	String
<input checked="" type="checkbox"/>	Amount	Amount	Decimal
<input type="checkbox"/>	Apttus_Approval_Approva...	Approval Status	Picklist
<input checked="" type="checkbox"/>	Attachment	Attachment	Attachment
<input type="checkbox"/>	Automation_Month_c	Automation_Month	Lookup
<input type="checkbox"/>	CampaignId	Campaign	Lookup
<input checked="" type="checkbox"/>	CloseDate	Close Date	Date
<input type="checkbox"/>	IsClosed	Closed	Boolean
<input type="checkbox"/>	CreatedById	Created By	Lookup

< Back Next > Cancel

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

### Selecting Field Options with *Query* Data Criteria

### Build a Quick App

Select Field Options.

Child Object:

Data Criteria :

Filters

Select All

☒ Display Fields

☒ Save Fields

Field Name	Display	Display Order	Save
Opportunity ID	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>
Account	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>
Name	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>
Amount	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>
Close Date	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>
Opportunity Type	<input checked="" type="checkbox"/>	6	<input checked="" type="checkbox"/>
Attachment	<input checked="" type="checkbox"/>	7	<input checked="" type="checkbox"/>
Account (Name)	<input checked="" type="checkbox"/>	8	<input checked="" type="checkbox"/>

< Back

Next >

Cancel

### Selecting Field Options with *User Selection* Data Criteria

## Build a Quick App

Select Field Options.

Parent Object :

Data Criteria :

Select All

☒ Display Fields

☒ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Phone	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Annual Revenue	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Source	<input checked="" type="checkbox"/>	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Attachment	<input checked="" type="checkbox"/>	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

8. Select from the options in the following table.


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


Column	Description
Display Fields	
Save Fields	Click this check box to include all object fields in the Save Map. <b>Note:</b> This check box is inactive unless Display Fields is checked.
<b>Grid Columns</b>	
Display	Click this check box to include this field's data in the data retrieved from Salesforce and displayed in the Excel workbook.
Order	This represents the order, left to right, that fields will be displayed. 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. <b>Note:</b> Fields that are not Display fields cannot be Save fields.
Search Fields	Click this check box to add a runtime search filter for this field. These filters will be used to limit the type of records returned when the runtime 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. <b>This field appears only when the Data Criteria is set to User Selection.</b> <b>Note:</b> You must select at least one field, and ID fields are <b>not</b> searchable.
Result Fields	Click this check box to display this field in the search results when retrieving and filtering data from Salesforce. <b>This field appears only when the Data Criteria is set to User Selection.</b> <b>Note:</b> You must select at least one field and ID fields <b>cannot</b> be used.

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

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

--

Field	Description
Salesforce Object and Field	<p>Click the lookup icon  to select the object and field that you want the filter to apply to.</p> <p><b>Note:</b> You cannot type the field name in the text field.</p>
Value Type	<p>Select the value type from the drop-down menu. <b>The value types change based on the Salesforce field or object selected.</b></p> <ul style="list-style-type: none"> <li>• <b>Input</b>—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.</li> <li>• <b>Static</b>—Select this value to prevent users from changing the Value field at runtime.</li> <li>• <b>System Variable</b>—Select this value to specify a system variable, such as <code>CurrentUser</code> or <code>CurrentDate</code>, as the filter value.</li> <li>• <b>User Input</b>—Select this value to allow users to change the Value field at runtime.</li> </ul>
Operator	<p>Select a relational operator from the drop-down menu.</p> <ul style="list-style-type: none"> <li>• <b>Equals</b>—The field value is equal to the specified value.</li> <li>• <b>Not equal to</b>—The field value is <b>not</b> equal to the specified value.</li> <li>• <b>Starts with</b>—The field value begins with the specified value.</li> <li>• <b>Contains</b>—The field value contains the specified value string.</li> <li>• <b>Not contains</b>—The field does not contain the specified value string.</li> <li>• <b>In</b>—The field value is in the specified list of hard-return-separated values.</li> <li>• <b>Not In</b>—The field value is in the specified list of hard-return-separated values.</li> </ul>
Value	<p>Type the comparison value in this field or select the value from the drop-down menu. <b>Your options will vary depending on the type of field you are filtering on.</b></p>

Field	Description
	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 <a href="#">remove all filter rows</a> .
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 <b>all</b> of your filter conditions are met or you might want to display results when <b>any</b> 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.

10. Click **Save**.
11. Click **Next**.
12. On Provide App Settings page, provide requisite information as described in the table.

Field	Description
Enable App Launch from Salesforce	Click this check box to allow users to open this X-Author app from within the Salesforce application (as well as from within Excel).
Enable Save for Google Sheets	Select this check box to allow users to open this X-Author app from within the Google Sheets 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.
	Type the maximum width of an Excel column (in characters).

Field	Description
Max Column Width	
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.
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.

13. Click **Finish**.

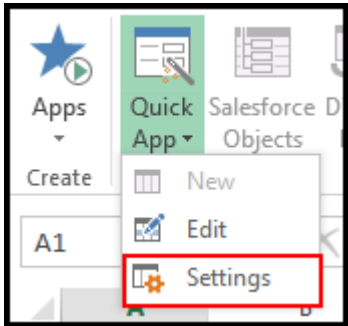
An app has been created with all design functions complete. You can now [Preview the app](#), run the app, or use X-Author Designer to make additional modifications.

## Customizing the Quick App Color Scheme

Using Quick App settings, you can customize the color scheme of any Quick App. This helps the runtime user to distinguish display fields from save fields when working with the app.

### To customize the color scheme

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



The Format Options window opens.

2. Provide requisite information in the Format Options window as described in the table.

Field	Description
<b>Worksheet Title</b>	
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.
<b>Header Display Fields</b>	



Field	Description
Fill Color	Click this field to open the color picker and specify the background color for the Display fields.
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.
<b>Header Save Fields</b>	
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.

The image shows a 'Quick App' dialog box with a 'Format Options' section. This section contains three columns: 'Fill Color', 'Text Color', and 'Font'. Under 'Fill Color', 'Worksheet Title' is set to yellow, 'Display Fields' to blue, and 'Save Fields' to green. Under 'Text Color', 'Worksheet Title' is set to black, while 'Display Fields' and 'Save Fields' are set to white. Under 'Font', 'Worksheet Title' is 'Calibri, 15.75pt, sty', 'Display Fields' is 'Calibri, 11.25pt', and 'Save Fields' is 'Calibri, 11.25pt'. At the bottom right, there are 'Apply' and 'Close' buttons, with the 'Apply' button highlighted by a red rectangle.

	Fill Color	Text Color	Font
Worksheet Title	Yellow	Black	Calibri, 15.75pt, sty
Header			
Display Fields	Blue	White	Calibri, 11.25pt
Save Fields	Green	White	Calibri, 11.25pt

### Example: 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 will 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.

Using these four steps, you are going to create an app that will:

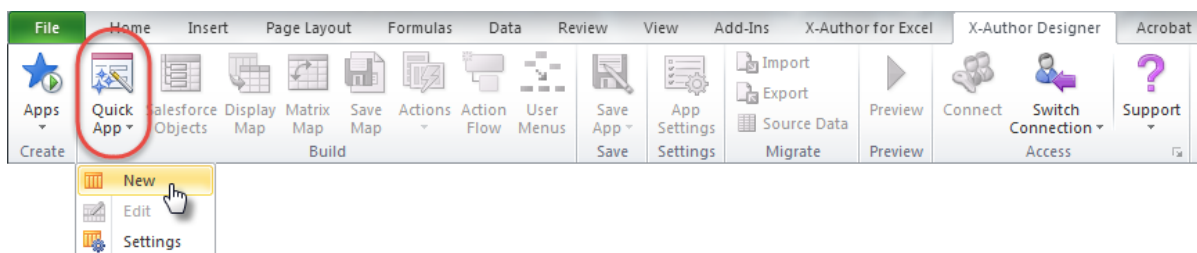
- Retrieve data from the Account and Opportunity objects in Salesforce.
- Place that data in a Microsoft Excel worksheet.

- 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 [Getting Started with X-Author for Excel](#) and [Connecting to Salesforce from Excel](#).

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



The Build a Quick App window opens.


2. In the App Name field, type the application name. The name appears as the Excel window title (located at the top of the Excel window) in the completed app.

**X-Author Designer**


**Build a Quick App**  
Provide App name and select App type.

App Name : Opportunities by Account

Select App Type :

☐  **Single Object List**

Allows for retrieval, editing, and saving of Salesforce data for a list of records for one object based on user selected fields.

☒  **Parent Child**

Allows for retrieval, editing, and saving of Salesforce data for both a parent record and a list of child object records based on user selected fields.

< Back   Next >   Cancel



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. Click the appropriate radio button and select an app type.

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. Select the parent object and fields to use in the app.

**X-Author Designer**

**Build a Quick App**  
Select an Object and Fields.

Parent Object :

Search Fields :  [Clear](#)

Field Id	Field Name	Data Type
----------	------------	-----------

< Back   Next >   Cancel

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.

**Build a Quick App**  
Select an Object and Fields.

Parent Object : Account

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	Id	Account ID	String
<input checked="" type="checkbox"/>	Name	Account Name	String
<input type="checkbox"/>	Description	Account Description	Textarea
<input type="checkbox"/>	Fax	Account Fax	String
<input type="checkbox"/>	Phone	Account Phone	String
<input type="checkbox"/>	AccountSource	Account Source	Picklist
<input type="checkbox"/>	Type	Account Type	Picklist
<input type="checkbox"/>	AnnualRevenue	Annual Revenue	Decimal
<input type="checkbox"/>	Approval_Preview_Status__c	Approval Preview Status	Picklist
<input type="checkbox"/>	Approval_Status__c	Approval Status	Picklist
<input type="checkbox"/>	Attachment	Attachment	Attachment

< Back   Next >   Cancel

Notice that the two primary fields are selected by default. Select the Account fields that you will use 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.

**Build a Quick App**  
Select an Object and Fields.

Child Object: Opportunity (AccountId)

Search Fields:  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	AccountId	Account	Lookup
<input checked="" type="checkbox"/>	Name	Name	String
<input checked="" type="checkbox"/>	Id	Opportunity ID	String
<input type="checkbox"/>	Amount	Amount	Decimal
<input type="checkbox"/>	Approval_Preview_Status__c	Approval Preview Status	Picklist
<input type="checkbox"/>	Apttus_Approval_Approva...	Approval Status	Picklist
<input type="checkbox"/>	Attachment	Attachment	Attachment
<input type="checkbox"/>	CampaignId	Campaign	Lookup
<input type="checkbox"/>	CloseDate	Close Date	Date
<input type="checkbox"/>	IsClosed	Closed	Boolean
<input type="checkbox"/>	CreatedById	Created By	Lookup

< Back   Next >   Cancel

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.

- Amount
- Close date
- Stage

8. Click **Next**.

Next, 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.

**Build a Quick App**  
Select Field Options.

Parent Object : Account  
Data Criteria : User Selection Filters

Select All  
☒ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Name	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Type	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

< Back Next > Cancel

9. 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 Display fields shown in the illustration below.



**X-Author Designer**

**Build a Quick App**  
Select Field Options.

Parent Object : Account

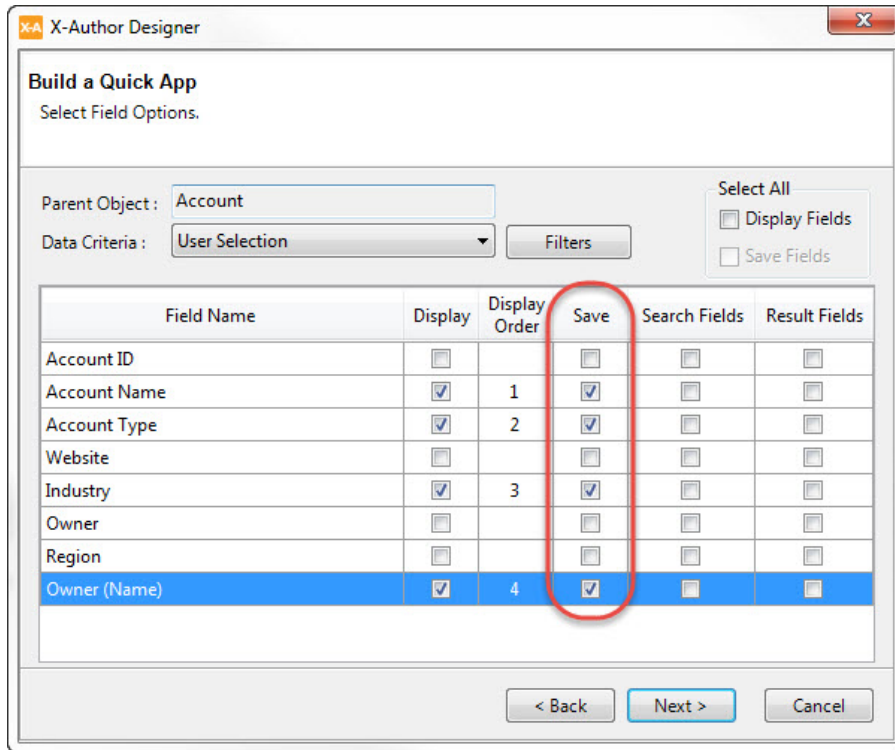
Data Criteria : User Selection Filters

Select All  
☐ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input checked="" type="checkbox"/>	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

< Back Next > Cancel

- 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.



**Build a Quick App**  
Select Field Options.

Parent Object :

Data Criteria :

Select All  
☐ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

< Back  Cancel

- 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.

**X-Author Designer**

**Build a Quick App**  
Select Field Options.

Parent Object:

Data Criteria:

Select All  
☐ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

< Back   Next >   Cancel

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

**Opportunities by Account** 1

Account Name      Acme  
Account Type      Integrator 2  
Industry      Consulting  
Owner Name      Jerret Kasper

Name	Stage	Account Name
Demo app 10/34	Prospecting	Acme
Acme Opp Gold	Prospecting	Acme
Sierra Semiconductor Deals11	Prospecting	Acme
Tesla Save	Prospecting	Acme
Acme Opp Powerplus	Prospecting	Acme
Acme Opp Platinum	Prospecting	Acme
Acme Silver 111	Needs Analysis	Acme

3

1 Worksheet title

2 Parent Object Fields

3 Child Object Fields

☐ Display Fields

☒ Save Fields

12. Specify the fields that you want your app users to be able to use as search filters when they click the corresponding check box in the Search Fields column. For this app, select the following Search fields.

**Build a Quick App**  
Select Field Options.

Parent Object : Account  
Data Criteria : User Selection

Select All  
☐ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

< Back   Next >   Cancel

13. 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 Result fields shown in the following illustration.

**X-Author Designer**

**Build a Quick App**  
Select Field Options.

Parent Object :

Data Criteria :

Select All  
☐ Display Fields  
☐ Save Fields

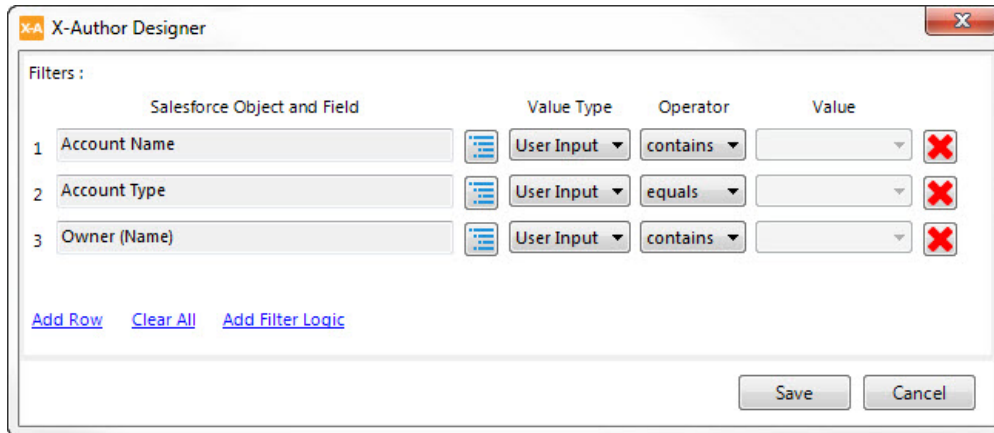
Field Name	Display	Display Order	Save	Search Fields	Result Fields
Account ID	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Account Type	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Website	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Industry	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Region	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner (Name)	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

< Back  Cancel

### 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.

- 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.




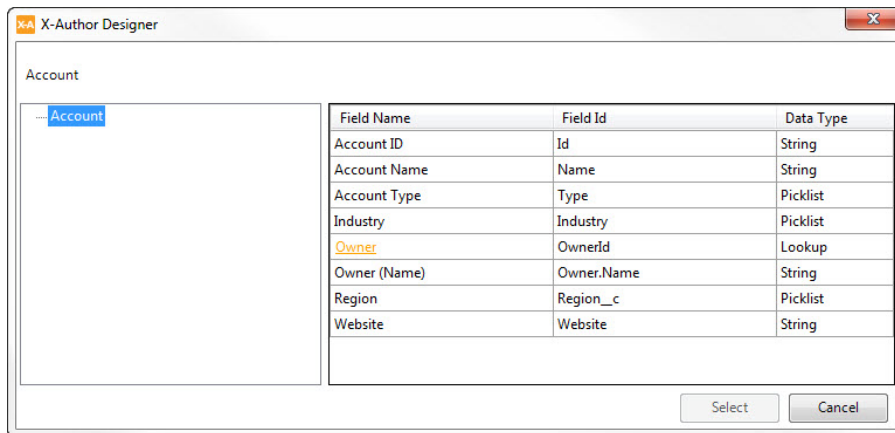
The X-Author Designer Filters dialog box shows three filters defined. Each filter has a 'Salesforce Object and Field' column, a 'Value Type' dropdown set to 'User Input', an 'Operator' dropdown, and a 'Value' field. The filters are:

	Salesforce Object and Field	Value Type	Operator	Value	
1	Account Name	User Input	contains		✗
2	Account Type	User Input	equals		✗
3	Owner (Name)	User Input	contains		✗

Below the filters are links for [Add Row](#), [Clear All](#), and [Add Filter Logic](#). At the bottom right are 'Save' and 'Cancel' buttons.

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.

15. Click the **Add Row** button to add a new filter.
16. In the new row, click the lookup icon  to select a field.



The X-Author Designer Field Selection dialog box shows a list of fields for the 'Account' object. The 'Account' object is selected in the left pane. The right pane shows a table of fields:

Field Name	Field Id	Data Type
Account ID	Id	String
Account Name	Name	String
Account Type	Type	Picklist
Industry	Industry	Picklist
Owner	OwnerId	Lookup
Owner (Name)	Owner.Name	String
Region	Region__c	Picklist
Website	Website	String

At the bottom right are 'Select' and 'Cancel' buttons.

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

**X-Author Designer**

Filters :

	Salesforce Object and Field	Value Type	Operator	Value	
1	Account Name	User Input	contains		X
2	Account Type	User Input	equals		X
3	Owner (Name)	User Input	contains		X
4	Industry	Static	equals	Agriculture	X

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

[Save](#) [Cancel](#)

18. 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.
19. Follow the same process to add another filter. This time, choose the Region field to filter on.
20. 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.

**X-Author Designer**

Filters :

	Salesforce Object and Field	Value Type	Operator	Value	
1	Account Name	User Input	contains		X
2	Account Type	User Input	equals		X
3	Owner (Name)	User Input	contains		X
4	Industry	Static	equals	Manufacturing	X
5	Region	User Input	in		X

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

[Save](#) [Cancel](#)



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.](#)

21. Click **Save**.

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

Account Name	Account Type	Industry	Owner (Name)
Acme1	Customer	Manufacturing	Jerret Kasper
ABC Corporation12	Customer	Manufacturing	Jerret Kasper
ABC Corporation, Japan	Customer	Manufacturing	Jerret Kasper
ABC Corporation International	Customer	Manufacturing	Jerret Kasper
ABC Corporation, UK	Customer	Manufacturing	Jerret Kasper

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



**Build a Quick App**  
Select Field Options.

Child Object : Opportunity  
Data Criteria : Query Filters

Select All  
☒ Display Fields  
☐ Save Fields

Field Name	Display	Display Order	Save
Opportunity ID	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Account	<input type="checkbox"/>		<input type="checkbox"/>
Name	<input type="checkbox"/>		<input type="checkbox"/>
Stage	<input type="checkbox"/>		<input type="checkbox"/>
Amount	<input type="checkbox"/>		<input type="checkbox"/>
Close Date	<input type="checkbox"/>		<input type="checkbox"/>
Owner	<input type="checkbox"/>		<input type="checkbox"/>
Account (Name)	<input type="checkbox"/>		<input type="checkbox"/>
Owner (Name)	<input type="checkbox"/>		<input type="checkbox"/>

< Back Next > Cancel

This window should look familiar to you. It looks different than the window where we configured parent-object options (it has fewer columns) because the data criteria is set to Query.

23. If the Data Criteria field is not set to Query by default, set it to Query now.
24. Select the following Display and Save fields. Note that the fields from the child object will render horizontally.

**X-Author Designer**  
Build a Quick App  
Select Field Options.

Child Object: **Opportunity** (1)  
Data Criteria: **Query** (2) **Filters** (3)

Field Name	Display	Display Order	Save
Opportunity ID	<input type="checkbox"/>		<input type="checkbox"/>
Account	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>
Name	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>
Stage	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>
Amount	<input type="checkbox"/>		<input type="checkbox"/>
Close Date	<input type="checkbox"/>		<input type="checkbox"/>
Account (Name)	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>

< Back Next > Cancel

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.

25. Click the **Filters** button.

**X-Author Designer**


Filters :

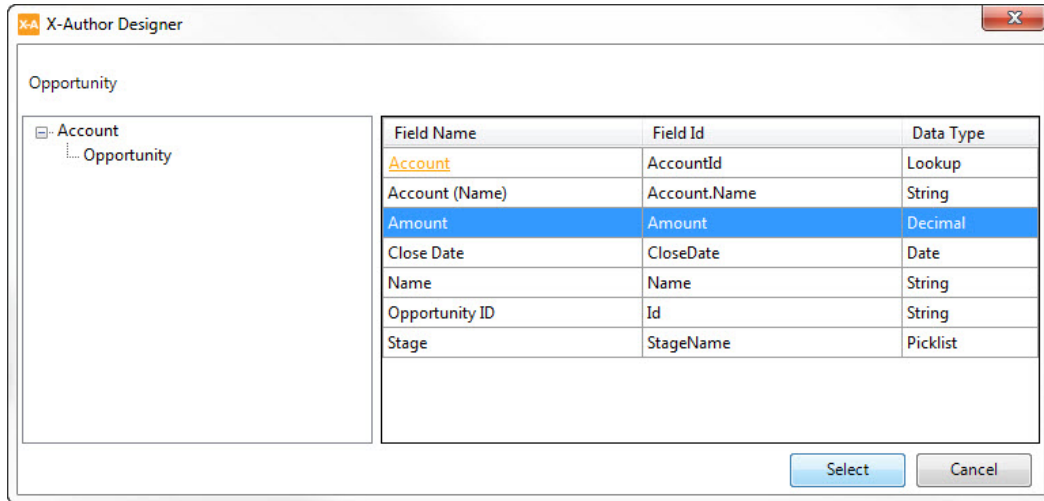
	Salesforce Object and Field	Value Type	Operator	Value	
1	Account	Input	equals	Account.Accou	✗
2					✗

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

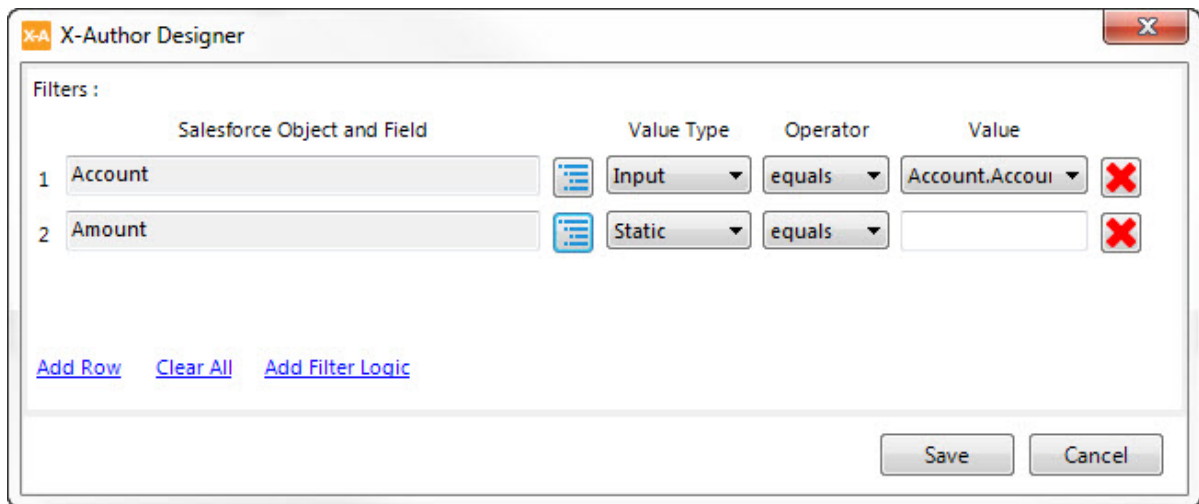
Save Cancel

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.


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



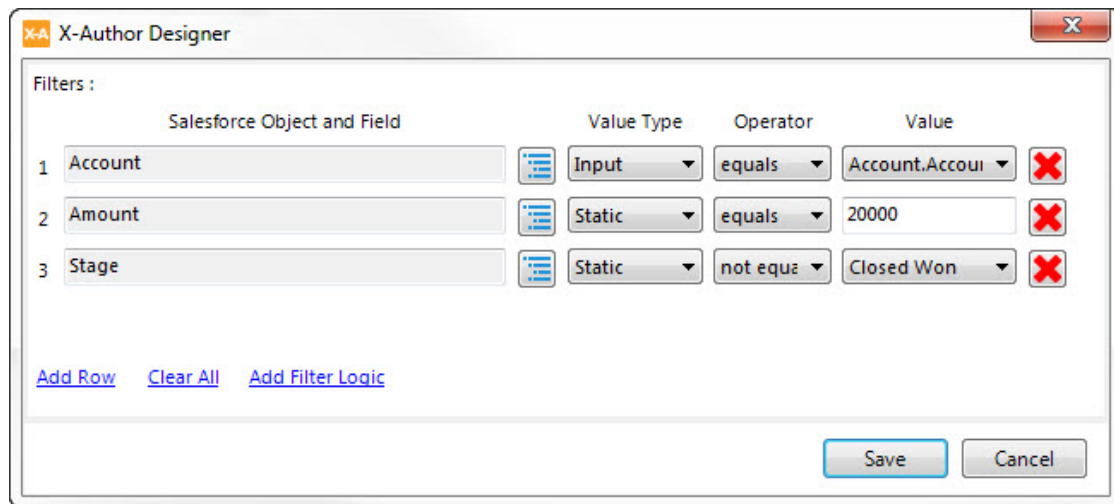
28. Click the Amount field and then click **Select**.



29. For the Amount field, click the Operator drop-down menu and select "greater than".
30. In the Value text box, type 20000.

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

Your filters should now look like this.



The screenshot shows the 'X-Author Designer' window. It contains a table with the following columns: 'Salesforce Object and Field', 'Value Type', 'Operator', and 'Value'. There are three rows of filters, each with a red 'X' icon in the rightmost column.

	Salesforce Object and Field	Value Type	Operator	Value	
1	Account	Input	equals	Account.Account	X
2	Amount	Static	equals	20000	X
3	Stage	Static	not equal to	Closed Won	X

Below the table are three links: [Add Row](#), [Clear All](#), and [Add Filter Logic](#). At the bottom right are 'Save' and 'Cancel' buttons.

36. Click **Save**.
37. Click **Next**. The App Settings window opens.

**X-Author Designer**

**Build a Quick App**  
Provide App Settings.

**General Settings**

☐ Enable App Launch from Salesforce

**Worksheet Settings**

Worksheet Title :

Max Column Width :

☐ View Gridlines

☒ Display Filters

**Menu Settings**

Group Label :

Retrieve Data Button Label :

Save Data Button Label :

☐ Enable Add Row

☐ Enable Delete Row

< Back    Next >    Finish

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.

38. 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.](#))
39. In the Worksheet Title text box, type Opportunities by Account. Remember, this title will appear in field A1 on your Excel workbook.
40. Change the Group Label to "Opportunities."
41. Click the Enable Add Row check box.
42. Click Finish.

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

## Adding Salesforce Objects to your App



Before you can add objects to your app, you need to create the app file. Refer to [Designing Apps](#) for instructions to create your app.

Every app you design uses objects and fields from Salesforce. Complete the tasks on this page to add objects to your app and define which fields from those objects you want to work with.

### Object Behavior

All objects used in an app are assigned a behavior that determines their availability in maps and actions. There are two behaviors:

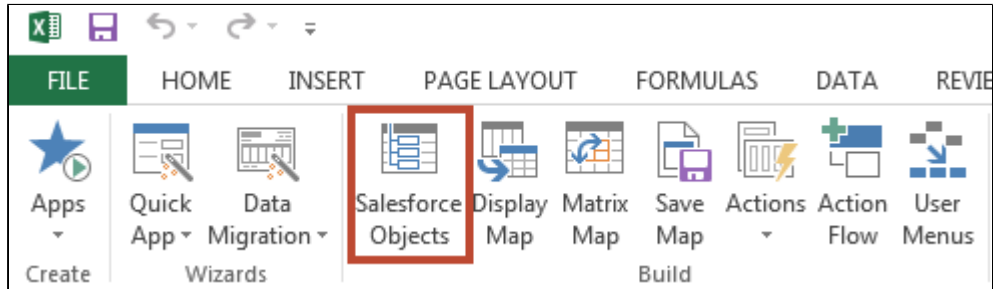
- **Individual:** Work with an individual record at a time for that object.
- **List:** Work with more than one record for that object.

### 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.

## To add objects and fields

1. From the X-Author Designer ribbon, click the **Salesforce Objects** button.

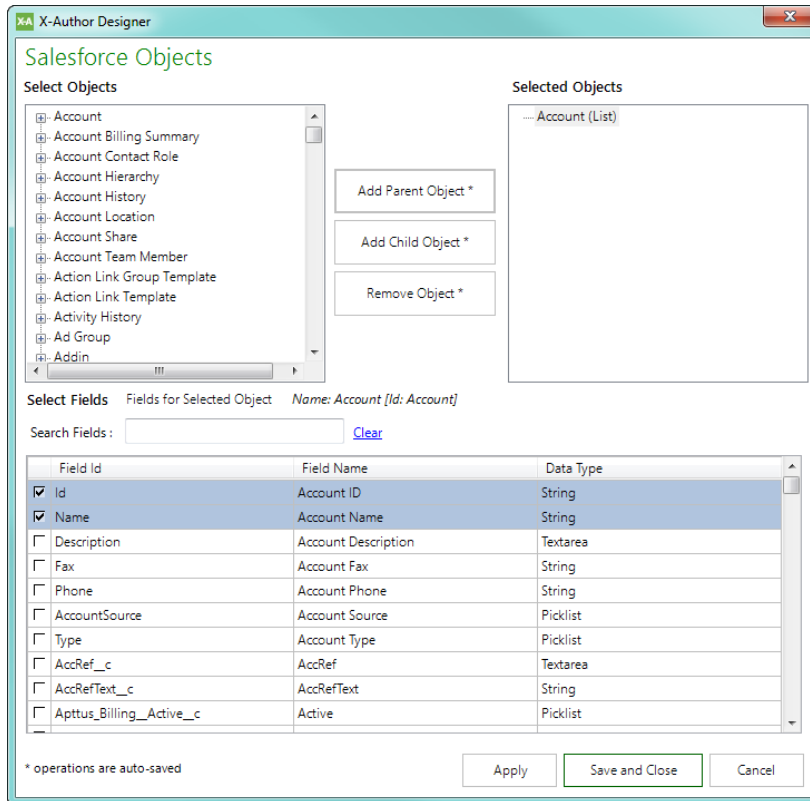


2. From the left pane under **Select Objects**, choose an object or drill down to other objects by clicking the **+** next to the object name.



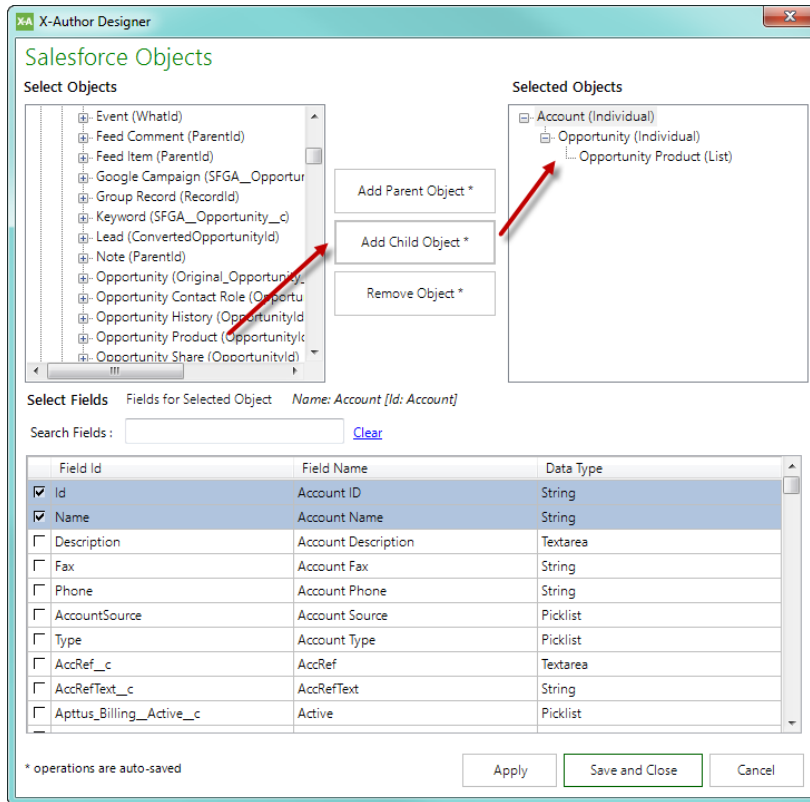
Type the first letter of the Object you want to add and X-Author will jump to the first matching object in the list.

3. To select a top-level (header level) object, click **Add Parent Object** to move the object to Selected Objects.



4. If you need to add one or more child objects, drill down from the Parent object one or more levels, select the child object and click **Add Child Objects** to add objects in the hierarchy (from child up to parent) to Selected Objects.

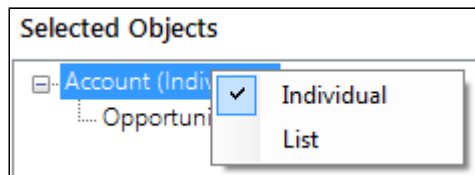




## Changing Object Behavior

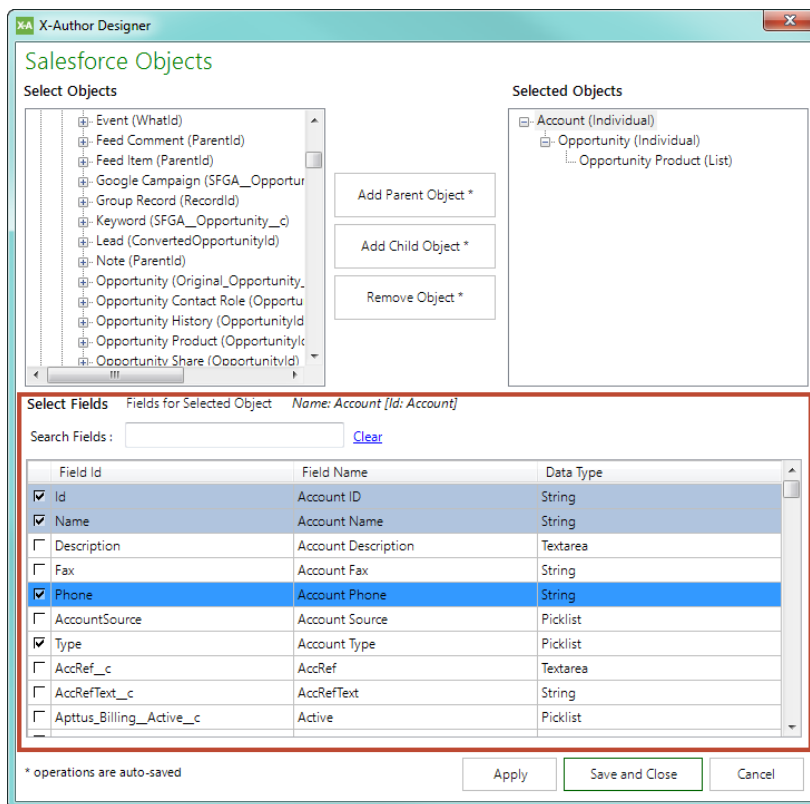
By default, when objects are added in a parent-child relationship, parent objects adopt the "Individual" behavior and the bottom child object in the hierarchy adopts the "List" behavior.

You can manually change the behavior of any object by right-clicking on the object name and choosing either option from the drop-down that is displayed (for example, if you want two objects to have a many-to-many relationship, make both objects "List").



**Caution:** Be careful when changing object behavior. If you have already created other parts of the app, changing object behavior may not make sense in some circumstances.

- Click **Remove Object** to remove a selected object from the list.
- From Select Fields, scroll through the list of fields for the object and select the check box next to one or more fields to add them to the app. You can also type the name of the field into the Search Fields dialog to narrow the list of displayed fields.



### Note

Click on the heading for any column in the fields table to sort fields by that category.

When a picklist field is selected, all picklist values will automatically be retrieved during runtime. Regular, dependent, and multi-select picklists are supported.

The following fields are automatically selected for each object and cannot be deselected.

- ID
- Name, if there is a Name field
- Look-up for a parent object that is selected as part of the object hierarchy (for example, "AccountId" for Opportunity).

7. Click **Apply** after you have finished adding fields for an object. You *must* click Apply as you move between objects or your selections will not be saved.
8. When you finished adding objects and fields to your app, click **Save and Close**.

### **Best Practices for adding Salesforce Objects**

Be careful when removing objects if your app already contains Display Maps or other components. 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.

### **Use Case: Adding Salesforce objects for a Sales Pipeline app**

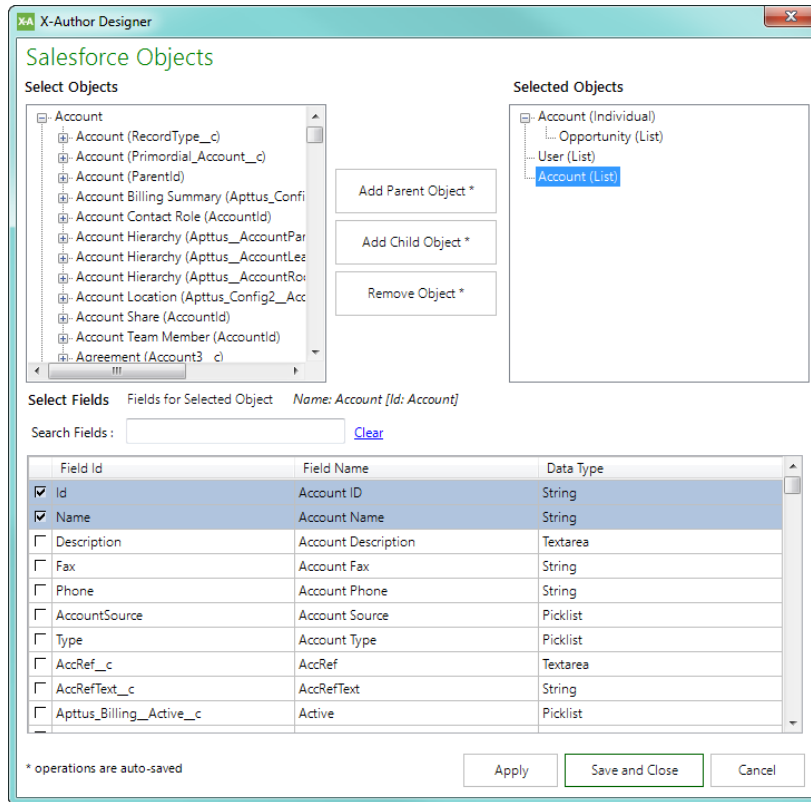
Danny Designer needs to create an app that will be used to manage Opportunity data for a Sales Pipeline.

He decides that his app is going to need data from the Account, Opportunity and User objects. Opportunity will be added as a child object of Account, and Account and User will be added separately as List objects.

Danny takes the following steps to add Salesforce objects to his app:

1. He goes to **Apps > New** and creates a new app named "Pipeline CC."
2. Danny clicks **Salesforce Objects**.

3. He finds the Opportunity child object under the Account object and clicks **Add Child Object**.
4. Next he clicks **Add Parent Object** for User and Account.



5. Danny selects fields for each object. After selecting fields for any object, he clicks **Apply** to save his selections.
  - a. For **Account (Individual)** and **Account (List)**: *Account Description, Account Phone, Account Type, Annual Revenue, Number of Employees (Account Name and Account Id are added by default)*

**Select Fields** Fields for Selected Object Name: Account [Id: Account]

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	Description	Account Description	Textarea
<input checked="" type="checkbox"/>	Id	Account ID	String
<input checked="" type="checkbox"/>	Name	Account Name	String
<input checked="" type="checkbox"/>	Phone	Account Phone	String
<input checked="" type="checkbox"/>	Type	Account Type	Picklist
<input checked="" type="checkbox"/>	AnnualRevenue	Annual Revenue	Decimal
<input checked="" type="checkbox"/>	NumberOfEmployees	Employees	Double
<input type="checkbox"/>	Fax	Account Fax	String
<input type="checkbox"/>	AccountSource	Account Source	Picklist
<input type="checkbox"/>	AccRef_c	AccRef	Textarea

- b. For **Opportunity (List)**: *Amount, Description, Stage Name, Is Closed, Close Date, Owner Id (Account (Lookup), Opportunity Id and Opportunity Name* are added by default)

**Select Fields** Fields for Selected Object Name: Opportunity [Id: Opportunity]

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	AccountId	Account	Lookup
<input checked="" type="checkbox"/>	Amount	Amount	Decimal
<input checked="" type="checkbox"/>	CloseDate	Close Date	Date
<input checked="" type="checkbox"/>	IsClosed	Closed	Boolean
<input checked="" type="checkbox"/>	Description	Description	Textarea
<input checked="" type="checkbox"/>	Name	Name	String
<input checked="" type="checkbox"/>	Id	Opportunity ID	String
<input checked="" type="checkbox"/>	StageName	Stage	Picklist
<input type="checkbox"/>	Alt_Account_c	Alt Account	Lookup
<input type="checkbox"/>	Approval_Comments_c	Approval Comments	Textarea

- c. For **User (List)**: *Active (User Id and Full Name* are added by default)

Select Fields Fields for Selected Object Name: User [Id: User]

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	IsActive	Active	Boolean
<input checked="" type="checkbox"/>	Name	Full Name	String
<input checked="" type="checkbox"/>	Id	User ID	String
<input type="checkbox"/>	AboutMe	About Me	Textarea
<input type="checkbox"/>	AccountId	Account	Lookup
<input type="checkbox"/>	UserPreferencesActivityRemindersPop...	ActivityRemindersPopup	Boolean
<input type="checkbox"/>	Address	Address	NotSupported
<input type="checkbox"/>	ReceivesAdminInfoEmails	Admin Info Emails	Boolean
<input type="checkbox"/>	Alias	Alias	String
<input type="checkbox"/>	ForecastEnabled	Allow Forecasting	Boolean

- Danny clicks **Save and Close** to save his objects and fields to the app.
- Finally, he click the **Save App** button in the X-Author Designer ribbon to update the app file in Salesforce.

## Working with Display Maps

Use Display Maps to **link Salesforce fields to Excel cells** for data that will be retrieved from Salesforce by an app. When data is retrieved from Salesforce by a query or similar action, this data is temporarily stored in Excel memory before it is placed into the worksheet. X-Author then uses a Display Map to place the data in the correct Excel cells. You use **Display Actions** in your app to implement Display Maps.

How many Display Maps you use is dependent on app design and the timing of data retrieval in your app. Remember the following when working with Display Maps:

- There is no limit to the number of Display Maps in a single app.
- A single Display Map can include fields from multiple objects.
- A single Display Map can be used across multiple worksheets (not workbooks).

### Note

You do not need a Display Map if your app is not being designed to retrieve data from Salesforce. If you are retrieving data from Salesforce, then you need at least one Display Map per app.

## Prerequisites

Before you can create a Display Map, you must have defined at least one Salesforce object for your app. Refer to [Adding Salesforce Objects to your App](#) for instructions.

## Display Maps and Object Behavior

When you added Salesforce objects to your app, you defined them as Individual (one record) or List (one or more records). When you map fields in a Display Map, you also choose whether the field type will be Individual or List. The behavior selected for an object determines how its fields are used in a Display Map.

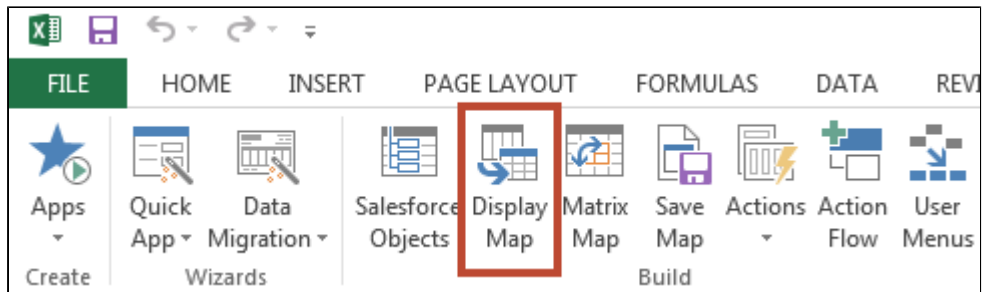
**Individual** object fields can be placed anywhere in the worksheet except in the same row that contains fields from a List object.

List object fields must all be placed in the same row. This forms a **List Range**. List object fields 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.

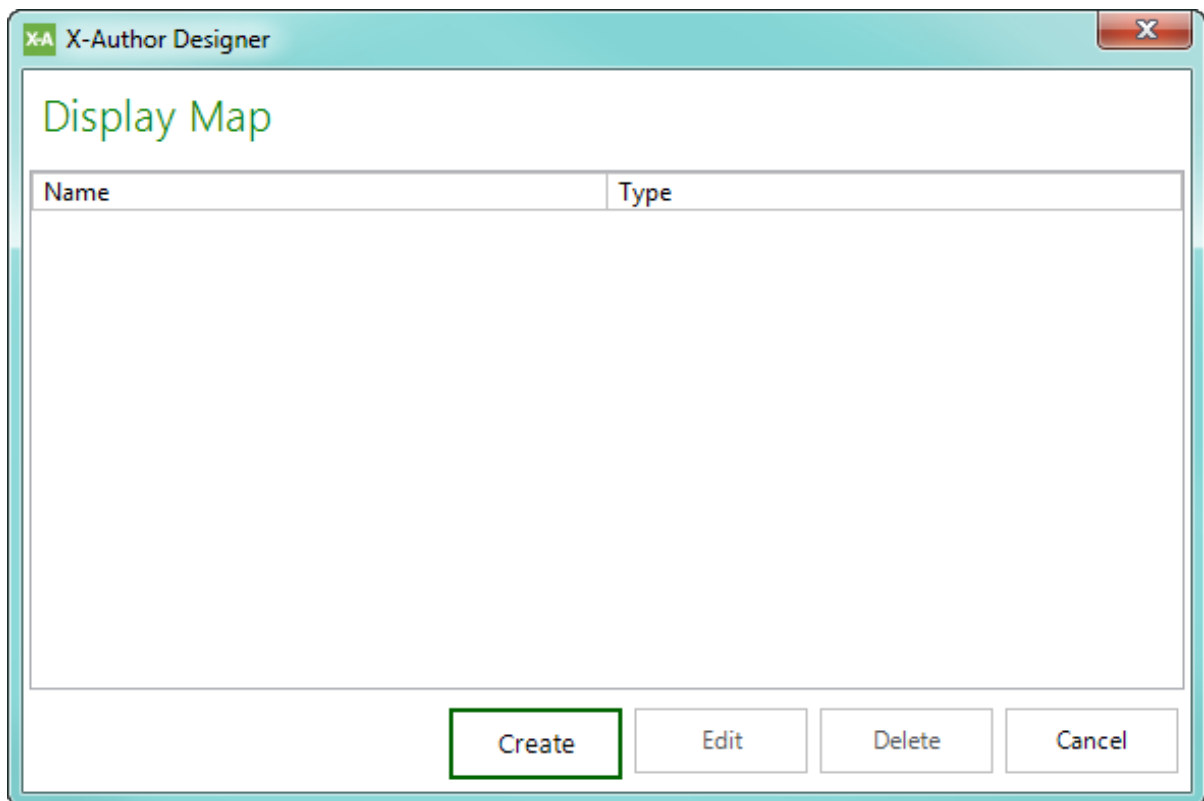
You can find more information on **Individual** object fields and **List** object fields in the following sections.

## To create a Display Map for an Individual object

1. From, the X-Author Designer ribbon, click **Display Map**.



2. Click **Create**. The Display Map panel opens in Excel.



3. In the Name field, enter a name for your Display Map. Pick a name that makes it easy to distinguish from your other Display Maps.
4. Select **Individual** from the Type picklist.



**Display Map** ▼ ✕

Name:  Type:

Search:  [Clear](#)

**Fields**

Layout: ☒ Vertical ☐ Horizontal

☒ Account

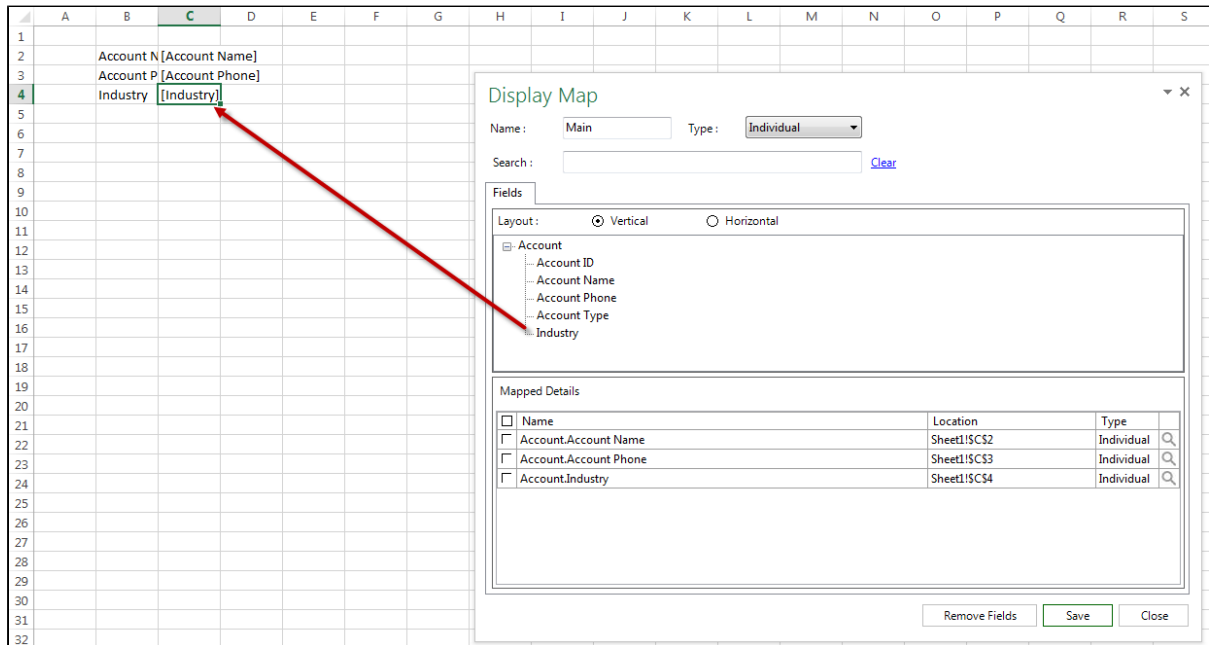
**Mapped Details**

<input type="checkbox"/> Name	Location	Type



Click on the ▼ icon in the upper right-hand corner and choose an option to move or resize the Display Map panel.

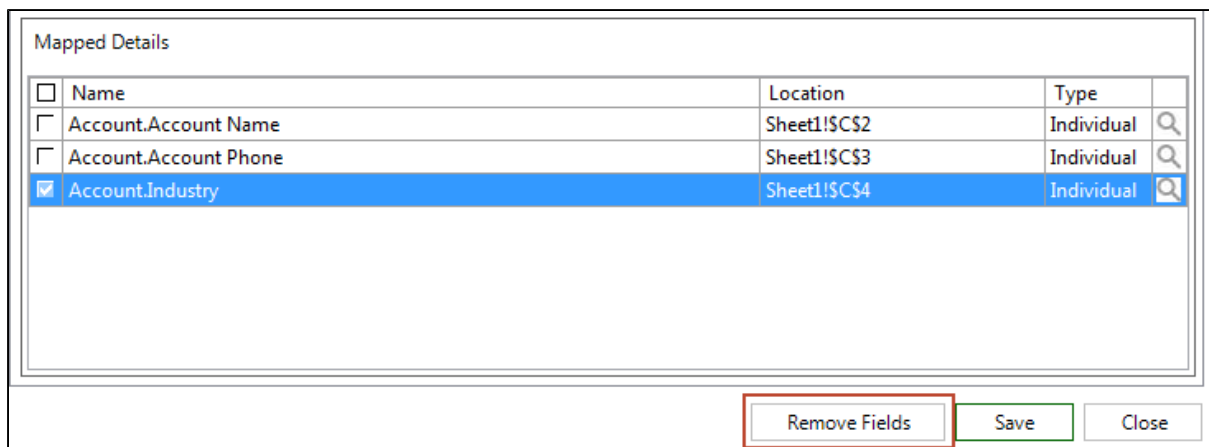
5. Add a field to the Display Map by dragging it from the Display Map panel and dropping it on the worksheet. The field label is automatically inserted into the cell directly to the left of the mapped cell. You can search for specific fields using the Search box.




### Best Practice

To make the Display Map more appealing to the end user, highlight a cell containing a mapped field displayed in brackets and press the Delete key. Never delete rows to remove these values as it will corrupt the spreadsheet.

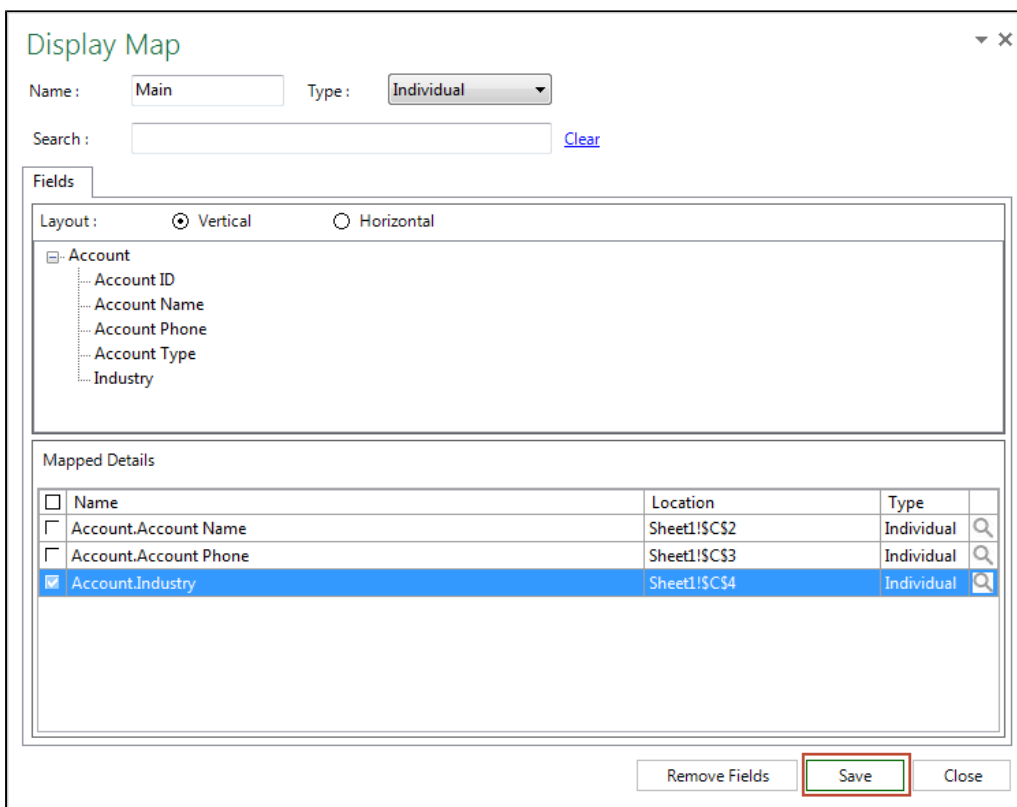
6. To remove a mapped field from the worksheet, select the check box next to the field in the Mapped Details section of the Display Map and click **Remove Fields**.



## Display Map Look Aheads

You can click the magnify icon  next to any mapped field to create a **Look Ahead**. This feature allows the runtime user to double-click on any cell in the retrieved list of data to lookup specific record values based on how you configure the Look Ahead. For information on the different types of Look Aheads and how to add them to your app design, refer to [Creating Look Aheads](#) in the Advanced App design section of this guide.

- After you have mapped all Individual object fields to your worksheet, click **Save**.



**Display Map**

Name:  Type:

Search:  [Clear](#)




**Fields**

Layout: ☒ Vertical ☐ Horizontal

Account

- Account ID
- Account Name
- Account Phone
- Account Type
- Industry

**Mapped Details**

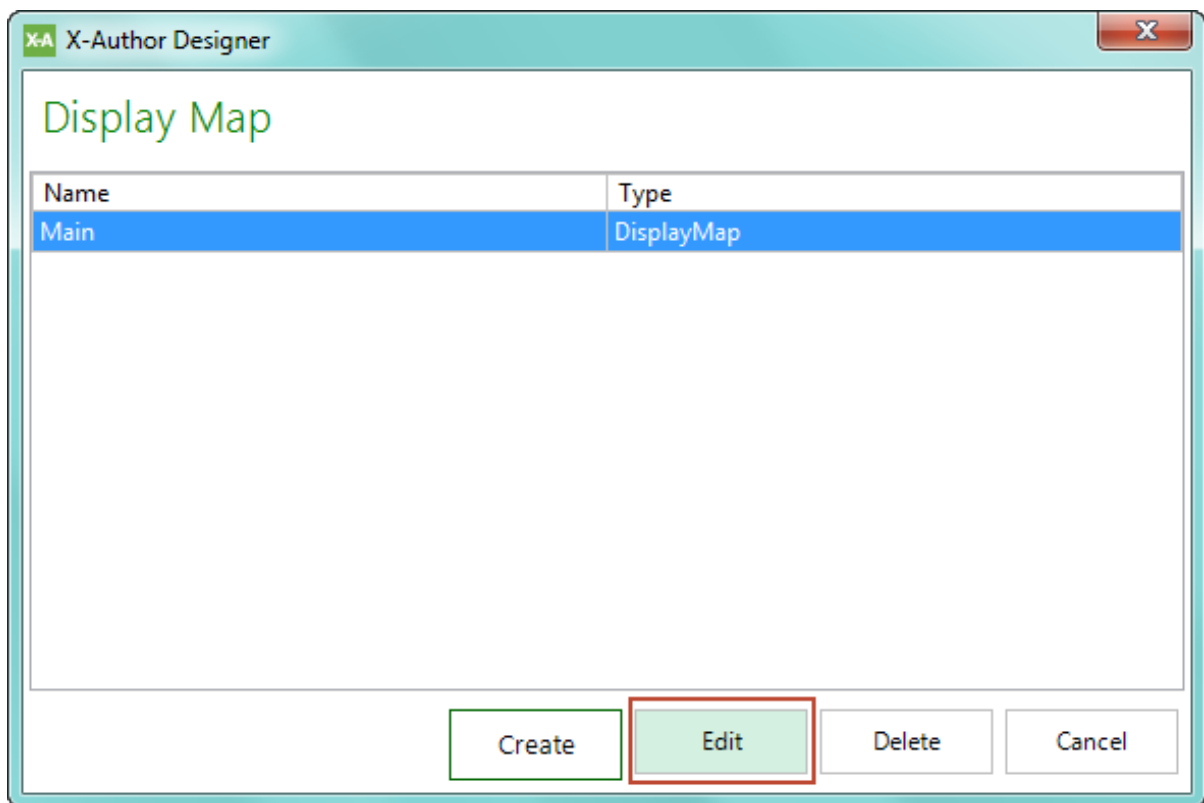
<input type="checkbox"/>	Name	Location	Type	
<input type="checkbox"/>	Account.Account Name	Sheet1!\$C\$2	Individual	
<input type="checkbox"/>	Account.Account Phone	Sheet1!\$C\$3	Individual	
<input checked="" type="checkbox"/>	Account.Industry	Sheet1!\$C\$4	Individual	

- Click **Close**.
- Click **Save App** to save your app.

## To map List object fields to a Display Map

**i** Note: In this task, you will map List fields to the same Display Map created in the previous task. You can also create a new Display Map for your List fields, but it is often easier, especially with parent-child apps, to use both Individual and List field types in your Display Map.

1. From the X-Author Designer ribbon, click **Display Map**.



2. Click **Edit**. The Display Map panel opens in Excel.
3. Select **List** from the Type picklist. All List objects and fields defined in Salesforce objects are displayed in the panel.

**Display Map** ▼ ✕

Name:  Type:

Search:  [Clear](#)

**Fields** **Options**

Layout: ☒ Vertical ☐ Horizontal

☒ Account.Opportunity
 

- ☒ Account(Account)
  - Account (Name)
  - Amount
  - Close Date
  - Name
  - Opportunity ID
  - Stage

**Mapped Details**

<input type="checkbox"/> Name	Location	Type	
<input checked="" type="checkbox"/> Account.Account Name	Sheet1!\$C\$3	Individual	
<input type="checkbox"/> Account.Account Phone	Sheet1!\$C\$4	Individual	
<input type="checkbox"/> Account.Industry	Sheet1!\$C\$5	Individual	

4. Select the **Vertical** check box so that the list of displayed records will be shown from top to bottom. Horizontal layouts are covered in a later section.
5. Add fields to the Display Map by dragging them from the Display Map panel and dropping them onto the worksheet in the same row. You can search for specific fields using the Search box. Fields are not required to be adjacent to each other, but there must be a start cell and end cell, which forms a **List Range**. You can also add fields from related objects (such as Account, if you are creating a list of Opportunities).

The screenshot shows an Excel spreadsheet with a table titled "Opportunities by Account". The table has columns: Account Name, Account Phone, Industry, Name, Amount, Close Date, and Stage. The "Display Map" dialog box is open, showing a list of fields: Account (Name), Amount, Close Date, Name, and Opportunity ID. A red arrow points from the "Amount" field in the dialog to the "Amount" column in the spreadsheet.

## Working with List Ranges

The following rules and guidelines apply when you are mapping 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*.

<b>List Rows are Inserted</b>	<ul style="list-style-type: none"> <li>As each List Salesforce record is retrieved and displayed, an Excel row is inserted in the workbook.</li> <li>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.</li> </ul>
<b>Fields from related objects</b>	<ul style="list-style-type: none"> <li>Fields selected from related objects in the hierarchy can also be included on the same row in the List Range. You can elect to save related object fields by including them in the Save Map. See <a href="#">Working with Save Maps</a> for more information.</li> </ul>
<b>List Ranges for different objects</b>	<ul style="list-style-type: none"> <li>These can be created below other ranges or on other worksheets.</li> </ul>
<b>Grouping and sorting</b>	<ul style="list-style-type: none"> <li>Map fields that you want to group by at the left of the list range. See Grouping and Sorting List Ranges for more information.</li> </ul>

- Click **Close**.
- Click **Save App** to save your app.

## Rich Text Fields in List Ranges

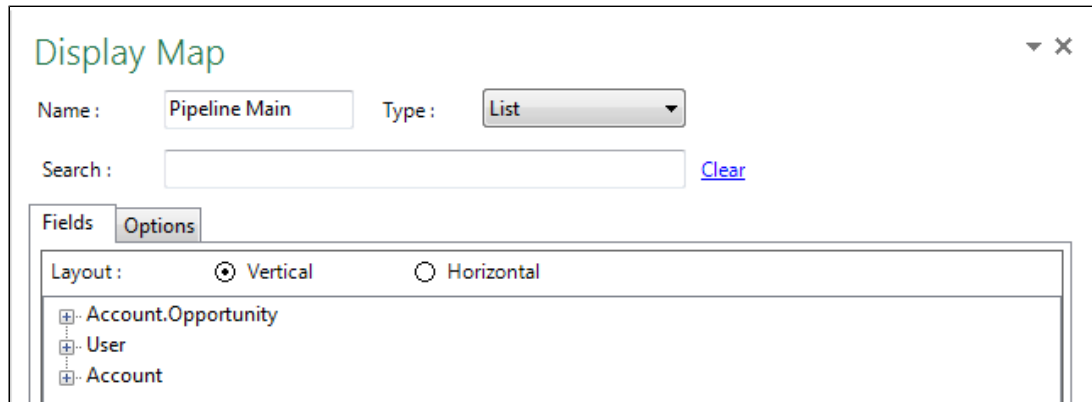
You can insert Rich Text fields in the same way you enter any other field. Rich text fields appear to runtime users as a single link. The link opens an editor that allows the runtime user to modify the Rich Text field. Rich Text fields can only be edited by the runtime user if it is part of a list range (List object field). Rich Text fields of the SaveOnly type cannot be edited.

## Use Case: Sales Pipeline Display Map Part 1

Danny Designer is designing an app with Account, Opportunity and User objects for sales persons to use to manage their sales pipeline. He has already selected the Salesforce objects for the app and now needs to create a Display Map to link data from Salesforce to the cells in the Excel workbook.

Danny takes the following steps to create a Display Map:

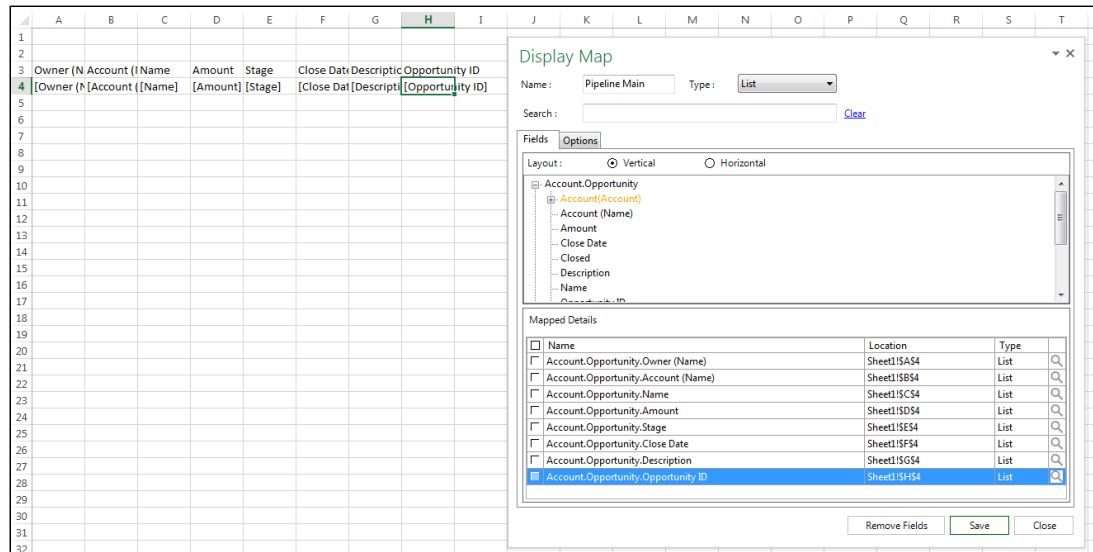
1. Danny connects to Salesforce and opens the Sales Pipeline app.
2. He clicks **Display Map** and clicks **Create** to start a new Display Map.
3. Danny enters the name "Pipeline Main" for his Display Map and selects **List** from the Type drop-down.



The screenshot shows the 'Display Map' configuration window. At the top, the title 'Display Map' is in green. Below it, there are two input fields: 'Name:' with the value 'Pipeline Main' and 'Type:' with a dropdown menu showing 'List'. To the right of the 'Type' dropdown is a small 'x' icon. Below these fields is a 'Search:' input field and a 'Clear' link. Underneath is a tabbed interface with 'Fields' and 'Options' tabs. The 'Fields' tab is active, showing a list of fields: 'Account.Opportunity', 'User', and 'Account'. Each field has a small icon to its left. Above the list, there is a 'Layout:' section with two radio buttons: 'Vertical' (selected) and 'Horizontal'.

4. Next, he drags and drops fields from the Account.Opportunity child object list to map them to the Display Map.





- Danny chooses to **Hide** column H. He added the field so that the user can save data back to Salesforce but he does not want to display it in the worksheet.
- Next, Danny adds a formula in D6 to SUM the Opportunity Amounts in D4.

D6    :          =SUM(D4,D5)							
	A	B	C	D	E	F	G
1							
2							
3	Owner (Name)	Account (Name)	Name	Amount	Stage	Close Date	Description
4	[Owner (Name)]	[Account (Name)]	[Name]	[Amount]	[Stage]	[Close Date]	[Description]
5							
6			Total Amount	0			
7							

- Danny clicks **Save** on the Display Map Panel and clicks **Close**.
- Danny clicks **Save App** to save all changes to the app.

## Grouping and Sorting List Ranges

After mapping List fields to a Display Map, you can define List Range sorting and group from the Display Map Options tab. In many cases, native Excel sorting and grouping options are more flexible.

## To sort and group List Ranges

1. From the Display Map panel, click on the **Type** picklist and select "List."
2. Click the **Options** tab.
3. (Optional) Type a description in the Header field to display above the label row of the list range.
4. For **Sort by field 1** and **Sort by field 2**, select an individual column to sort by. Choose whether records will be sorted in Ascending or Descending order.
5. For **Group by field 1** and **Group by field 2**, select a column in the repeating range to group by.

### **Best Practice**

Fields you intend to Group by should be mapped at the leftmost side of the List Range when you create a Display Map.

6. Click Save.

**Display Map**

Name :  Type :

**Fields** Options

Objects :

Header :

Layout :

Sort by field 1:

Sort by field 2:

Group by field 1:

Group by field 2:

Total fields

### **① Grouping and Sorting support for numbers, decimals and currency**

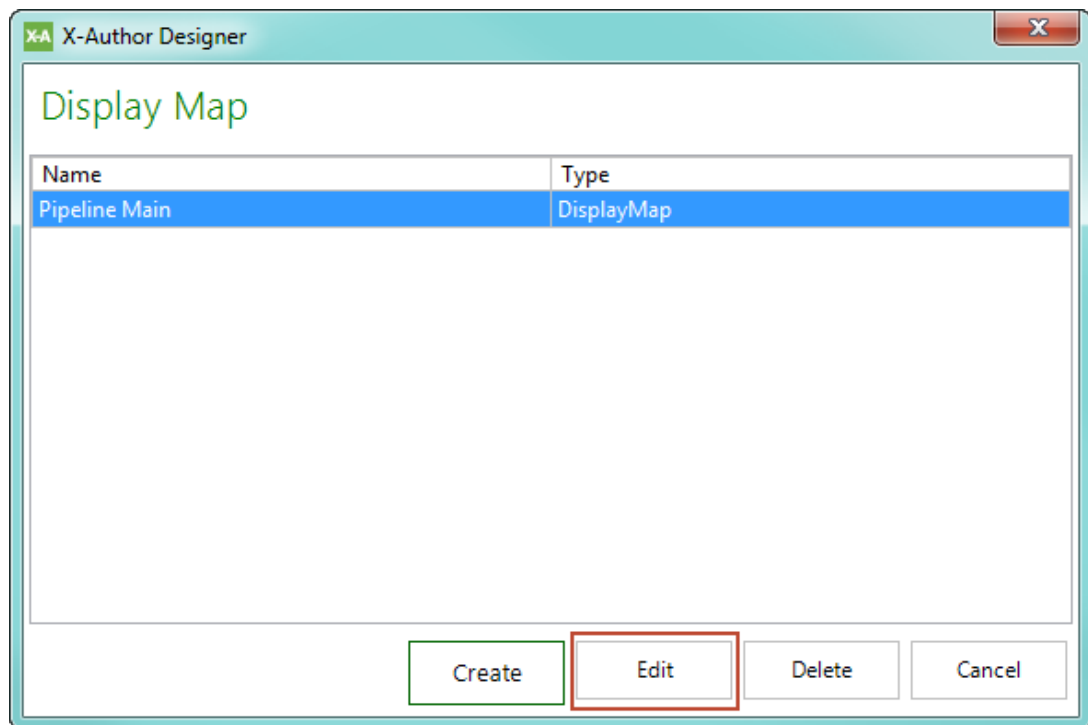
Support is included for both single and two levels of grouping in Display Maps:

- You can total all numbers, decimals and currency fields
- You can create totals on all Formula fields. Totals are displayed for formula fields having datatype as number, decimal and currency.

Account	Product	List Price
		1,000.00000
		900.00000
		5.00000
		4.00000
		12,995.00000
		12,995.00000
	Total	27,899.00000
	01t15000004sV/rAAM	
		20,100.00000
	Total 01t15000004sV/rAAM	20,100.00000
	01t15000004sV/sAAM	
		200.00000
		200.00000
	Total 01t15000004sV/sAAM	400.00000

## ① Use Case: Sales Pipeline Display Map Part 2

1. Danny decides that the Display Map he created (see [Working with Display Maps](#)) needs some grouping and sorting options. He clicks on **Display Map**, selects the "Pipeline Main" map and clicks **Edit**.



2. From the Display Map panel, Danny clicks on the **Type** picklist and selects "List."
3. He clicks on the **Options** tab in the Display Map panel and selects the following options:
  - Objects: Account.Opportunity
  - Layout: Vertical
  - Sort by field 1: Account.Opportunity.Name
  - Group by field 1: Account.Opportunity.Owner (Name)
  - Group by field 2: Account.Opportunity.Account (Name)

4. To improve the appearance of the app, Danny removes the bracketed text from each cell in row 4, removes the gridlines and adds some color to the app.

	A	B	C	D	E	F	G
1	Sales Pipeline Opportunities						
2							
3	Owner (Name)	Account (Name)	Name	Amount	Stage	Close Date	Description
4							
5							
6				Total Amount			0
7							
8							

5. He clicks **Save** on the Display Map and **Close**.
6. Danny clicks **Save App** to save all changes to the app.

## 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 as 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.

## Moving and Removing Mapped Fields

After a cell has been mapped to a field in Excel, it is critical that X-Author be able to find that location in runtime. X-Author can easily adjust its map links when fields need to be relocated, provided these changes are performed correctly. Follow the tasks in this section to properly move or remove fields from Display Maps and Save Maps to preserve cell mapping.



Do not use Excel's Delete Row or Delete Column functions to manually delete a row, field or column in the worksheet. Cells will no longer be correctly mapped and your app will be corrupted.

## To remove a field from a Display Map

1. From the X-Author Designer ribbon, click **Display Map**.
2. Select the Display Map you want to change and click **Edit**.
3. Under Mapped Details, select the field to be removed. If you want to remove all the fields in the map, select the check box next to the left of the Name column.

Mapped Details				
<input type="checkbox"/>	Name	Location	Type	
<input type="checkbox"/>	Account.Opportunity.Owner (Name)	Pipeline!\$A\$4	List	
<input type="checkbox"/>	Account.Opportunity.Account (Name)	Pipeline!\$B\$4	List	
<input type="checkbox"/>	Account.Opportunity.Name	Pipeline!\$C\$4	List	
<input type="checkbox"/>	Account.Opportunity.Amount	Pipeline!\$D\$4	List	
<input type="checkbox"/>	Account.Opportunity.Stage	Pipeline!\$E\$4	List	
<input type="checkbox"/>	Account.Opportunity.Close Date	Pipeline!\$F\$4	List	
<input type="checkbox"/>	Account.Opportunity.Description	Pipeline!\$G\$4	List	
<input type="checkbox"/>	Account.Opportunity.Opportunity ID	Pipeline!\$H\$4	List	
<input checked="" type="checkbox"/>	Account.Opportunity.Closed	Pipeline!\$I\$4	List	

4. Click **Remove Fields**.

Mapped Details

<input type="checkbox"/>	Name	Location	Type	
<input type="checkbox"/>	Account.Opportunity.Owner (Name)	Pipeline!\$A\$4	List	
<input type="checkbox"/>	Account.Opportunity.Account (Name)	Pipeline!\$B\$4	List	
<input type="checkbox"/>	Account.Opportunity.Name	Pipeline!\$C\$4	List	
<input type="checkbox"/>	Account.Opportunity.Amount	Pipeline!\$D\$4	List	
<input type="checkbox"/>	Account.Opportunity.Stage	Pipeline!\$E\$4	List	
<input type="checkbox"/>	Account.Opportunity.Close Date	Pipeline!\$F\$4	List	
<input type="checkbox"/>	Account.Opportunity.Description	Pipeline!\$G\$4	List	
<input type="checkbox"/>	Account.Opportunity.Opportunity ID	Pipeline!\$H\$4	List	
<input checked="" type="checkbox"/>	Account.Opportunity.Closed	Pipeline!\$I\$4	List	

Remove Fields

Save

Close

5. Click **Save**.

6. Click **Close**.

When you remove a field from a Display Map that is already part of a Save Map, the field will be removed from the Save Map as well.

## To move a field from one cell to another

1. Close all open Display Maps and Save Maps.
2. Place your cursor in the cell to be moved. Right-click on the cell and select **Cut**.
3. Place your cursor in the target cell. Make sure there is not already a mapped field in that cell. Right-click on the cell 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 App**. This last step relocates the field mapping.



6. Open the Display Map or Save Map that contains the mapped field or fields. The Mapped Details pane shows the new mapped cell locations.

### To move a column

1. Close all open Display Maps and Save Maps.
2. Highlight the column. Right-click and select **Cut**.
3. Highlight the target column. Right-click and select **Paste**.
4. Click **Save App**.
5. Open the Display Map or Save Map and check the Mapped Details to confirm cells were properly remapped.

### To insert or delete rows or columns

1. Close all open Display maps.
2. Place your cursor on the row or column heading where you want to perform an insert/delete.
3. Right-click on the row or column and select **Insert** or **Delete**.
4. Click **Save App**.
5. Open the Display Map or Save Map to confirm cells were properly remapped.

### To delete an existing Display Map



You cannot delete a Display Map that is being used by a [Display Action](#).

1. From the ribbon, click the **Display Map** button.
2. From the list of Display Maps, select a Display Map and click **Delete**.

## Working with Save Maps

Use **Save Maps** to link Salesforce fields to Excel cells for data that will be saved to Salesforce. Save Maps often rely on Display Maps (see [Working with Display Maps](#)). You use [Save Actions](#) in your app to implement Save Maps.

- [Saving Data to Salesforce](#)
  - [Updates](#)
  - [Inserts](#)
  - [Deletes](#)
- [Working with Save Maps](#)
  - [To create a Save Map for Display Map Fields](#)
  - [To create a Display Map for Other Fields](#)
  - [To delete an existing Save Map](#)
- [Creating Preloaded Grids](#)
  - [To create a Preloaded Grid](#)

### Saving Data to Salesforce

X-Author for Excel supports updates, inserts, upserts, and deletes.

#### Updates

Save Maps can be designed to update records in Salesforce in three ways:

- Records were retrieved from Salesforce and the data from one or more fields in that record were placed on the worksheet through a Display Map. Using the **Add Display Map Fields** function in a Save Map, user modifications to any cell can be updated on the record.
- Using the **Add Other Fields** function to save data from one or more fields that were not retrieved from Salesforce. It is not recommended to update fields to Salesforce using this function, but if you do update in this way, you must always include a Record ID in the same row as the fields you are updating.

- Using the **Add Matrix Map Section** function in a Save Map, modifications to matrix map sections can be updated on the record.

## Inserts

New records can be created in three ways:

- When a user adds a new data row using the **Add Row** menu button.
- When the App automatically adds a new data row using the **Add row Action** and pastes data into the new row.
- Using a **Preloaded Grid** (see [Creating Preloaded Grids](#)).

## Deletes

Rows can be deleted from Display Map lists by the user clicking the Delete Rows from Salesforce menu button. To delete one or more records from a List, 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.

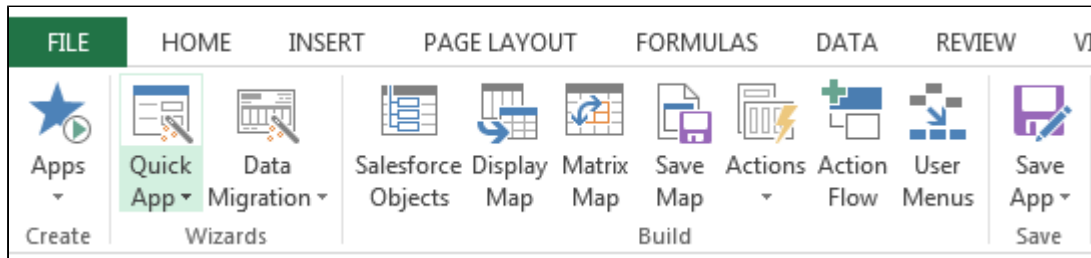
## Working with Save Maps

How many Save Maps you use is dependent on app design and how many save options you have for the user at runtime. Remember the following when working with Save Maps:

- There is no limit to the number of Save Maps you can have in an app. Multiple Save Maps allows you to control which fields are updated when the user clicks a menu button associated with a specific save process.
- A Save Map can include fields from any number of objects.
- A Save Map can be used across multiple worksheets (not workbooks).

## To create a Save Map for Display Map Fields

1. From the X-Author ribbon, click **Save Map**.



2. Click **Create**. The Save Map panel opens in Excel.
3. In the Name field, enter a name for your Save Map. Pick a name that makes it easy to distinguish from your other Save Maps.
4. Click **Add Display Map Fields**.

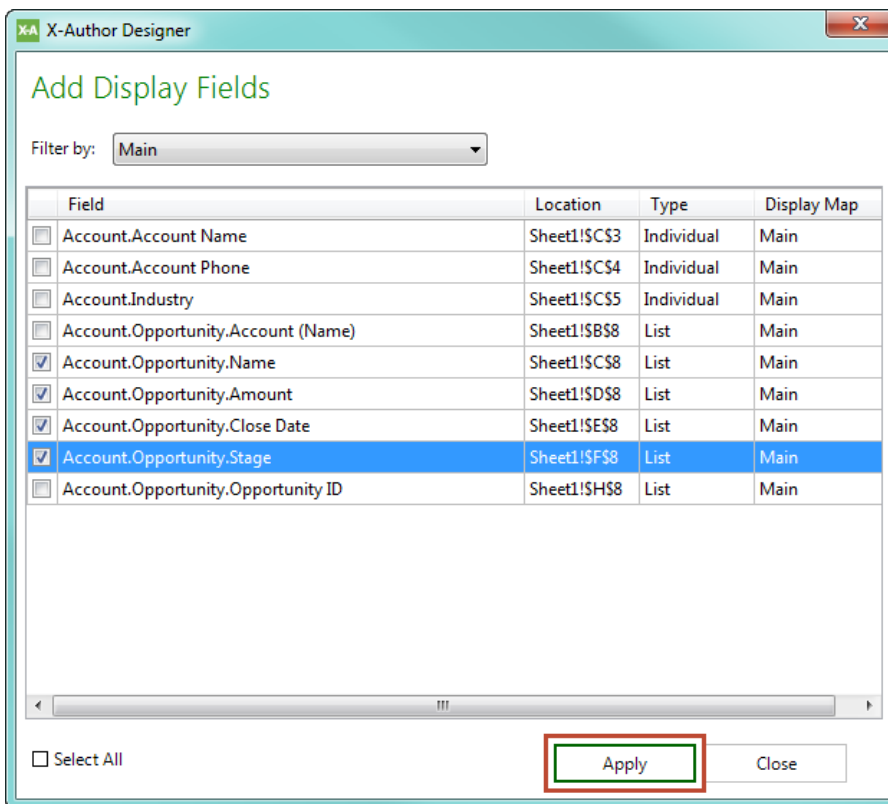
The image shows the 'Save Map' dialog box. At the top, the title 'Save Map' is displayed. Below the title is a 'Name' field with the text 'Map'. There are three buttons: 'Add Display Map Fields' (highlighted with a red border), 'Add Matrix Map Sections', and 'Add Other Fields'. Below these buttons is a link labeled 'Show Preloaded Grid'. Underneath is a section titled 'Mapped Fields' containing a table with the following columns: 'Name', 'Location', 'Type', and 'Field Type'. The table is currently empty. At the bottom of the dialog are three buttons: 'Remove Fields', 'Save' (highlighted with a green border), and 'Close'.

Click on the ▼ arrow icon in the upper right-hand corner and choose an option to move or resize the Save Map panel.

5. Use the **Filter by** drop-down to filter the list of fields for a specific Display Map.



6. Select the fields to add to your Save Map, and click **Apply**. Click **Select All** if you want to choose all Display Map fields.



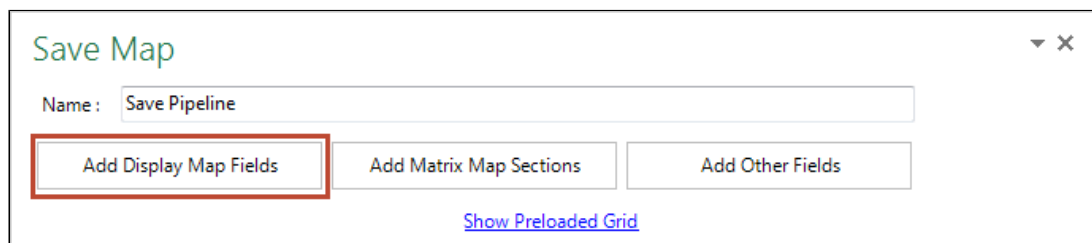
7. Click **Save**.
8. Click **Close**.
9. Click **Save App**.

## **① Use Case: Sales Pipeline Save Map**

Danny Designer is designing an app with Account, Opportunity and User objects for sales persons to use to manage their sales pipeline. He has created a Display Map to link data from Salesforce to the cells in the Excel workbook and now needs to create a Save Map for saving data back to Salesforce.

Danny takes the following steps to create a Save Map:

1. Danny connects to Salesforce and opens the Sales Pipeline app.
2. He clicks **Save Map** and clicks **Create** to start a new Display Map.
3. Danny enters the name "Save Pipeline" and clicks **Add Display Map Fields**.



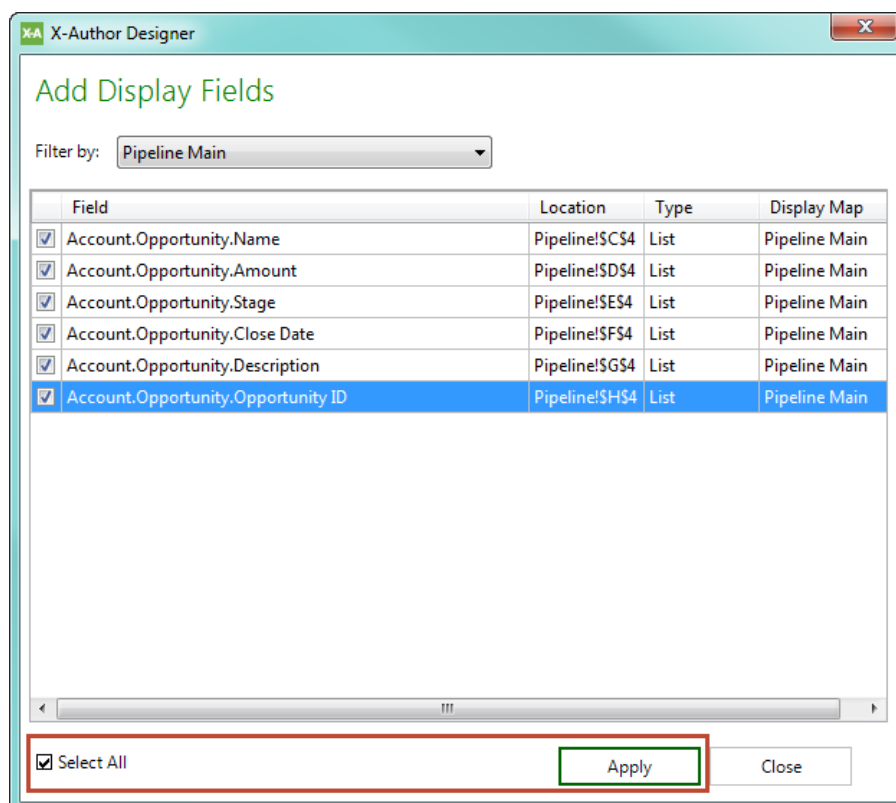
Save Map

Name : Save Pipeline

Add Display Map Fields Add Matrix Map Sections Add Other Fields

[Show Preloaded Grid](#)

- Danny decides that all of the Opportunity fields should be enable for saving back to Salesforce. He clicks **Select All** and clicks **Apply**.



- Danny clicks **Save** to save the map and clicks **Close**.
- Danny clicks **Save App** to save all changes to the app.

## To create a Display Map for Other Fields

- From the X-Author ribbon, click **Save Map**.
- Click **Create** (or select an existing Save Map and click **Edit**).
- In the Name field, enter a name for your Save Map. Pick a name that makes it easy to distinguish from your other Save Maps.

4. Click **Add Other Fields**.

**Save Map**

Name :

[Show Preloaded Grid](#)

Mapped Fields

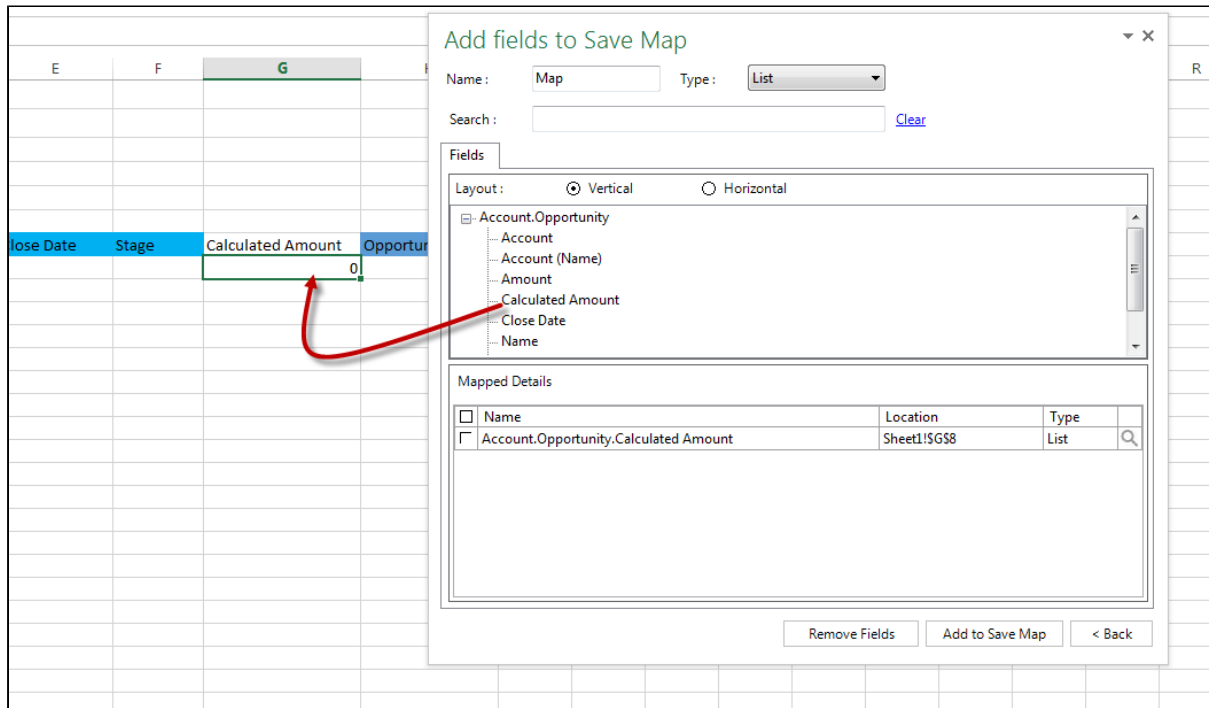
	Name	Location	Type	Field Type
<input checked="" type="checkbox"/>	Account.Opportunity.Amo...	Sheet1!\$D\$8	List	Display
<input type="checkbox"/>	Account.Opportunity.Clos...	Sheet1!\$E\$8	List	Display
<input type="checkbox"/>	Account.Opportunity.Name	Sheet1!\$C\$8	List	Display
<input type="checkbox"/>	Account.Opportunity.Stage	Sheet1!\$F\$8	List	Display

5. Select **Individual** or **List** from the Type drop-down.

6. Select the **Vertical** layout.



- Use the Search dialog or navigate the object hierarchy to locate fields. Add fields to the Save Map by dragging them from the panel and dropping them onto the Excel worksheet. You can search for specific fields using the Search box.



- After adding one or more fields, click **Add to Save Map**. The Save Map now displays the added fields in the Save Map panel and "Save Only" as the Field Type.

## Save Map

Name :

Add Display Map Fields

Add Matrix Map Sections

Add Other Fields

[Show Preloaded Grid](#)

Mapped Fields

	Name	Location	Type	Field Type	
<input type="checkbox"/>	Account.Opportunity.Amount	Sheet1!\$D\$8	List	Display	
<input checked="" type="checkbox"/>	Account.Opportunity.Calcul...	Sheet1!\$G\$8	List	SaveOnly	<input type="checkbox"/>
<input type="checkbox"/>	Account.Opportunity.Close ...	Sheet1!\$E\$8	List	Display	
<input type="checkbox"/>	Account.Opportunity.Name	Sheet1!\$C\$8	List	Display	
<input type="checkbox"/>	Account.Opportunity.Stage	Sheet1!\$F\$8	List	Display	

Remove Fields

Save

Close

9. Click **Save** and **Close**.

10. Click **Save App**.

**i** For information on adding Matrix Map sections to Save Maps, refer to [Creating a Matrix](#) under Advanced App Design.

## To delete an existing Save Map

**i** 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**.

## Creating Preloaded Grids

When your save requirements require something more complex than a list, you can use a Preloaded Grid with your Save Map. A Preloaded Grid can only be used to add new records. 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**.

## Customizing the App Experience

Actions are used to create a specific user experience. They include being able to query, display, save, and manipulate data within the workbook. The top section of the Actions menu 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
	Adds a row to the top or bottom of your Display Map.

Action Name	Action Type
Add Row Action	
Check-In	Saves the Excel file to Notes and Attachments in Salesforce and unlocks the file.
Check-Out	Opens a previously saved X-Author for Excel file from Salesforce Notes and Attachments and then locks the file to prevent editing by any other user.
Clear	Clears cell data.
Dataset Action	Combines outputs of Query Actions to overcome SOQL limitations.
Display	Places retrieved data in the worksheet based on a Display Map.
Macro	Runs an existing macro.
Paste Action	Counts the rows from which data has been copied and then pastes the copied data into those rows.
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 as an xls or pdf.

Action Name	Action Type
<a href="#">Search and Select</a>	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.
<a href="#">Switch Connection</a>	Switches between orgs to allow uploading and downloading data across multiple orgs.

## Display Action

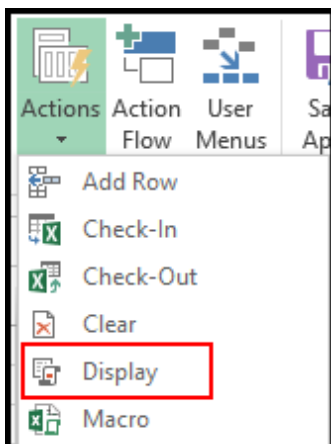
When data is retrieved from Salesforce using a **Query** or **Search & Select** action, it remains in memory until a Display Action places that data into the workbook based on a **Display Map**.

### Note

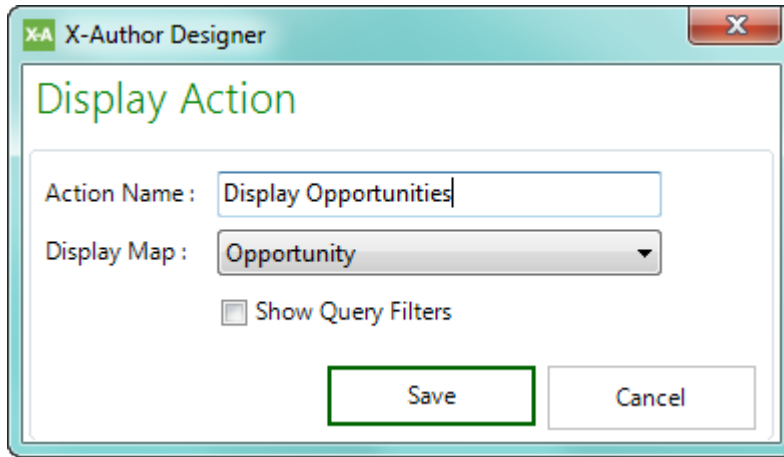
To create a Display Action, you must have an existing [Display Map](#).

## To create a Display Action

1. Click the lower section of the Actions button and select **Display**.



2. In the Action Name field, enter a name for your Display Action. Use a name that describes the records to display.

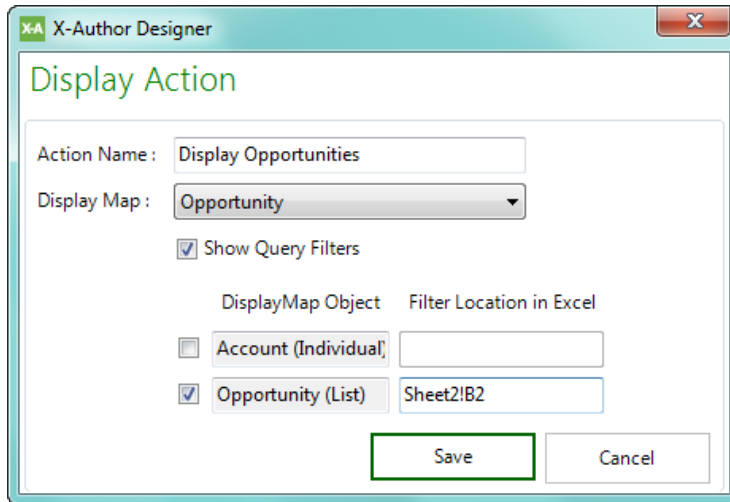


3. Click the drop-down next to Display Map to select the Display Map to use.
4. Click **Save**.

### To show query filters in the worksheet

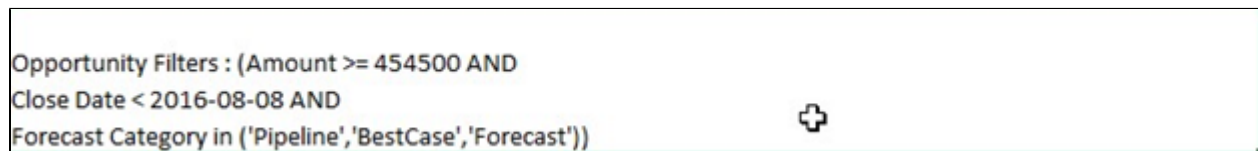
Perform the following optional steps when creating a Display Map if you want to display the filters used in any Query or Search & Select actions used to display records for the specified Display Map.

1. Select the **Show Query Filters** check box.
2. Select one or more **Display Map Objects** for which you want to display the query filters for during runtime.
3. For each Display Map object selected, enter a worksheet cell location in the **Filter Location in Excel** field. At runtime, all configured filters for queries used in that Display Map are populated in the specified cell location. For example, specifying the filter location **Sheet2!B2** displays query filters for the corresponding Display Map object in the cell **B2** on **Sheet 2**. You can also specify a named range as a filter location.



4. Click **Save**.

**Runtime Behavior:** When the App is executed during runtime, the query filter is displayed in Sheet 2 in cell B2. The query filters can also be displayed in merged cells.



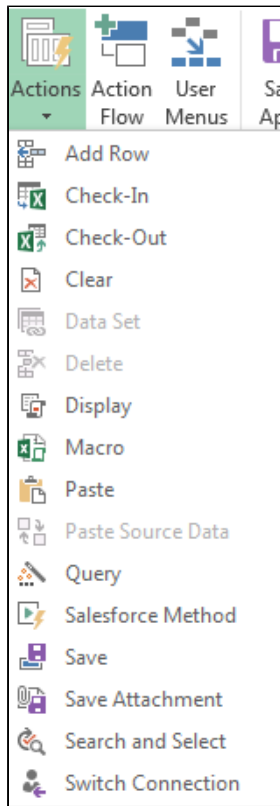
## Query Action


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

You can add many filters to Query Actions that are either hard-coded or that rely on either user or spreadsheet input at runtime. These filters can also be connected using filter logic. Query actions work with Display Actions **to retrieve and display records** in apps.

## To create a Query action

1. Click on the arrow beneath the Actions button and select **Query**.



2. Enter an **Action Name**. Use a descriptive name, for example, *Query Opportunities*.
3. From the **Object** drop-down list, select an Object.
4. Type a number in **Max Records** to specify the maximum number of records to be returned. Leaving it blank will return all records unless otherwise restricted by the filters below.
5. In the default filter row, 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, the object and field must have been selected in Salesforce Objects.
6. Select one of the following **Value Types**:



- **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". If you set the Value Type as Static, and the operator as in or not in and the query has more than 20,000 characters, the query is executed successfully during run-time. If the query condition comprises more than 20,000 characters, the query is chunked into multiple queries executed separately and the results are combined into a single query execution.
- **System Variables:** There are three system variables available for use.
  - Current User - Used to restrict records based on the currently logged in user.
  - Current Date - Used to restrict records to the current date.
  - ExportRecordId - Used for Apps that are [launched from Salesforce](#).
- **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.

The screenshot displays the 'Query Action' configuration window. The 'Filter' section is highlighted with a red border. It contains a table with filter rules. The first rule is for 'Account Name' using the 'User Input' type and the 'equals' operator. A pop-up menu is shown over the 'Account Name' field, allowing the user to assign a label for the filter (applies to User Input). The pop-up shows 'Account Name' selected, with a green checkmark and a red cross mark. The main interface also shows 'Action Name: Query Action', 'Object: Account (Individual)', and 'Max records:'.

Filter	Type	Operator	Value
1 Account Name	User Input	equals	

If you set the Value Type as User Input, the operator *as in* or *not in* and the user input query has more than 20,000 characters, the query is executed successfully during run-time. If the query condition comprises more than 20,000 characters, the query is chunked into multiple queries executed separately and the results are assimilated as a single query execution.

- **Cell Reference:** This will filter the query results based on the value of a cell reference provided. If you set the Value Type as Cell Reference, and the operator *as in* or *not in* and the query has more than 20,000 characters, the query is executed successfully during run-time. If the query condition comprises more than 20,000 characters, the query is chunked into multiple queries executed separately and the results are combined into a single query execution.



The Cell Reference **Value Type** is not supported in Presto Apps.

7. Select an Operator. The following table describes combinations of field types, value types, and operators (click to enlarge).

	Operator	Equals	Not Equal To	Starts With	Contains	In	Greater Than	Greater or Equal	Less Than	Less or Equal
<b>Datatype</b>										
<b>Value Type</b>										
<b>String</b>										
Static		•	•	•	•	•	•			
User Input					•	•				
Cell Reference					•					
<b>Picklist</b>										
Input										
Static		•	•	•	•	•				
User Input		•				•				
Cell Reference		•				•				
<b>Lookup</b>										
Input		•	•			•				
Static		•	•			•				
System Variable		•	•			•				
User Input		•								
Cell Reference		•								
<b>Date / DateTime</b>										
Static		•	•			•	•	•	•	•
System Variable		•	•			•	•	•	•	•
User Input		•	•			•	•	•	•	•
Cell Reference		•	•			•	•	•	•	•
<b>Decimal</b>										
Static		•	•			•	•	•	•	•
System Variable										
User Input		•	•			•	•	•	•	•
Cell Reference		•	•			•	•	•	•	•
<b>Double</b>										
Static		•	•			•	•	•	•	•
User Input		•	•			•	•	•	•	•
Cell Reference		•	•			•	•	•	•	•
<b>Boolean</b>										
Static		•	•							
User Input		•								
Cell Reference		•								



Formula field types are also supported. The corresponding supported operator types depend on the the datatype the formula field returns. The supported return types for formula fields are boolean, double, string, currency, percent, date and email.

8. Click **Add Row** to create more filter conditions.
9. Click **Save**.

A Query action is created.

Query Action

Action Name :

Object :  Max records :

Filters

	Salesforce Object and Field		Value Type	Operator	Value	
1	<input type="text" value="Account Name"/>		<input type="text" value="User Input"/>	<input type="text" value="equals"/>	<input type="text"/>	

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

## Filter Logic

X-Author uses the same filter logic method as Salesforce.

**Example:** For the defined filters, find records that match both the first two filters or the third. Enter filter logic as (1 AND 2) OR 3, or 3 OR (1 AND 2).

Find records where the amount is greater than “\$10 million” AND the close date is earlier than “03 /19/2016” OR any opportunity where stage equals “Qualification”.

Amount > 10000000, Close Data < 03/19/2016, Stage = 'Qualification'

**Query Action**

Action Name :

Object :  Max records :

Filters :

	Salesforce Object and Field	Value Type	Operator	Value	
1	Amount	Static	greater than	10000000	X
2	Close Date	Static	less than	03/19/2016	X
3	Stage	Static	equals	Qualification	X

[Add Row](#) [Clear All](#) [Clear Filter Logic](#)

Filter Logic :

See also: [https://help.salesforce.com/htviewhelpdoc?id=working\\_with\\_advanced\\_filter\\_conditions\\_in\\_reports\\_and\\_list\\_views.htm&siteLang=en\\_US](https://help.salesforce.com/htviewhelpdoc?id=working_with_advanced_filter_conditions_in_reports_and_list_views.htm&siteLang=en_US).

## Using Filter Logic at Runtime

When a query defined as User Input includes a picklist field as one of the criteria, you can see a preview of the values in the list.

Stage :

To specify a value for the query, click the ellipsis button next to a field to see additional picklist values.

When you click the ellipsis button a pop-up appears.

Select Opportunity.Stage Values

Go Clear

Available Values

- Prospecting
- Qualification
- Needs Analysis
- Value Proposition
- Id. Decision Makers
- Perception Analysis
- Proposal/Price Quote
- Negotiation/Review
- Closed Won
- Closed Lost

>>

<<

Selected Values

Save Cancel

- Add one or many values to the Selected Values area using the right arrow.
- Quick Search is a wildcard search which automatically filters down the list of available values in the list below.
- The values moved to the right under Selected Values area are selected for In or Not in" operator processing.
- After the user selects a list of values and clicks on Save, the window closes and the text on the Runtime screen shows the values separated by a comma in the textbox.

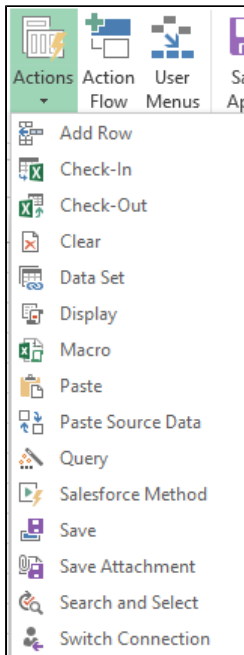
## Search and Select Action

The Search & Select action provides an interactive way for the end user to search for and select one or more records to retrieve from Salesforce. Along with the [Query Action](#), Search & Select is the primary means for retrieving records from Salesforce. Search & Select actions work with [Display Actions](#) to **retrieve and display records** in apps. A Search & Select action executes two calls to Salesforce when invoked:

- the first call assists the user in searching and selecting the data they need
- the second call retrieves all fields for the records selected by the user

## To create a Search and Select action

1. Click on the arrow beneath 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, for example, *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 **Single** for an Individual object, or **Multiple** for a List object.)
5. In the Fields to Display in Search Results, select one more fields to display in the search results. Select a check box in the **Sort By** column to choose the primary sort. Note: At 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. Refer to the [Query Action](#) for details on how to create filters.

### Search And Select Action

Options
Filters

Search And Select Action

Action Name : Search and Select for Account

Object : Account (Individual)

Searchable Fields

Searchable	Field Name	Field Label	Data Type
<input checked="" type="checkbox"/>	Name	Account Name	String

Result Selection :

☒ Single
☐ Multiple

Fields to Display in Search Results

Display	Field Name	Header Name	Sort By
<input checked="" type="checkbox"/>	Name	Account Name	<input type="checkbox"/>

↑

↓

Save

Cancel



**Search And Select Action**

Options Filters

Filters

	Salesforce Object and Field		Value Type	Operator	Value	
1	Account Name		User Input	contains		X

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

Save Cancel

7. Click **Save**.

## Save Action

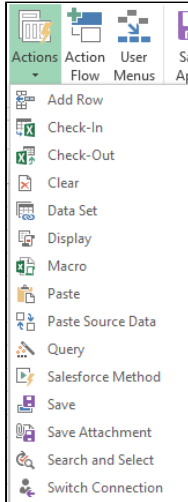
A Save Action pushes Excel workbook data into Salesforce based on a Save Map.



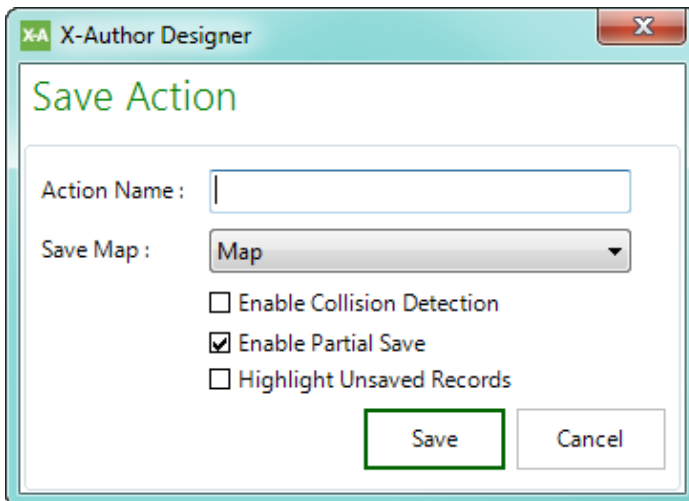
To create a Save Action, you must have an existing [Save Map](#).

## To create a Save Action

1. Click the lower section of the Actions button and select **Save**.



2. In Action Name, type a name. Pick a descriptive name to identify each Save action you create.



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

**Note**

Consider the following optional settings:

- Select the **Enable Collision Detection** check box to 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.
- Select the **Enable Partial Save** check box to allow valid rows to be saved to Salesforce even if other rows have errors or other issues (for example, rows that cannot be saved due to collision detection). Deselecting this box will prevent all rows from being saved even if there is a single error.
- Select the **Highlight Unsaved Records** check box to highlight the cells or rows that have an error during a Save Action.

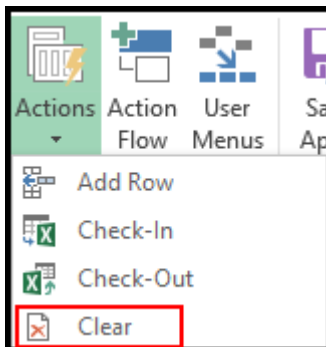
4. Click **Save**.

## 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. Another example would be when you want to combine process creation and editing in the same app, like quoting.

## To create a Clear action

1. Click on the arrow beneath the Actions button and select **Clear** from the drop-down. 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, select an existing map.
4. Click the check box 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.

### Clear Fields Action

Action Name :

Display / Save Map :

☒ Account  
☒ Agreement  
☒ Opportunity

Selected Objects

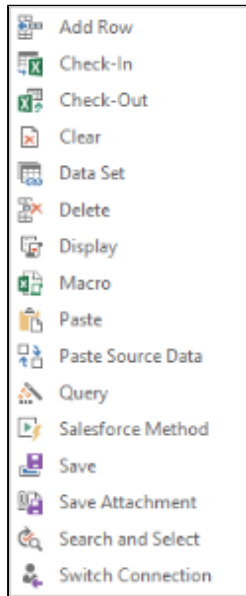
Map Name	Object	Object Type
Main Map-Save - (Save)	Agreement	Repeating
Main Map-Save - (Save)	Account	Independent
Main Map-Save - (Save)	Opportunity	Repeating

## Delete Action

Using this Action, you can delete records for an object based on specific filter criteria. You can delete records for only those objects referenced in the Save Map having Display Map fields along with Save other fields. If you do not specify any filter criteria, all the records for the object are deleted. You can specify a maximum of 20 filters. The Delete action is supported only on Enterprise and Power Admin Editions.

To create a Delete Row action

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



### Delete Action

Action Name :

Save Map :

Object :

**Filters**

	Salesforce Object and Field	Value Type	Operator	Value	
1	<input type="text" value="Configuration Name"/>	<input type="text" value="Static"/>	<input type="text" value="equals"/>	<input type="text"/>	<input type="text" value="X"/>
2	<input type="text" value="Record ID"/>	<input type="text" value="Static"/>	<input type="text" value="equals"/>	<input type="text"/>	<input type="text" value="X"/>

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

**Warning :** Delete Row Action will delete all records retrieved for Product Configuration if there are no filter defined

☒ Prompt Confirmation Message Before Deleting record(s)

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 deleting rows.
Object	Select the object that you want to delete the records for.

3. Specify the filters based on which you want to delete the records. For example, you can choose to delete records if the Approval Status of the record is Rejected. Operators are supported based on the datatype of the field, such as equals, not equal to, greater than,



greater than equal to, less than, and less than equal to. IN and Not In operators are not supported in this release. Only Static value type is supported in the May 2016 release and later. You can specify a maximum of 20 filters.

Filters are displayed based on the following conditions:

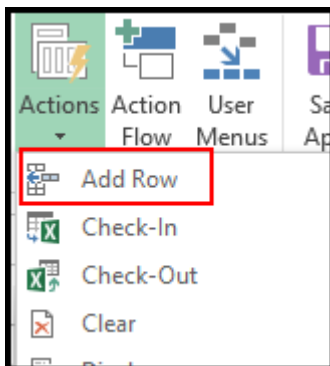
- Save map containing only display map fields will show all filter fields.
  - Save map containing display and save other fields both will show all fields.
  - Save map containing only save other fields shouldn't show anything.
  - Empty columns that are a part of the display map also appear in the filter fields.
4. To ensure that the user knows that a delete action is executed at runtime, the Prompt Configuration Message Before Deleting records check box is selected by default. If the user clicks Yes, the records are deleted and the next action in the action flow is executed. If the user clicks No, the next action in the action flow is executed.
  5. Click Save.

## 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.

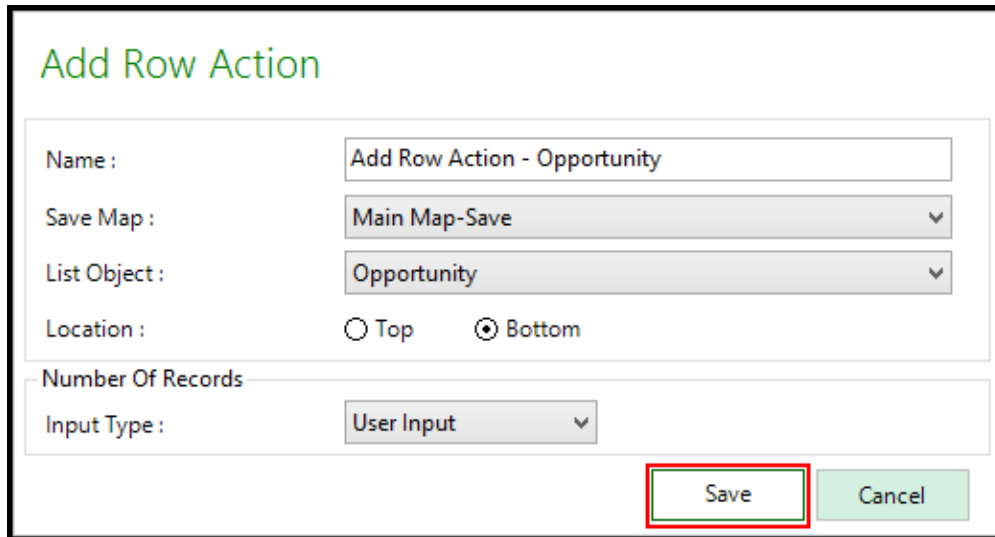


2. Complete the fields described in the following table.

Field	Description
Name	Type a descriptive name for action (for example, <i>Add Row – Opportunities</i> ).
Save Map	Select an existing Save Map to use when adding rows.

List Object	Select the list object for which you will add rows.
Location	Specify whether to add the new rows before or after the existing data.
Input Type	<p>Select the method used to determine how many rows will be added.</p> <ul style="list-style-type: none"> <li>• <b>Action Flow Step Input</b> – 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.</li> <li>• <b>Cell Reference</b> – Select this option to specify a cell in the workbook, such as <i>Sheet3!A1</i>, that contains the numerical value of the number of cells you want to add.</li> <li>• <b>Number of Rows</b> – Select this option to specify a static number of rows to add.</li> <li>• <b>User Input</b> – Select this option to allow the runtime user to specify the number of rows to add.</li> </ul>

3. Click Save.



**Add Row Action**

Name : Add Row Action - Opportunity

Save Map : Main Map-Save

List Object : Opportunity

Location : ☐ Top ☒ Bottom

Number Of Records

Input Type : User Input

Save Cancel

## Check-In Action

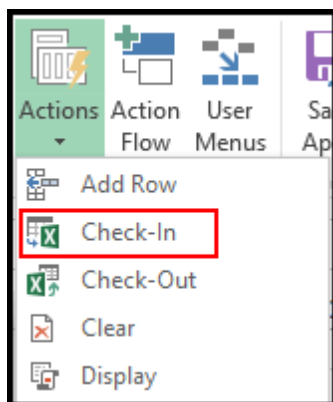
The Check-In action saves the Excel workbook to the Notes and Attachments of a designated record and then unlocks the workbook and enables it to be checked out by another user.

### Prerequisite

Because the workbook must be attached to a single record, your app must identify at least one Individual object in Salesforce Objects.

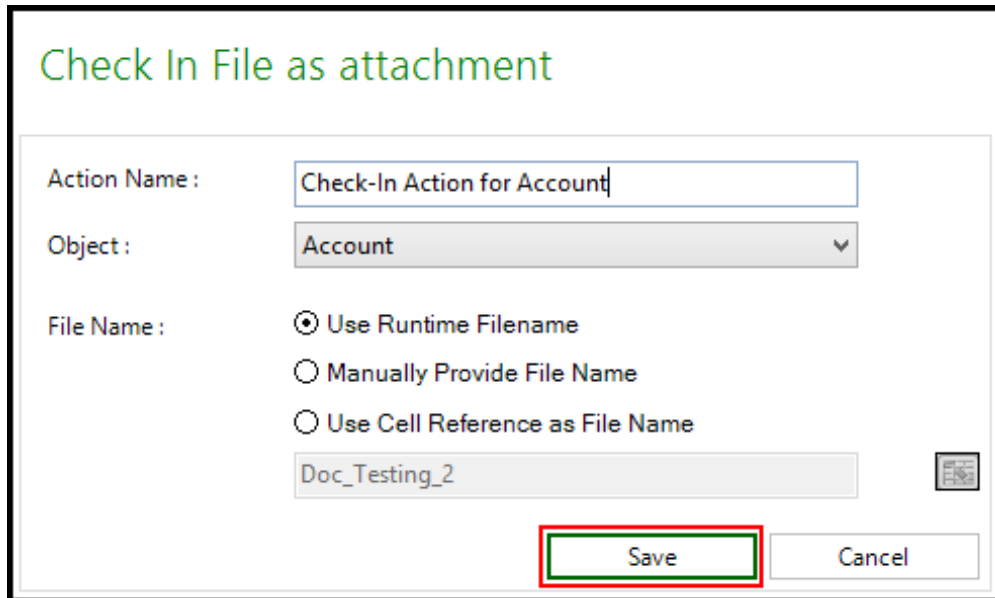
## To create a Check-In action

1. Click on the arrow beneath the Actions button and select **Check-In**.



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 only include Individual Salesforce Objects.
4. In the File Name, select the option to type the file name that will be used when the file is checked-in.

Field Name	Description
Use Runtime Filename	Select this option to allow your user to enter the File Name at run time.
Manually Provide File Name	Select this option to allow your user to enter the File Name manually. Enter the file name manually.
Use Cell Reference as File Name	Select this option to allow your user to use a cell reference as the File Name. Select the cell and click the icon. <b>Note: This feature is not available with Presto apps.</b>



**Check In File as attachment**

Action Name :

Object :

File Name : ☒ Use Runtime Filename  
☐ Manually Provide File Name  
☐ Use Cell Reference as File Name

## Check-Out Action

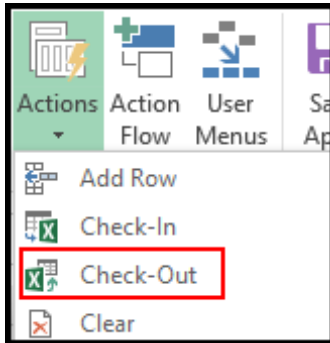
The Check-Out action provides the runtime user with a list of all files previously checked into the designated object or record and they can select one of those files to open. This then locks the file for other users until it is checked back in by the current user.



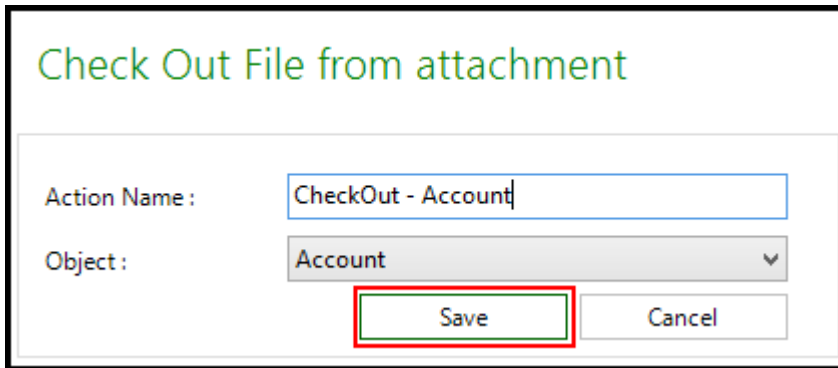
Because the workbook is checked out from a single record, you must have at least one object identified as *Individual* in Salesforce Objects. You must also have at least one [Check-In](#) action.

## To create a Check-Out action

1. Click on the arrow beneath 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.

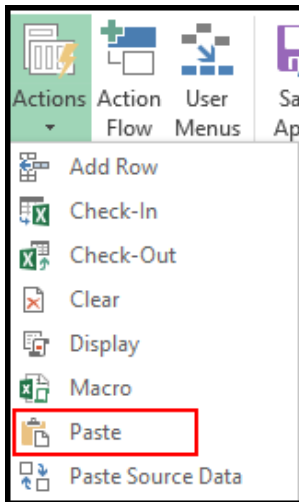


## Paste Action

The Paste action adds rows and pastes data from the clipboard into those rows.

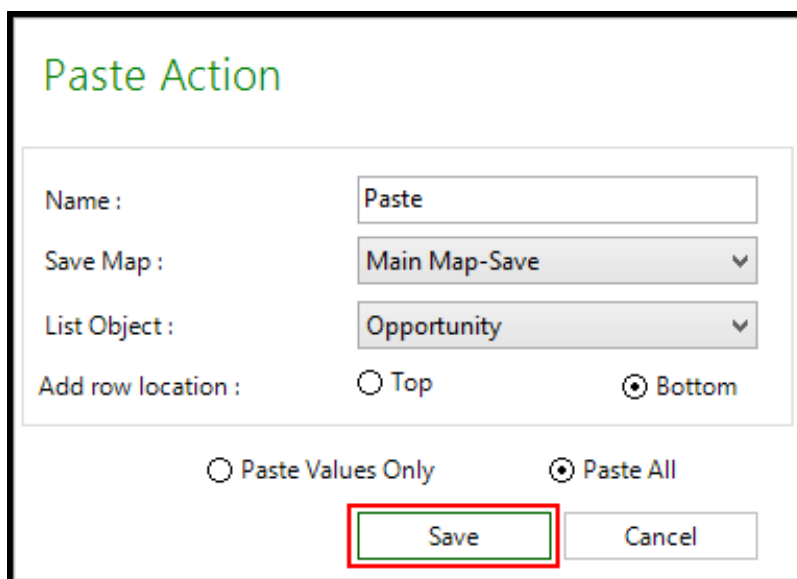
## To create a Paste action

1. Click on the arrow beneath the Actions button and select **Paste**.



2. In the Name field, enter a descriptive name for Paste action.
3. From Save Map picklist, select the Save Map for your Paste action.
4. From the List Object picklist, select the appropriate object (of List type).
5. Do any one of the following:
  - a. To add the rows at the top of the current data selection, select **Top** from Add row location.
  - b. To add the rows at bottom, select **Bottom** from Add row location.
6. Do any one of the following
  - a. If you want to paste all the rows and data, select **Paste All**.
  - b. If you want to paste the values only, select **Paste Values Only**.
7. Click **Save**.





**Paste Action**

Name :

Save Map :

List Object :

Add row location : ☐ Top ☒ Bottom

☐ Paste Values Only ☒ Paste All

## Paste Data Source Action

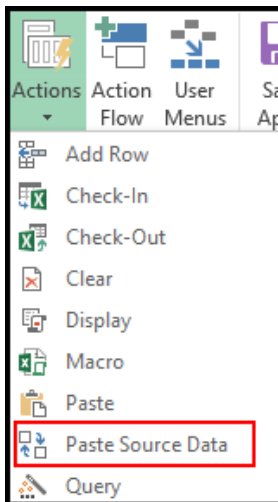
The Paste Data Source action is the automated equivalent of the runtime **Paste with Mapping** feature and is primarily used for migrating data between Salesforce orgs. The manual process maps data between source and target orgs using the Source Data function and then uses Paste from Mapping to copy the data downloaded from the Source org into the Target workbook using the Source Data layout. This action automates that process so that the migration can run uninterrupted.

### Prerequisite

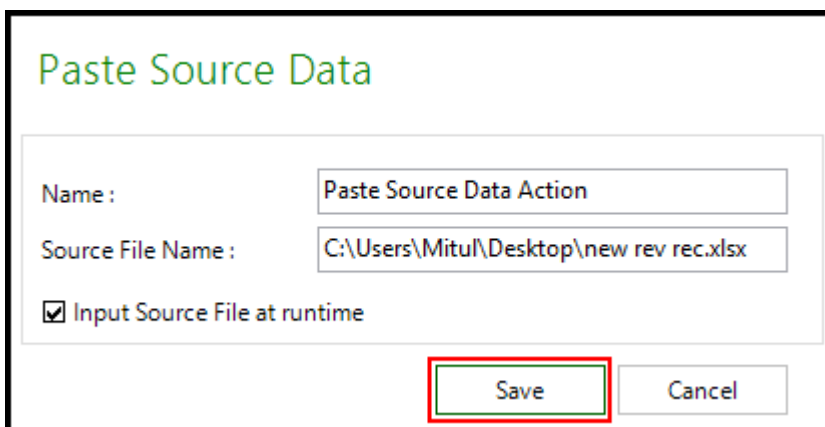
The Paste Data Source action requires a licensed edition that includes data migration.

## To create a Paste Data Source action

1. Click on the arrow beneath 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 called from the Action Flow, select the **Input Source File at Runtime** check box.
4. Click **Save**.



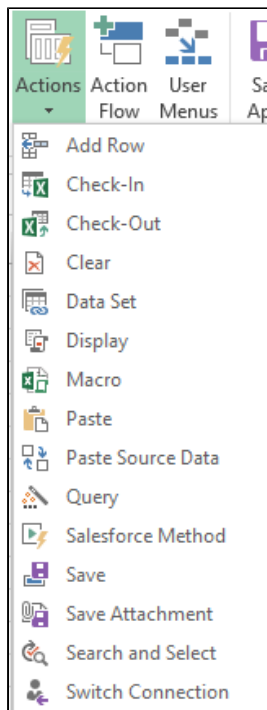
## Switch Connection Action

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, 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 runtime users are prompted at the beginning of the action flow to add the connection information.

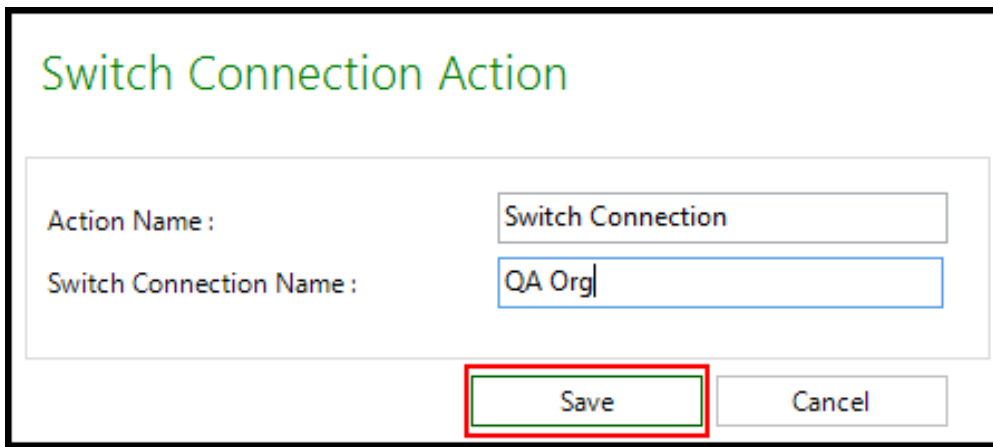
### To create a Switch Connection action

1. Click on the arrow beneath the Actions button 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 runtime 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 runtime user will be prompted for connection information when the app starts. If multiple connection changes are required, the runtime user will be prompted for all connection information at one time.

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

## Macro Action

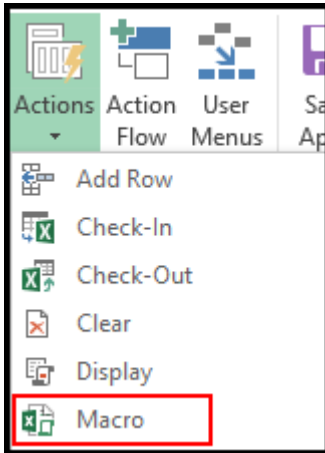
The Macro action executes a macro from the workbook. It can be executed at any time during an Action Flow and it can also be used to conditionally stop the next step in an Action Flow from executing.



To create a macro action, you must have an existing Macro in the Excel workbook.

## To create a Macro action

1. Click on the arrow beneath the Actions button and select **Macro**.

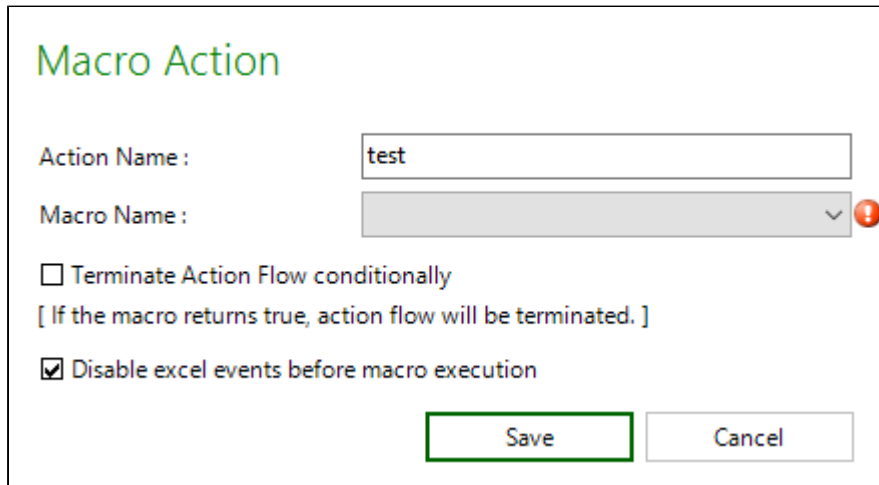


2. In Action Name, type a value that describes which Macro will be executed.
3. From Macro Name, select an existing macro.
4. Select the **Terminate Action Flow conditionally** check box if you want to terminate the action flow conditionally. The table below describes the behavior of the check box.

Macro	Terminate Action Flow conditionally check box	Result
True	Selected	The action flow will be terminated.
False	Selected	The action flow will continue.

5. Select the **Disable excel events before macro execution** checkbox.

Depending upon the value of the checkbox, enable or disable the excel events in macro action.



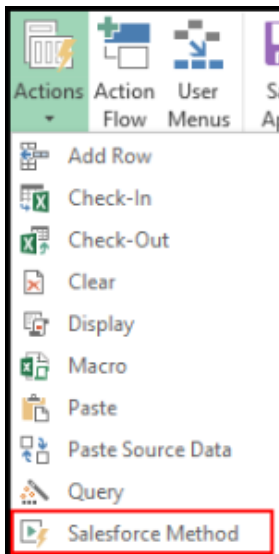
The image shows a 'Macro Action' dialog box. It has a title bar 'Macro Action' in green. Below the title, there are two input fields: 'Action Name' with the text 'test' and 'Macro Name' which is a dropdown menu. Below these fields, there are two checkboxes: 'Terminate Action Flow conditionally' (unchecked) and 'Disable excel events before macro execution' (checked). A note in brackets says '[ If the macro returns true, action flow will be terminated. ]'. At the bottom right, there are 'Save' and 'Cancel' buttons. The 'Save' button is highlighted with a green border.

## Salesforce Method Action

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

### To create a Salesforce Method action

1. Click on the button beneath the Actions button and select Salesforce Method.



2. Under Action Details, type an Action Name for the call procedure action.

3. Under Salesforce Method Details, select a **Class** and type the name of the **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 parameters. 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 displays an input field to the user at runtime and passes that input to the Salesforce Method.
7. Click **Add Param** to add more than one parameter.
8. Click **Save**.

### Salesforce Method Action

**Action Details**

Action Name :

**Salesforce Method Details**

Class :

Method :

☒ Has return value      Return type :

☒ Has parameters      ☒ Enable Cache

**Specify Parameters**

[Add Param](#)      [Clear All](#)

## Save Attachment Action

The 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 Check-Out action if file locking is required.

An attachment can be saved either as an Excel or PDF file format.



When an attachment is downloaded, the file name is the actual filename in Salesforce. **<filename in Salesforce>.<extension in Salesforce>**.

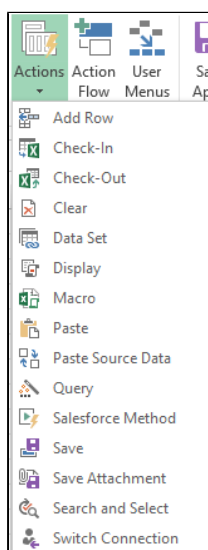
If a file with the same name and extension exists, a sequence number is appended after the filename, such as **<filename in Salesforce> (2).<extension in Salesforce>** and **<filename in Salesforce> (3).<extension in Salesforce>**.



To create a Save Attachment action, at least one object must be identified as *Individual* in Salesforce objects.

## To create a Save Attachment action

1. Click on the arrow beneath 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.

Field Name	Description
Use Runtime File Name	Select this option to use the existing File Name at runtime.

Field Name	Description
Manually Provide File Name	Select this option to allow your user to provide the File Name manually.
Use Cell Reference as File Name	Select this option to allow your user to use a cell reference as the File Name. Select the cell and click the icon. <b>Note: This feature is not available with Presto apps.</b>

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 specific worksheets. Hold the Ctrl key to select multiple worksheets listed in **Include Sheets**.
7. Click **Save**.

The screenshot shows the 'X-Author Designer' window with the 'Save Attachment Action' dialog box open. The dialog has a title bar with the X-Author logo and a close button. The main area is titled 'Save Attachment Action' in green. It contains the following fields and controls:

- Action Name :** An empty text input field.
- Object :** A dropdown menu currently showing 'Account'.
- Format:** Two radio buttons: 'Excel' (selected) and 'PDF'.
- File Name :** Three radio buttons: 'Use Runtime File Name', 'Manually Provide File Name' (selected), and 'Use Cell Reference as File Name'.
- File Name Input:** A text box containing 'CC Power Admin' and a small icon button to the right.
- Custom Sheet Selection:** A checked checkbox.
- Include Sheets :** A list box containing 'Sheet1', 'Sheet2', 'Sheet3', and 'Sheet4'. 'Sheet1' is highlighted in blue.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

## Dataset Action

Salesforce has various query limitations:

1. SOQL statements cannot have more than one level of sub-query.
2. Sub-queries cannot be executed when there is an 'OR' clause in the filter.
3. SOQL statements cannot exceed 20,000 characters in length.

If your query exceeds 20K characters, an SOQL error appears stating the query cannot exceed 20,000 characters in length. X-Author helps overcome these limitations using a combination of automated functionality and the Data Set Action (simulates a SQL union function).

X-Author checks the queries in runtime and automatically runs sub-queries and data chunking functionality to overcome a >20K character query. If during testing you find that even this is insufficient, then create a Data Set Action where you can break the query action into multiple query actions and use the query output as an input to the Data Set action.

Using the Data Set Action, you can merge the output of two queries into a single Data Set based on the OR clause or AND clause. When you specify 'OR' clause for query outputs, one output Data Set is created by union of the two query outputs. When you specify the AND clause for query outputs, one output Data Set is created by intersection of two query outputs.

## Prerequisites

1. Make sure that the application type is **Power Admin**. The Data Set Action can be only executed in applications that have the Power Admin edition.
2. For all queries that cross the limit of 20K, split the query into separate query actions. Ensure that your split query meets the following criteria:
  1.
    - If there are two 'IN' filters or 'NOT IN' filters and if the query crosses the limit of 20K, ensure that you:
      - Separate the 'IN' clause from the query and create a different query action.
      - Create two separate Query Actions each having a single 'IN' filter.
      - While creating an action flow and selecting a query action, ensure that 'Persist Data in Output' checkbox is selected.
      - Using the Data Set action procedure below, merge the data retrieved by both the query action in a single Data Set using a Data Set Action. You can choose to use the 'OR' clause or 'AND' clause to merge the data into a Data Set.
    - If there are two or more 'IN' filters or 'NOT IN' filters and if the query crosses the limit of 20K, ensure that you:
      - Separate the 'IN' clauses into a separate query action.
      - While creating an action flow and selecting a query action, ensure that 'Persist Data in Output' checkbox is selected.
      - Using the Data Set action procedure below, merge the output of those query action flows using a Data Set Action into a single data set.

## To create a Data Set action

1. Click on the arrow beneath the Actions button and select **Data Set**.
2. Enter an Action Name. Use a descriptive name. For example, Data Set Product Name.
3. From the Object drop-down list, select an Object. The object of the Data Set must be the same as those of the split Query Actions.

**Data Set Action**

Name :

Target Object :

Operation : ☒ OR Operation ☐ AND Operation

4. Specify the operator for the Data Set as AND or OR. The output of the different queries is merged into the Data Set, either as a union or an intersection.
5. Navigate to Action Flows, do any of the following:
  - a. To create a new Action flow, click **Create**, specify a name for the action flow. Click **Add a step**.
  - b. To add a Data Set to an existing Action Flow, select the existing Action Flow and click **Edit**. Click **Add a Step**.
6. Enter the **Step Name**.



Before you create a Data Set action step for an Action flow, make sure that steps for the Query Actions exist.

7. Select the Data Set Action. The Data Set Action that you specify must have subsequent Query Action flows created for it based on the object you select.

8. Specify the name of the Data Set **Action Output**.
9. Make sure to select the **Provide Input** and **Persist Data in Output** check boxes.
10. In the Action Input Area, specify the query actions from which the output is merged.

The screenshot displays the X-Author configuration window. On the left, a tree view lists various query actions. The action '30. Related Price List Item based on All Price List Items' is selected. The right pane shows the configuration for this step. The 'Step Name' is '30. Related Price List Item based on All Price List Items'. The 'Select Action' dropdown is set to '30. Related Price List Item based on All Price List Items'. The 'Persist Data as Output' and 'Provide Input' checkboxes are checked. The 'Action Output' name is 'Related Price List Items'. The 'Action Input' section shows 'Get External Input' unchecked, and two input fields are set to 'Related Price List Items - Price List' and 'Related Price List Items - Related Price List Items'. The 'Apply' button is visible below the input fields. At the bottom right, 'Save' and 'Close' buttons are present.

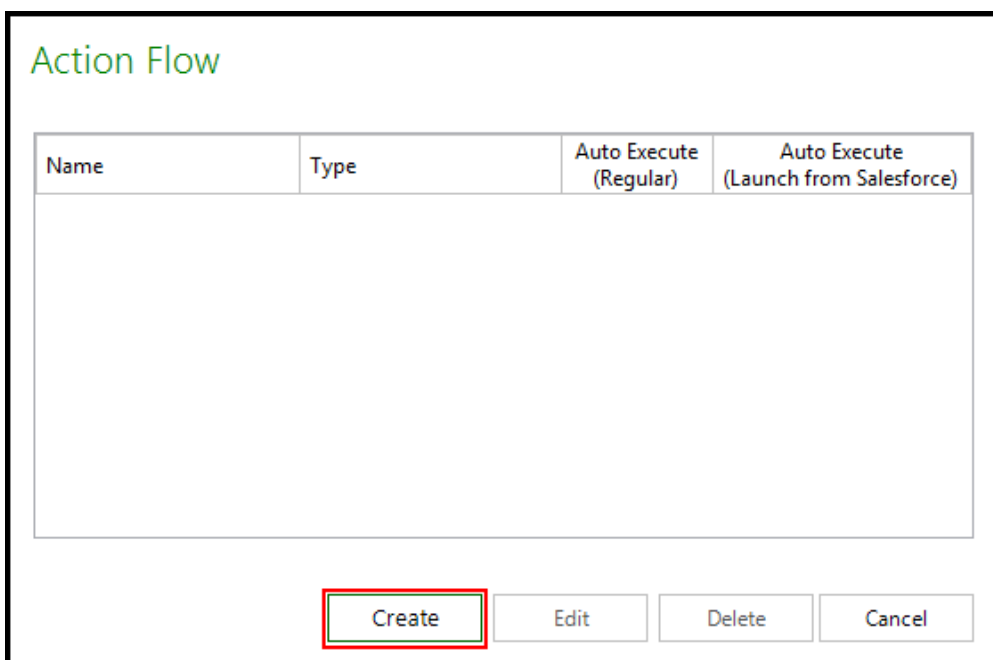
11. Click **Apply**.

## Executing Actions

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 provide outputs to subsequent actions.

### To create an Action Flow

1. Click the **Action Flow** button on the X-Author Designer ribbon.
2. Click **Create**.



Name	Type	Auto Execute (Regular)	Auto Execute (Launch from Salesforce)
------	------	------------------------	---------------------------------------

Create Edit Delete Cancel

3. Enter an Action Flow Name and then click **Apply**.

#### **Note**

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.

4. To add a step in the flow, click **Add Step**.
5. On the right pane, under Step, type a **Step Name** (use any name, such as *1* or *Search Account*) and then click **Apply**. 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 (like *Opps*). If this action requires an input, select **Provide Input** and select an input from the Action Input list. Based on the type of Action, X-Author Designer provides input suggestions. See the [Action Flow Check Boxes](#) table below for more information on when to use inputs and outputs.

6. Repeat steps 3 – 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.
7. 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.
8. After you have created the Action Flow, click **Save** and click **Save App**.



The screenshot shows the X-Author Designer interface. On the left, a tree view displays the action flow structure:

- Retrieve AC Opp Agreements
  - Search Account
    - Search Account (SearchAndSelect)
  - Search Opportunity
    - Search Opportunity (SearchAndSelect)
  - Agreements Query
    - Agreements Query (Query)
  - Display
    - Display AC Opp Agreements (Display)

At the bottom left of the tree view is a checkbox labeled ☐ Display Apttus CPQ Actions.

On the right, the 'Action Flow' section contains:

- Buttons: Add Step, Remove Step, Step Up, Step Down.
- Action Flow Name: Retrieve AC Opp Agreements (in a text box).
- Buttons: Apply, Save, Close.

The 'Save' button is highlighted with a red rectangle.

## Action Flow Check boxes

Field	Description
Persist Data as Output	Select this check box if an action generates any output. Type a <b>Name</b> for the output. You can use any name. For example, <i>Opps</i> .
Provide Input	Select this check box if an action requires input. Select an action from <b>Action Input</b> list. Based on the type of Action, X-Author Designer provides input suggestions.

Field	Description
Suppress Save Message	Select this check box if you have multiple save actions in a single action flow and you want to suppress the save message of that particular action. <b>Note:</b> This check box is enabled for the Save action only.

## Suppress Message Validations

When no records found in a Save Action check box (App Settings)	Suppress Save Message check box (Action Flow)	Behavior
Yes	No	Message is not displayed.
No	No	Message is displayed.
No	Yes	<b>No Records Found for Save</b> message is displayed.
Yes	Yes	Message is not displayed.

When all records are saved successfully in a Save Action (App Settings)	Suppress Save Message check box (Action Flow)	Behavior
No	No	Message for each Save action is displayed.
No	Yes	Message is not displayed.

When all records are saved successfully in a Save Action (App Settings)	Suppress Save Message check box (Action Flow)	Behavior
Yes	No	Message is not displayed.
Yes	Yes	Message is not displayed.

### Action Flow Required Inputs and Outputs

Action	Input	Output
<b>Check-in</b>	Always requires an input because the Excel workbook is saved to an individual record. Although the individual object is already identified when you create the action itself, it is possible you could have used that object in different ways in various action flows. Therefore, the Action Flow requires specific output from a prior action as input to this one.	Never
<b>Check-out</b>	Always requires an input because the Excel file is opened from an individual record. Although the individual object is already identified when you create the action itself, it is possible you could have used that object in different ways in various action flows. Therefore, the Action Flow needs to have the specific output from a prior action as input to this one.	Never
<b>Clear Fields</b>	Never. Clear Fields is tied to a specific Save Map.	Never
<b>Display</b>	Always requires an input. Any object on a Display Map must be represented by an output from a prior Action.	

Action	Input	Output
<b>Macro</b>	Never.	Never
<b>Query</b>	Only if INPUT has been used in the query action filter.	Always
<b>Dataset</b>	Always requires an input. Datasets take the outputs of query action as inputs. Output can be saved as a single dataset.	Always
<b>Salesforce Method</b>	Optional.	Optional
<b>Save</b>	Never.	Never
<b>Search and Select</b>	Input is required only if INPUT has been used in the Search and Select action filter.	Always

## 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.

You can auto-execute the action flow and [launch an action from Salesforce](#) as well.

## To launch an Action Flow when the app is opened in runtime

An Action Flow can be kicked off automatically either when the app is opened directly in Excel or from a Salesforce button.

## When the App is opened from Excel

1. Click **Action Flow**.

2. Select the **Auto Execute (Regular)** check box for the action flow that should execute automatically.
3. Click **Cancel**.

## When the App is opened from a Salesforce button

There are two choices:

- Follow the steps above and select the **Auto Execute (Launch from Salesforce)** check box
- Execute a different Action Flow for each Salesforce button.

For example, if you are creating a quote from an Opportunity, you may want to execute the Create Quote Action Flow from the Create Quote button in Salesforce. For the Edit Quote you may want a completely different user experience so you would create a second Action Flow and attach that to the Edit Quote button.

## Creating User Menus

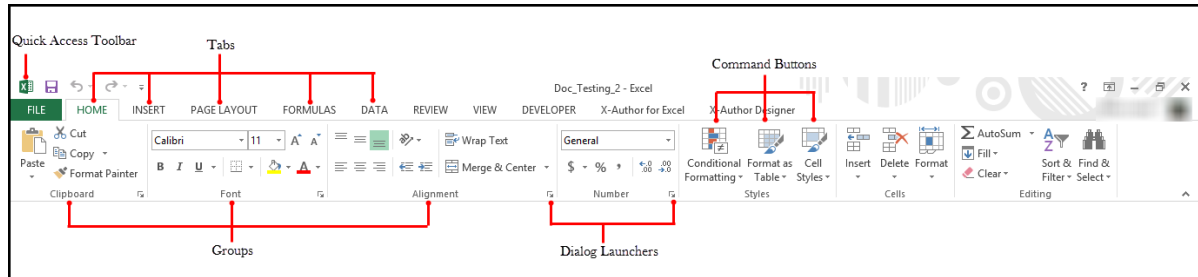
User Menus allow the Designer to create menus that the end user will see in their Excel ribbon when they open an X-Author App in runtime. There are two types of menu items:

- Configurable menu buttons that 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.

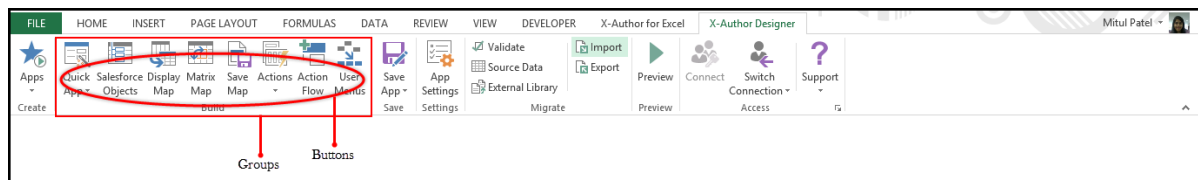
User Menus are divided into two areas:

- 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 the procedure below or any other icon from the list. You must enter the icon capitalization exactly as described in the procedure.

## 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.
  - 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 need 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 need 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**.

## Saving or Cloning an Application

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

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

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 (for example, Save Map or Display Map).

Click 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.

## Deleting an Application

You cannot delete an App using the X-Author for Excel add-in. To delete an app, you must log in to your Salesforce org using a web browser. Note: You must have sufficient administrator privileges to perform this task.

### To delete an Application

1. Log in to your Salesforce org using a standard web browser.
2. From the Force.com App Menu, select **Apttus X-Author for Excel**.
3. Click on the **Apps** tab.
4. Click **Go** next to the X-Author for Excel picklist. This action displays all of the X-Author Designer-created Apps.
5. Click **Del** next to the App name to delete the App, and then click **OK**.

## Example: Opportunities by Account App

### App Objectives

- The end user will search for and select an Account from Salesforce.
- Based on the account selected, the app will automatically retrieve all Opportunities into Excel.
- For the Account, the app will include the Name and Phone fields.



- For each Opportunity, the app will include the Name, Amount, Close Date, and Stage fields.
- Allow the end user to save modifications to the Amount and Stage fields for each Opportunity.

## App Components

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

Component	Component Description
Salesforce Objects	Account (Individual) and Opportunity (List)
Display Map	Individual and List fields per objectives
Save Map	Using fields from the Display Map
Actions	<ul style="list-style-type: none"> <li>• Search and Select Action for the Account</li> <li>• Query Action to retrieve the Opportunities for the Account</li> <li>• Display Action to transfer the retrieved data to the Display Map</li> <li>• Save Action to save back to Salesforce any user modifications</li> </ul>
Action Flows	One Action Flow to retrieve the data and display it and one Action Flow to save the data
User Menus	One button for the user to retrieve opportunities and one to save opportunities to Salesforce.

## To create the Opportunities by Account App

1. In the ribbon menu, click **Apps** and select **New Application**.
2. In the App Name Field, type *Opportunities by Account*.
3. In the Select Template section,click the **New** radio button.

4. Click **Create**.
5. For the app type, select the product edition. For example, *Enterprise*.

## To select Salesforce Objects

1. Click **Salesforce Objects** and locate Account > Opportunity and then click **Add Child Object**.
2. In the right-hand window, click **Account** (Ind).
3. Select fields from Add Fields below area: *Region*, *Name* and *Phone*.
4. Click **Apply**. Note: You must click Apply after you have selected the fields for any object.
5. In the right-hand window, click **Opportunity (List)**.
6. Select fields from Add Fields below area: *Account Id* (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. Click **Save App**.

## To create a Display Map

1. Select **Display Maps** and click **Create**.
2. In the Name field, type *Main* and from the Type list, select *Individual*.
3. Drag the *Account Name* field 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 the *Name* field 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. In the Name field, type *Main*.
3. Click **Add Display Map Fields** and select *Amount* and *Stage*.
4. Click **Apply** and then click **Save**.
5. Click **Save App**.

## To create a Search and Select Action

1. In the main ribbon under Actions, click the arrow and select **Search and Select**.
2. In the Action Name field, type Select **Account**.
3. From the Salesforce Objects list, choose **Account**.
4. In the Search window, click the check box next to Name.
5. In the **Name in Fields to Display in Search Results**, click the check box next to it and the Sort check box to the right.
6. From the Results area, choose **Single**.
7. Click **Save**.
8. Click **Save App**.

## To create a Query Action

1. Click the arrow under **Actions** on the X-Author for Excel ribbon and select **Query**.
2. In the Action Name field, type *Query Opportunities*.
3. From the Object list, choose *Account.Opportunity*.

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. (Account.AccountID)

5. Click **Save**.

## To create a Display Action

1. Click the arrow under **Actions** and select **Display**.
2. For the Action Name, type *Display Opportunities*.
3. From the Display Map list, choose **Main**.
4. Click **Save**.
5. Click **Save App**.



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.

## To create a Save Action

1. Click the arrow under **Actions** and select **Save**.
2. In the Action Name field, type *Save Opportunities*.
3. From the Save Map list, select **Main**.

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. In the Step Name field, type *Select Account*. From the Action list, select **Select Account**.
5. Select the check box next to the **Persist Data as Output** check box and enter **Account** in the Action Output field.
6. Click **Apply** and then click **Add Step**.
7. In the Step Name field, type *Query Opportunities*.
8. From the Action list, select **Query Opportunities**.
9. Select the check box next to **Persist Data as Output** and enter *Opps* in the Action Output field.
10. Select the check box next to **Provide Input** and then select **Account** from the picklist.
11. Click **Apply** and then click **Add Step**.
12. In the Step Name field enter *Display Opportunities* and select **Display Opportunities** from the Action list.
13. Select the check box next to **Provide Input** and select Account and Opps from the Action Input list.
14. Click **Apply** and then click **Save**.
15. Click **Save App**.

## To create a flow to save Opportunities

1. Click **Action Flow** and then click **Create**.
2. In the Action Name field type *Save Opportunities* and then click **Apply**.

3. Click **Add Step** and from the Action list choose **Save Opportunities**.
4. Into the Step Name field, enter *Save Opportunities* and click **Apply**.
5. Click **Save**.
6. Click **Save App**.

The **final step** is to **add the end-user menus**.

The workflow to add end-user menus is as follows:

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 in the procedure below or any other icon from the list. You must enter the icon capitalization exactly as it is described in the procedure below.

The steps to add end-user menus are described in detail in the following sections.

## To create User Menus

1. Click **User Menus**.
2. Click **Add Group**.
3. In the Name field, enter *Opportunities* and *1* in the Order field.
4. Click **Add Button** and in the Name field enter *Retrieve Opportunities*.
5. In the Icon field enter "GetExternalDataFromOtherSources".
6. Enter *1* in the Order field.
7. From the Action Flow list, select **Retrieve Opportunities**.

8. Click on **Opportunities** to highlight the group and then click **Add Button**.
9. In the Name field, enter *Save Opportunities*.
10. In the Icon field, enter *SaveObjectAs*.
11. In the Order field, enter 2.
12. From the Action Flow list, select **Save Opportunities**.
13. Click **Save**
14. Click **Save App**.

## Improving the appearance of end-user menus

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.  
Make sure to place your cursor in the cell and use the Delete key. Do not use row delete!

After you follow all the procedures in this section, a new App with the name Opportunities by Account has been created.

## Example: Opportunity Forecast Application

### App Objectives

1. The end-user invokes all opportunities, for which they are an owner, from Salesforce into Excel.
2. For each Opportunity include: Name, Amount, Close Date, Stage, and Owner */d*.
3. Allow the end-user to save modifications to only the Amount and Stage for each Opportunity.

## App Components

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

Component	Component Description
Adding Salesforce Objects to your App	<i>Opportunity</i> (List)
Display Map	<i>Opportunity</i> Fields as above
Save Map	Using fields from the Display Map
Customizing the App Experience	<ul style="list-style-type: none"> <li>• Execute Query Action with a restriction on "Owner ID"</li> <li>• Display Action to transfer the retrieved <i>Opportunity</i> data to the Display Map</li> <li>• Save Action to save back to Salesforce any user modifications</li> </ul>
Executing Actions	One to retrieve opportunities and one to save them
Creating 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.

### **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 Opportunity Forecast App

1. In the ribbon menu, click Apps and select **New App**.
2. In the App Name Field, type *Opportunity Forecast* and in the Select Template section, click the **New** radio button.
3. Click **Create**.
4. Select edition. For example, *Enterprise*.

## To select Salesforce Objects

1. In the ribbon menu, click **Salesforce Objects**.
2. In the Objects list, locate Opportunity, 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 **Save App**.

## To create a Display Map

1. Click **Display Map** and select **Create**.
2. In the Name field, type *Main* and from the Type drop-down, select **List**.
3. Drag the *Account Name* field to A7. This field was not available in Salesforce Objects, but X-Author for Excel automatically includes 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. To save the Display Map, click **Save** and then click **Close**.
7. Add a title in A1, for example, *Opportunities Forecast*.
8. Save the **App**.

## To create a Save Map


1. Click **Save Maps** and select **Create**.
2. In the Name field, enter *Main*.
3. Click **Add Display Map Fields** and select *Amount* and *Stage*.
4. Click **Apply** and then click **Close** for the Save Map.
5. Click **Save App**.

## 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**.
8. Click **Save App**.

## To create a Query Action

1. Under Actions, click the arrow and select **Query**.
2. In the Action Name field, type *Query Opportunities* and from the Object picklist select **Opportunity**.
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 Id* field. Because this is a lookup field, click 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**.
5. Click **Save App**.

## To create a Display Action

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.



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.

## 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. Click **Save App**.

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. In the Action Flow Name field, type Retrieve Opportunities and click **Apply**. The left-hand window is updated.
3. Click **Add Step**.
4. In the Step Name field, type *Query Opportunities*.
5. From the Action list, select *Query Opportunities*.
6. Check **Persist Data as Output**.
7. Enter *Opps*, then click **Apply**.
8. Click **Add Step**.
9. In the Step Name, enter *Display Opportunities*.
10. From the Action list, select *Display Opportunities*.
11. Check **Provide Input** and choose *Opps* as default.

12. Click **Apply** and then click **Save**.
13. Click **Save App**.

## To create an action flow to save Opportunities

1. Click **Action Flow** and then click **Create**.
2. In the Action Flow Name field, enter *Save Opportunities* and click **Apply**. The left hand window is updated.
3. Click **Add Step** and in the Step Name field, enter *Save*.
4. From the Action list, select **Save Opportunities**.
5. Click **Apply**.
6. Click **Save**.
7. Click **Save App**.

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

- 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 button 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 the icons described in the procedure below or use any other icon from the list. You must enter the icon name, including capitalization, exactly as it is documented in the procedure below.

The steps are described in detail in the following sections.

## To create User Menus

1. Click **User Menus**.

2. Click **Add Group**.
3. In the Name field, type *Opportunities* and *1* in the Order field.
4. Click **Add**, and in the Name field, type *Retrieve Opportunities*.
5. In the Icon field, type "GetExternalDataFromOtherSources" and in the Order field enter *1*.
6. Select **Retrieve Opportunities** from the Action Flow list.
7. Click on **Opportunities** to highlight the group and then click **Add Button**.
8. In the Name field, type *Save Opportunities*.
9. In the Icon field, type "SaveObjectAs" and in the Order field enter *2*.
10. From the Action Flow list, select **Save Opportunities**.
11. Click **Save**.
12. Click **Save App**.

## To make the end-user menus look presentable

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. (Optional) 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 will see a message that the records were saved. The changes are reflected in Salesforce.

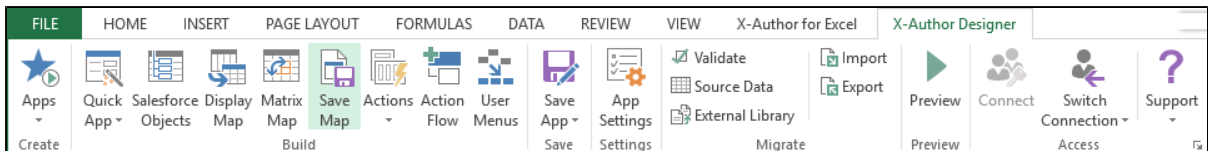


## Previewing Apps


You can use the **Preview** function in X-Author Designer to preview your app. Preview allows you to test the app as a runtime user.

### To preview an app

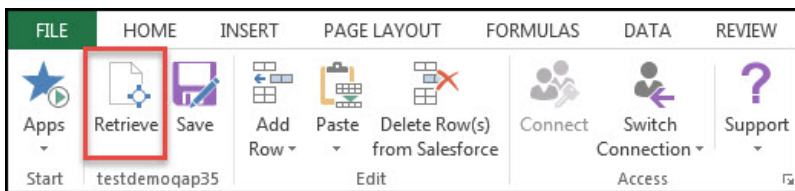
1. After creating an app, click the **Preview** button in the ribbon menu.



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.

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

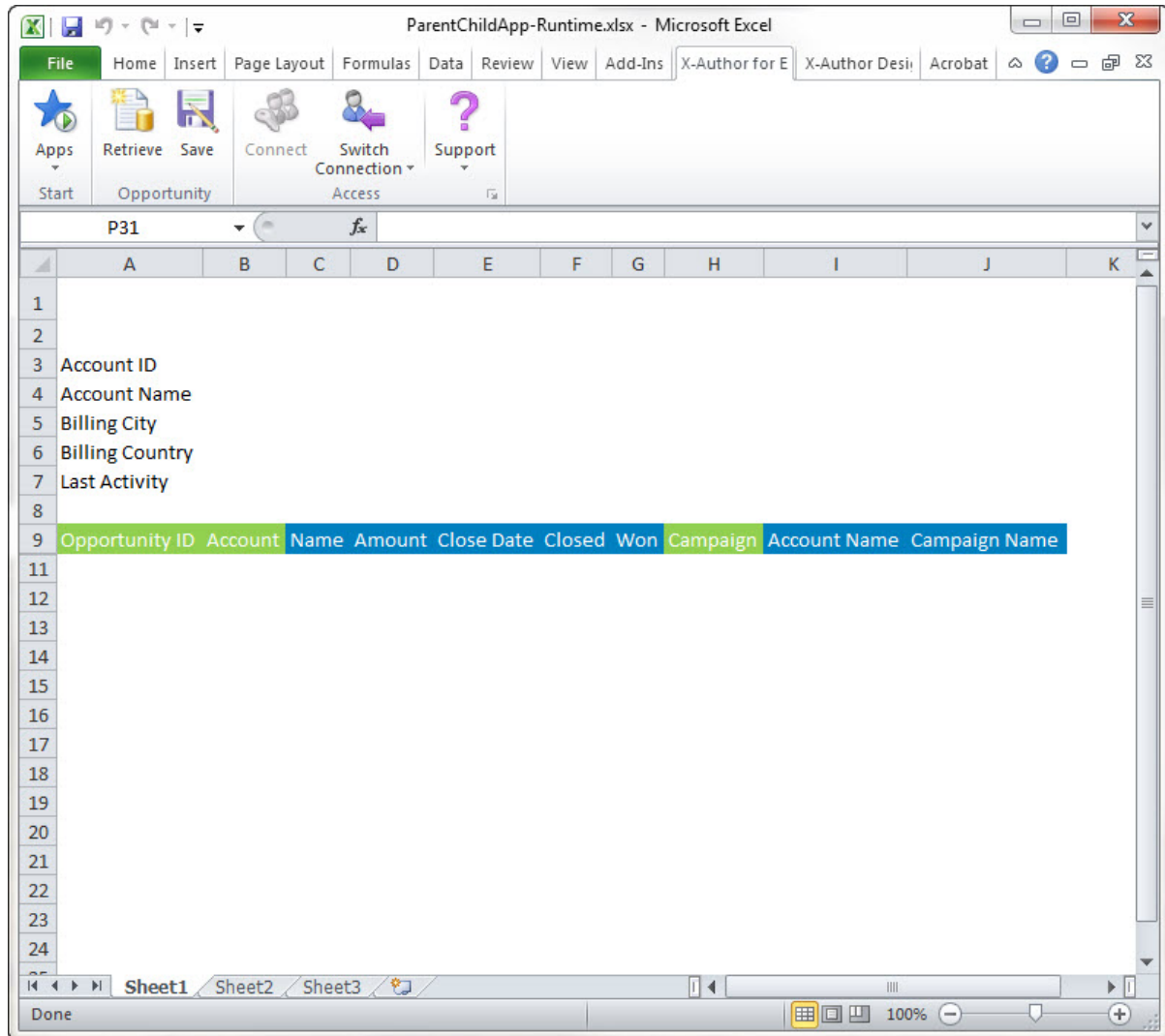


You can now test your app by retrieving data and adding, pasting, and deleting rows.



## Example of previewing an app

In this example, we will 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.)

Account Name	Billing City	Billing Country	Last Activity
dj			
Acme	New York	USA	
salesforce.com	San Francisco	USA	
Global Media	Toronto	Canada	
Automation Account			
VN Account Test1			
ExclusionBAccount			
ValidationAccount			
ExclusionAccount			
Automation Assets Account			
AutoPerfTestAccount			
Automation Multi-Location Account			
Automation Exclusion Account			
Automation Multi-Location Accoun...			
Automation Ramp Account			
Auto Price Breakup			
Automation MIL Account			
Auto PRPG Account			

Notice that you can search by *Account Name* and *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 into 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 will search for all opportunities in Toronto, Canada.

**Search Account**

Search Across Filters :    [Hide Filters](#)

Account Name :  Billing City :

1 Account record(s) retrieved

Account Name	Billing City	Billing Country	Last Activity
Global Media	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.)

**Search Opportunity**

Search Across Filters :    [Hide Filters](#)

Account :  Campaign :

Won : ☐ Close Date :

Account Name :  Campaign Name :

☒ Select All 3 Opportunity record(s) retrieved

	Account	Name	Amount	Close Date	Closed	Won	Campaign	Account Name	Campaign Name
<input checked="" type="checkbox"/>	001i0000007fx...	salesforce.co...	500000	2011-01-05	true	true		Global Media	
<input type="checkbox"/>	001i0000007fx...	salesforce.co...	50000	2011-01-05	true	true		Global Media	
<input type="checkbox"/>	001i0000007fx...	Global Media ...	40000	2011-03-08	false	false		Global Media	

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 your results.

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

The search window closes and our data is retrieved 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										

# Activating and Assigning Apps

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

## About Activating Apps

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

### To activate apps

1. Log in 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 have 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.

### **Note**

If the Assign button is not available on the Apps page, an administrator must add it to the layout. Refer to [Completing Designer Setup](#) for instructions.

## 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**.

# Managing Apps

Refer to the following sections to help you manage settings for your apps, import and export apps, and use apps to migrate data between Salesforce orgs.

## Application Settings

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

### App Settings

#### General

☐ Disable local Save File
 ☐ Disable Print
 Max. Attachment Size (in MB) :

☐ Disable Rich Text editing
 ☐ Ignore Picklist Validation During Save

#### Format

☐ Auto Size Column Width
 Max Column Width :

☐ Display Filters
 Color for Unsaved Records:

#### Suppress Messages

☐ When no records are retrieved from a Query Action
 ☐ When no records are retrieved for a parent object from a Query Action
 ☐ When no records are found to save in a Save Action
 ☐ When all records are saved successfully in a Save Action

#### Protect Sheet Setting

Name	Password

## 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 links that allow the user to add or edit rich text. For more information about rich text editing, refer to [Rich Text Fields](#).

## Ignoring Picklist Validation on Save

To prevent validation of picklist data during Save Actions: Use this option when you want the picklist validation to be ignored when saving the data back to Salesforce in runtime.

## Suppressing Messages

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

Field	Description
When no records are retrieved from a Query Action	Select this check box if you want to suppress the messages when no records are retrieved from a Query Action.
When no records are retrieved for a parent object from a Query Action	Select this check box if you want to suppress the messages when no records retrieved for a parent object from a Query Action.
When no records are found to save in a Save Action	Select this check box if you want to suppress the messages when no records are found to save in a Save Action.



Field	Description
When all records are saved successfully in a Save Action	Select this check box if you want to suppress the messages when all records are saved successfully in a Save Action.

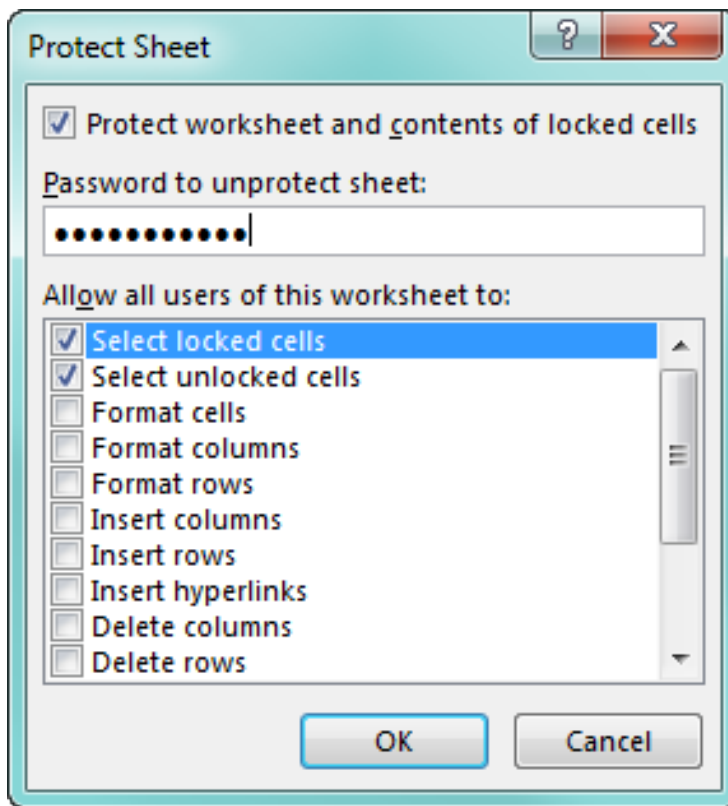
## Using Protected Worksheets

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**.

All the options of the protection settings are now supported per sheet. The protection settings are remembered (per sheet) even if you unprotect the sheets. The protection settings are reapplied when you re-protect the sheets back. X-Author for Excel will not allow the addition of rows or columns in a protected sheet.

## To protect a sheet

1. Right-click on the sheet that you want to protect and select **Protect Sheet**.
2. Set the permissions by selecting appropriate check boxes.
3. Enter the password in Password to unprotect sheet text-box.
4. Click OK.



### Note

You will receive a validation message when you provide a wrong password in App Settings in order to protect the sheet. The name of the sheet is also provided with the validation message.

## Max Attachment Size Limits

Salesforce has a File Size Limit which limits the size of the attachments to be uploaded for a single record on Salesforce. For more information regarding attachment limits refer, [File Size Attachment Limits in Salesforce](#).

If you increase the file size limits in Salesforce, you can set the file size limit per attachment in a record in X-Author for Excel. Navigate to App Settings and set the Max File Size limit. This limit indicates the maximum size of an attachment that can be uploaded to Salesforce.

Suppose the records in your application comprise the attachment field and have multiple attachments. During insert or upsert, the insert or upsert transaction request is chunked into multiple Salesforce transactions so that all the attachments are inserted or upserted successfully.

## Color for Unsaved Records

During runtime when a record is not saved due to an error, the following events occur:

- For a cell: The cell corresponding to the field in the Save map is highlighted based on the selected color.
- For a row: The row containing the error record is highlighted based on the selected color.

Set the color you want for the unsaved records or fields in App Settings.

## Importing and Exporting an Application

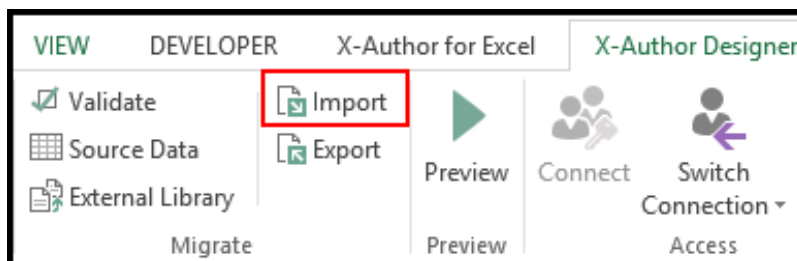
X-Author for Excel Apps can be easily migrated from one Salesforce org to another provided that the Salesforce objects being used in the exporting org match those in the importing org.

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

### To import an app

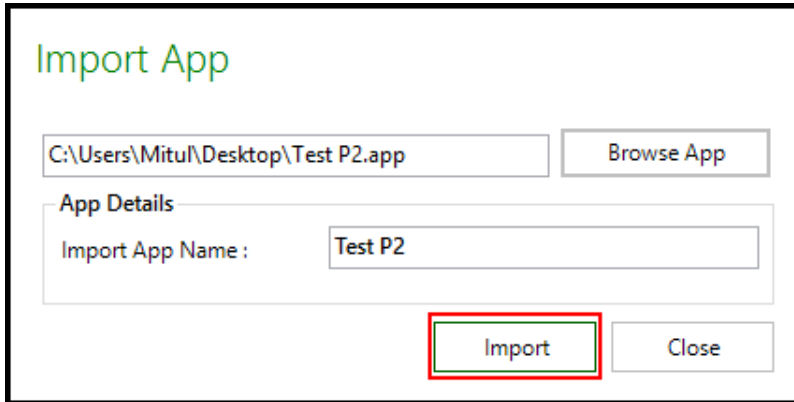
To import a previously exported X-Author for Excel .app file follow the steps below:

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 an *.app* extension and click **Open**.

4. Under App Details, in Import App Name, change the name of the App, if required, and click **Import**.



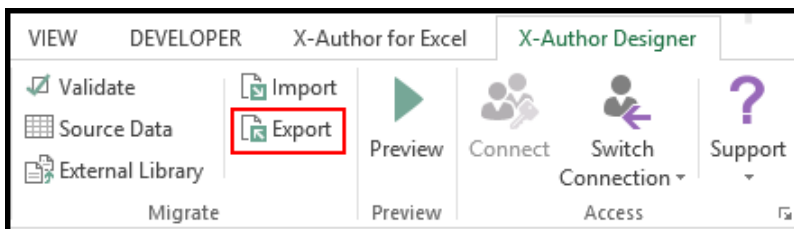
The App is imported into your current Salesforce org.

### **Best Practice**

To avoid errors using the app in the new org, make sure to **Validate** the app after the import is complete. This will identify any existing mismatches in objects and fields defined in the app. If you do encounter errors upon validation, check the objects and fields affected. Make changes to your app or in your org as needed, and perform the import process again.

## To export an app

1. Click Export.



2. Choose a location and enter a filename.

### 3. Click **Save**.

An app file with .app extension is saved at the chosen location. The Export function saves an App as an .app file. This file can then be used by the Import function to add that App to a different Salesforce org. You should also use this function to back up your apps while creating them.

## Migrating Data Between Salesforce Orgs



Refer to [Using the Data Migration Wizard](#) for step-by-step instructions using the Data Migration Wizard to create an app for data migration between orgs.

Migrating data requires the following functions:

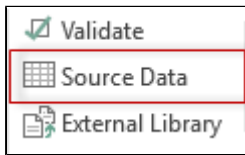
- Source Data provides a mapping mechanism for helping to migrate data between Salesforce orgs.
- It is combined with the Paste from Mapping menu option in runtime or Paste Source Data action function to copy data from an entire workbook to another workbook in the desired format.

The most common use case for this is for migrating data from one Salesforce org to another far more easily and quickly than can be done with the traditional single object data loading tools. The source App is for downloading all the data you need to migrate. Each object can be represented in one worksheet for example. The target app is a replica of the source app except it is for uploading data to the target org. The Source Data function provides the mapping between the Apps.

You could also choose to create a more advanced single app that handles both the source download and target upload. For this you would need to make use of both the Switch Connection and Clear Actions as well.

### 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.



4. Select **Source Excel Sheet**, locate the file above and click **Open**.
5. Use the drop-down menu 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 menu 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.

**i** 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.

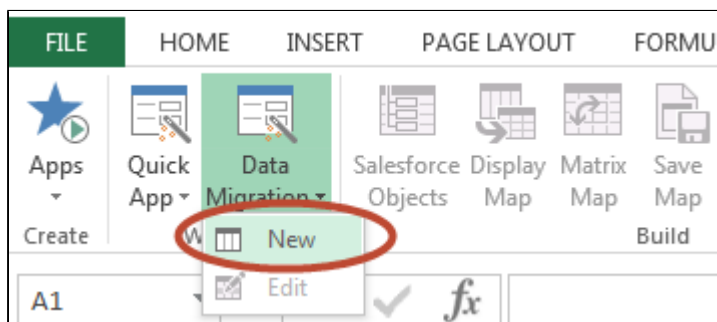
## Using the Data Migration Wizard

X-Author for Excel provides a built-in solution for migrating data from one Salesforce org to another with the **Data Migration** wizard. Using the Wizard, you can choose Objects, Fields and related Lookups to build an app configured to retrieve data from the connected source org and save data to a target org. The wizard takes care of all Display Maps, Save Maps, Action Flows and user menus so you have an app ready to use when you are done. As a designer, you can also customize the app after completing the wizard steps to tailor it to your requirements.

**i** Certain restrictions apply to app design after you have completed the Data Migration wizard. Consult the following sections for a list of restrictions.

## To start building a new Data Migration app using the Wizard

1. Go to the X-Author for Excel Designer tab and Connect to your Salesforce org.
2. From the X-Author for Excel Designer ribbon, click on the arrow next to **Data Migration**.



3. Click **New** to open the Data Migration Dashboard.

The screenshot shows the 'Data Migration Wizard' window in the X-Author Designer. The window has a title bar 'X- Author Designer' and a close button. Below the title bar, the main heading is 'Data Migration Wizard'. To the right of the heading is an 'App Name' input field. Below the heading is a descriptive text: 'Define and update your data migration app. Start by adding objects. To learn more [click here](#)'. Below this text are three buttons: 'Add Objects', 'Add Lookups', and 'Settings'. The main area of the window contains a table with the following columns: 'Salesforce Object', 'Display Name', 'Fields', 'Lookup Fields', 'Load Sequence', 'Action', 'Filter', 'External ID', 'Copy', and 'Remove'. The table is currently empty. At the bottom right of the window are two buttons: 'Create App' and 'Cancel'.

X- Author Designer

## Data Migration Wizard

App Name :

Define and update your data migration app. Start by adding objects. To learn more [click here](#)

Add Objects Add Lookups Settings

Salesforce Object	Display Name	Fields	Lookup Fields	Load Sequence	Action	Filter	External ID	Copy	Remove

Create App Cancel

4. Enter an **App Name** for your Data Migration app.

**Next step:** Add some objects and fields.

## Adding Objects and Fields to your Data Migration App

To begin building your Data Migration app, you need to configure which Objects and Fields should be part of the migration. If you are working with an existing Data Migration app, you can make changes to the Objects and Fields already present in the app in the same way.

When you create a new Data Migration app or edit an existing one, the following screen is displayed.

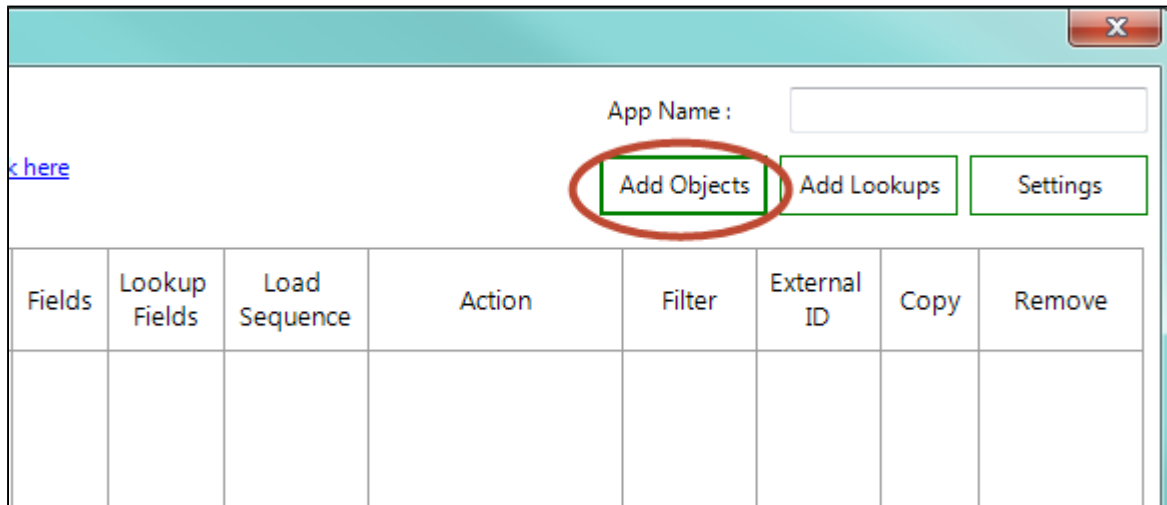
The screenshot shows the 'Data Migration Wizard' window in the X-Author Designer. The window has a title bar 'X-Author Designer' and a close button. Inside, the title 'Data Migration Wizard' is in green. Below it, a subtitle reads 'Define and update your data migration app. Start by adding objects. To learn more [click here](#)'. In the top right, there is an 'App Name' input field and three buttons: 'Add Objects', 'Add Lookups', and 'Settings'. The main area contains a table with 10 columns: 'Salesforce Object', 'Display Name', 'Fields', 'Lookup Fields', 'Load Sequence', 'Action', 'Filter', 'External ID', 'Copy', and 'Remove'. The table is currently empty. At the bottom right, there are two buttons: 'Create App' and 'Cancel'.

Salesforce Object	Display Name	Fields	Lookup Fields	Load Sequence	Action	Filter	External ID	Copy	Remove
-------------------	--------------	--------	---------------	---------------	--------	--------	-------------	------	--------

## To Add Objects to your App

1. From the upper right-hand corner of the screen, click on **Add Objects**.





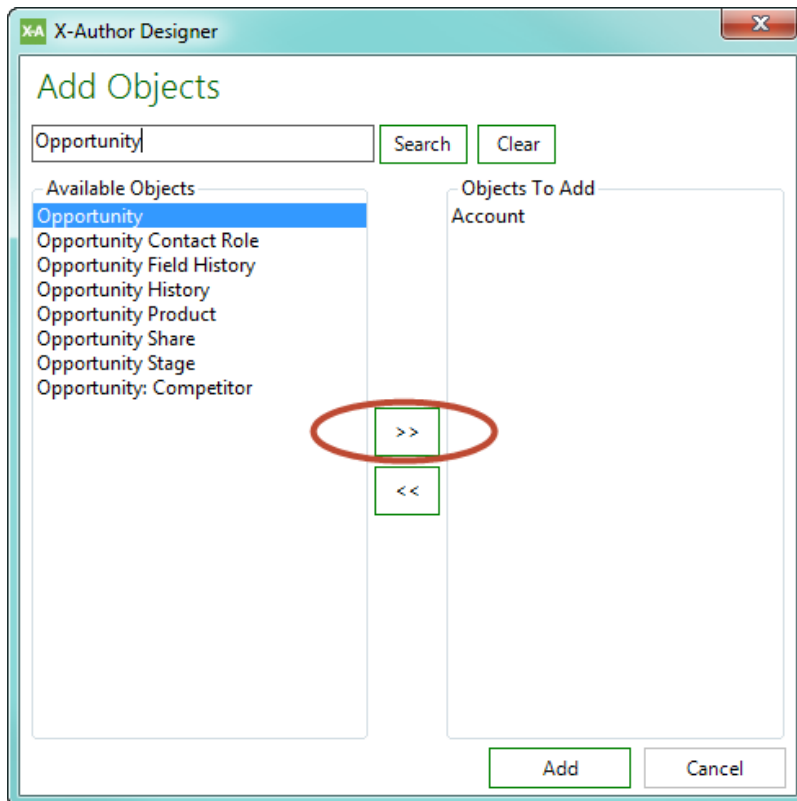
App Name :

[Click here](#)

**Add Objects** **Add Lookups** **Settings**

Fields	Lookup Fields	Load Sequence	Action	Filter	External ID	Copy	Remove

2. Double-click on an Object or highlight and Object and use the arrow button to add Objects to your app. To refine the list of Available values, click on **Search**.



**Add Objects**

Opportunity| **Search** **Clear**

**Available Objects**

- Opportunity
- Opportunity Contact Role
- Opportunity Field History
- Opportunity History
- Opportunity Product
- Opportunity Share
- Opportunity Stage
- Opportunity: Competitor

**Objects To Add**

Account

**>>** **<<**

**Add** **Cancel**

3. Click **Save** to add the selected Objects to your app. The Data Migration Dashboard updates with an entry for each Object you added.

**Data Migration Wizard**

Define and update your data migration app. Start by adding objects. To learn more [click here](#)

App Name :

[Add Objects](#) [Add Lookups](#) [Settings](#)

Salesforce Object	Display Name	Fields	Lookup Fields	Load Sequence	Action	Filter	External ID	Copy	Remove
Opportunity Product	Opportunity Product	<a href="#">2</a>	0	1	Search & Select		Set		
Account	Account	<a href="#">2</a>	0	2	Query		Not Set		
Opportunity	Opportunity	<a href="#">2</a>	0	3	Query		Not Set		

[Create App](#) [Cancel](#)

### **Note**

Do not worry about the sequence of Objects in your app at this time. Sequence will adjust depending on how External ID and Lookups are set.

## To Add Fields to your App

After at least one Object has been added to your Data Migration app, you can add fields for each selected Object.

1. From the Data Migration Wizard screen, click on the link under the **Fields** column corresponding to the Object you want to work with.

Account	<a href="#">2</a>	0
Opportunity	<a href="#">2</a>	0

2. Select the check box for each field you want to add to your app. To refine the list of fields, enter text into the **Search** field. Note: You cannot deselect the Object fields ID or Name.

**Opportunity Fields**

Search Fields :  [Clear](#)

	Field Id	Field Name	Data Type
<input checked="" type="checkbox"/>	Amount	Amount	Decimal
<input checked="" type="checkbox"/>	CloseDate	Close Date	Date
<input checked="" type="checkbox"/>	Name	Name	String
<input checked="" type="checkbox"/>	Id	Opportunity ID	String
<input checked="" type="checkbox"/>	StageName	Stage	Picklist
<input type="checkbox"/>	AccountId	Account	Lookup
<input type="checkbox"/>	Alt_Account__c	Alt Account	Lookup
<input type="checkbox"/>	Approval_Comments__c	Approval Comments	Textarea
<input type="checkbox"/>	Approval_Reason_Code__c	Approval Reason Code	Picklist
<input type="checkbox"/>	Approval_Required__c	Approval Required	String
<input type="checkbox"/>	Apttus_Approval_Approval_Status__c	Approval Status	Picklist
<input type="checkbox"/>	Approval_Submitter__c	Approval Submitter	Lookup
<input type="checkbox"/>	Approval_Valid_to_Date__c	Approval Valid to Date	Date

[Apply](#) [Close](#)

3. Click **Apply** to add fields to your app for the selected Object.
4. Repeat steps 1 – 3 for each Object in your app, as required.





If Fields for one of the Objects in your app have been previously added, you can click the hyperlink under the "Fields" column to add or remove more Fields.

## Working with the Data Migration Dashboard

After adding Objects and Fields to your app, you can take the following actions as needed:

- Specify a unique **Display Name** for each Object in the app.
- Click on the link under the **Fields** column to add or remove more Fields for an Object.

- Click on the link under the **Lookup Fields** column to add or remove more Lookup Fields for an Object.
- Click on the  **Copy** icon to copy an Object listing and modify it as required (for example, you want to add the same Object and a different set of Fields for filtering purposes).
- Click on the  **Remove** icon to remove an Object from your app.
- Specify **Load Sequence**, **Actions**, and **Filters**. Refer to [Configuring your Data Migration App](#) for step-by-step instructions for these tasks.

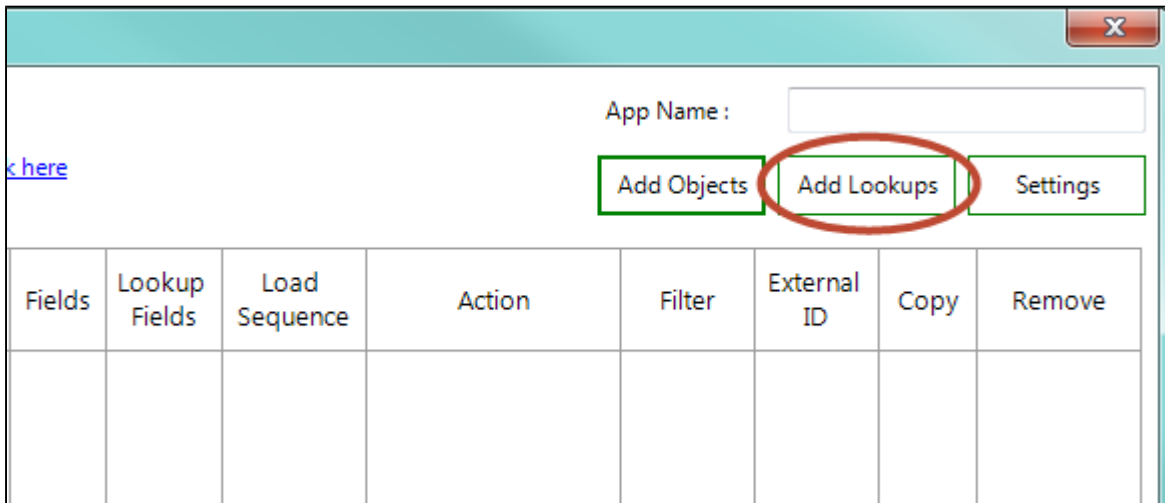
**Next:** [Add Lookups to your app.](#)

## Adding Lookups to your Data Migration App

In addition to Objects and their Fields, you can also add Lookup fields that reference other Objects. The Lookup Object can only be referenced if it has been included in the fields for that Object in Salesforce. Also, fields already added to your app will not be visible in the Lookup dialog.

### To add Lookup Fields to your App

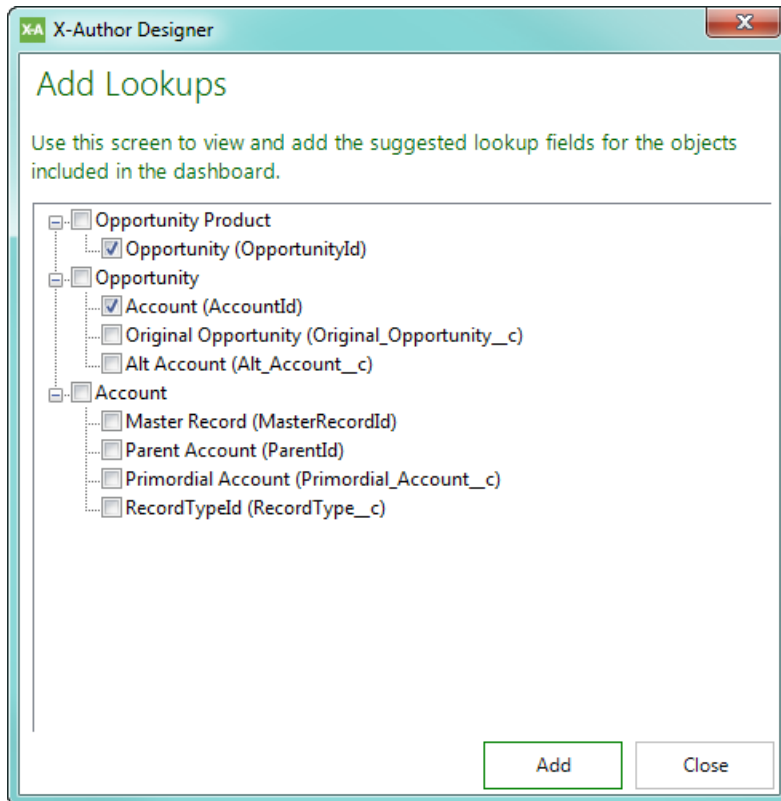
1. From the Data Migration screen, click on **Add Lookups**.



The screenshot shows a window titled "X" with a teal header. Below the header, there is a text input field labeled "App Name :". To the left of this field is a blue link labeled "Click here". Below the "App Name" field are three buttons: "Add Objects", "Add Lookups", and "Settings". The "Add Lookups" button is circled in red. Below these buttons is a table with the following columns: Fields, Lookup Fields, Load Sequence, Action, Filter, External ID, Copy, and Remove. The table has one empty row below the headers.

Fields	Lookup Fields	Load Sequence	Action	Filter	External ID	Copy	Remove

2. Navigate the Object/Field tree and select a check box to add one or more Lookup fields to your app. Click the check box next to an Object name to select all Lookup fields for that Object. Fields that have already been added to your app are not displayed.



3. Click **Add** to add all selected Lookup fields to your app.
4. Click **OK**. The Lookup fields are added to all related Objects in the app. When Lookup fields are added:
  - Each affected Object displays the number of Lookup fields that will be migrated when the app is run.
  - Fields representing the Lookup fields are added to the Object under the "Fields" column.
  - You can click on the link under the "Lookups" column to view all Lookup fields added for a specific Object.

- The Object sequence is automatically adjusted based on Lookup relationship between Objects (the sequence for dependent Objects are changed).
- A filter (see below) is automatically created for the Object. **Do not remove this filter or you will encounter errors during migration.**

The screenshot shows the 'X-Author Designer' window with a 'Filters' tab. It contains a table with columns: 'Salesforce Object and Field', 'Value Type', 'Operator', and 'Value'. Row 1 has 'Account' in the first column, 'Input' in the second, 'in' in the third, and 'Account.Account ID' in the fourth. There is a red 'X' icon to the right of the 'Value' field. Below the table are links: 'Add Row', 'Clear All', and 'Add Filter Logic'. At the bottom right are 'Save' and 'Cancel' buttons.

	Salesforce Object and Field	Value Type	Operator	Value
1	Account	Input	in	Account.Account ID

[Add Row](#) [Clear All](#) [Add Filter Logic](#)

Save Cancel

**Next:** [Configure settings](#) or [complete app configuration](#).

## Configuring your Data Migration App

Before your Data Migration App can be finalized and ready to use, you need to specify load behavior, actions and filters for data retrieval.

## Determining Load Sequence

By default, Salesforce Objects retrieved from the source org will load in the sequence listed on the Data Migration screen. If you want to change the load sequence behavior, you need to specify a new load sequence for Objects in the list.

To accomplish this, edit the **Load Sequence** column for your objects. Use caution when making changes, as load sequence is important when migrating data. Users will encounter errors, for example, if a Child Object load sequence is specified prior to its Parent Object. Also, you cannot finalize the app if there is a number missing in the load sequence or two or more Objects share the same Load Sequence.

## Choose a User Action for each Object

Each Object in the Data Migration must have a corresponding action in the runtime app. This allows the user to make use of X-Author's Search and Select or Query actions to request data from the source org during the migration.

To choose actions, click the drop-down menu next to each Object and select either **Search & Select** or **Query**.

Salesforce Object	Display Name	Fields	Lookup Fields	Load Sequence	Action
Account	Account	<u>9</u>	0	1	Search & Select ▾
Opportunity Product	Opportunity Product	<u>4</u>	<u>1</u>	3	Search & Select Query

For both **Search & Select** and **Query** actions, the Options are automatically filled based on the Objects and Fields you added to your app. You can make changes to these options as required when you define *Filters*.




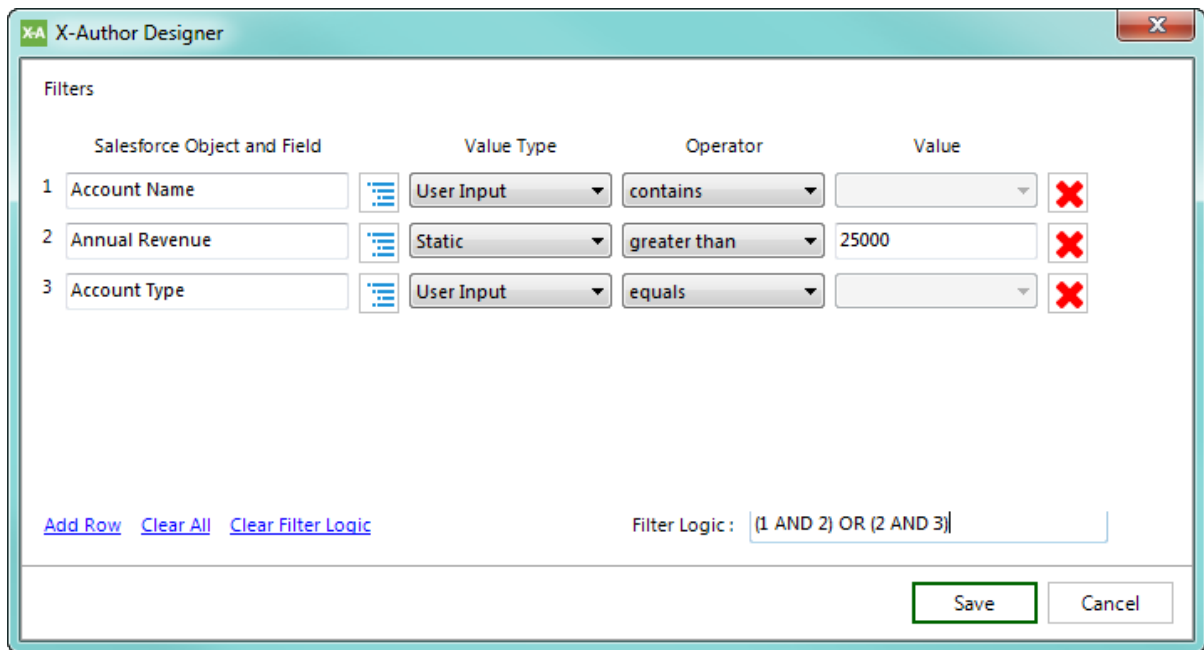
For more information on these actions, refer to [Search and Select](#) and [Query](#) actions in this guide




## Filtering Search and Select and Query Actions

To refine the data selected for migration when the user goes through the Data Migration app Action Flow, you can specify one or more filters for each Object being migrated. The process for defining filters is the same for each action type.

To define Filters for an action:

1. Click on the  **Filter** icon next to the corresponding action.
2. Define one or more filters to apply to data retrieved for the selected Object and its Fields:
  - a. Use the Expression builder to specify a **Value Type**, **Operator**, and **Value** for your filter.
  - b. Click **Add Row** to create an additional filter.
  - c. Click **Add Filter Logic** to specify additional logic for the filters you create.



	Salesforce Object and Field		Value Type	Operator	Value	
1	Account Name		User Input	contains		✗
2	Annual Revenue		Static	greater than	25000	✗
3	Account Type		User Input	equals		✗

[Add Row](#) [Clear All](#) [Clear Filter Logic](#)

Filter Logic: (1 AND 2) OR (2 AND 3)

[Save](#) [Cancel](#)

3. Click **Save** when you are finished defining Object and Field filters.



When building filters, you can only create expressions for Objects and/or Fields already added to the app.

For complete instructions on building S&S and Query filters, refer to [Query Action](#).



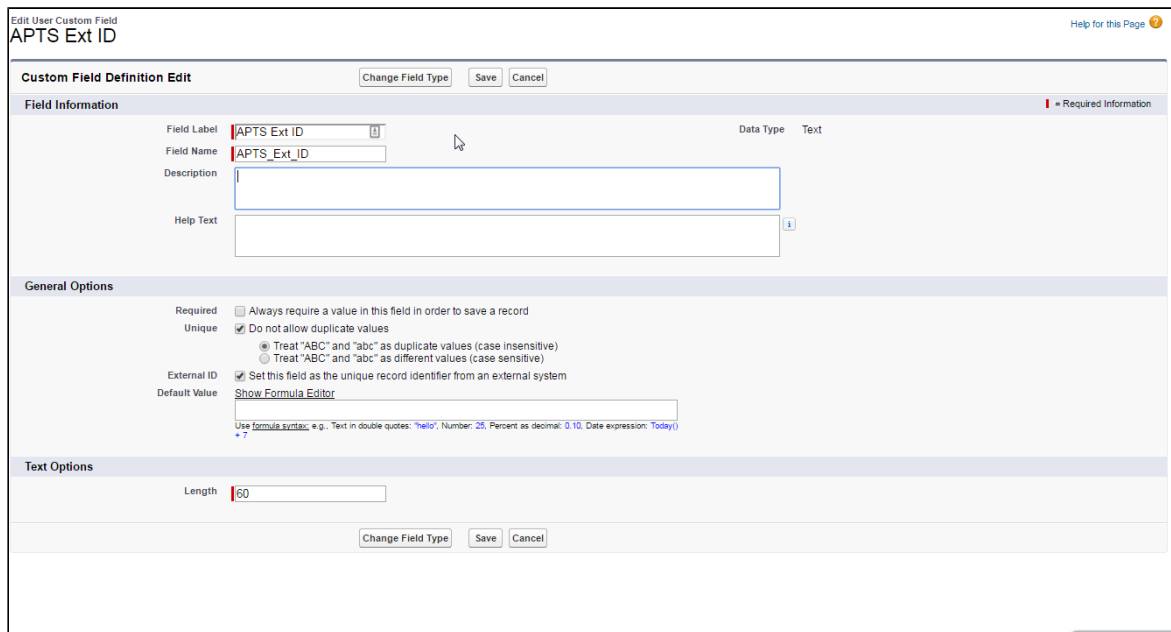
## Specify External ID

This is an optional step that is only required if your Object has more than one Salesforce External ID. Most Objects will only have a single External ID, but in some cases an Object will have more than one (for example, you export data for this object to more than one external source). Setting the External ID is important to prevent duplication of records during data migration.

### Note

All Objects must have a set External ID. **If an Object does not have an External ID set, you will not be able to save the app.** It is recommended to configure External ID's in Salesforce for every Object that you want to migrate before creating the Data Migration app. You can configure External ID's for your Objects in one of two ways:

- Use Apttus' Metadata utility to create External IDs for each Object you want to migrate. Contact your Apttus representative to get access.
- Configure External ID's for your Objects using Salesforce. Create a text field labeled **APTS Ext ID**, field name **APTS\_Ext\_ID** per configuration in the image below. Refer to Salesforce administrator documentation for help on how to add fields to your Objects.



The screenshot shows the 'Edit User Custom Field' interface for a field named 'APTS Ext ID'. The interface is divided into three main sections: Field Information, General Options, and Text Options.

**Field Information:**

- Field Label:** APTS Ext ID
- Field Name:** APTS\_Ext\_ID
- Description:** (Empty text area)
- Help Text:** (Empty text area)
- Data Type:** Text

**General Options:**

- Required:** ☐ Always require a value in this field in order to save a record
- Unique:** ☒ Do not allow duplicate values
  - ☒ Treat "ABC" and "abc" as duplicate values (case insensitive)
  - ☐ Treat "ABC" and "abc" as different values (case sensitive)
- External ID:** ☒ Set this field as the unique record identifier from an external system
- Default Value:** [Show Formula Editor](#)

**Text Options:**

- Length:** 60

At the bottom of the interface, there are buttons for 'Change Field Type', 'Save', and 'Cancel'.

- You will need to restart X-Author after adding the External IDs.

When an Object has more than one External ID, a Not Set link will be visible under the "External ID" column.

## To specify External ID for an Object

- From the Data Migration screen, click **Not Set** under the "External ID" column corresponding to the Object you want to set.

App Name : CC DM Wizard App 1

Load Sequence	Action	Filter	External ID	Copy	Remove
1	Search & Select ▼		<u>Not Set</u>		
3	Query ▼		Set		

- Using your cursor, highlight the correct External ID field to use for your Object.

X-A X-Author Designer







### Account ExternalIDs

Field Id	Field Label
ExternalId_c	External Id Field
APTS_Ext_ID_c	APTS Ext ID

- Click **Apply**. The chosen External ID is updated in the Data Migration Object listing.

App Name : CC DM Wizard App 1

[Add Objects](#)
[Add Lookups](#)
[Settings](#)

Load Sequence	Action	Filter	External ID	Copy	Remove
1	Search & Select ▾		<a href="#">Set</a>		
3	Query ▾		Set		

**Next:** [Finalizing your app.](#)

## Configuring Data Migration App Settings







You can specify settings for your Data Migration app to customize the Action Flow, User Menus, Filters, and place restrictions on run-time actions.

### To configure settings for your Data Migration App

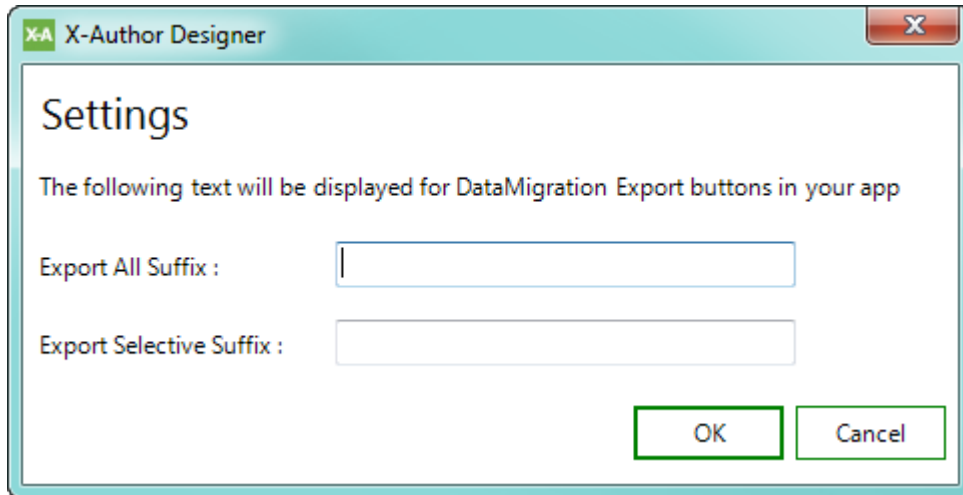
1. From the Data Migration screen, click on **Settings**.

App Name : CC DM Wizard App 1

[Add Objects](#)
[Add Lookups](#)
[Settings](#)

	Action	Filter	External ID	Copy	Remove
1	Search & Select ▾		<a href="#">Set</a>		
3	Query ▾		Set		

2. Specify values for your app settings. Refer to the table below for a list of settings and their descriptions.

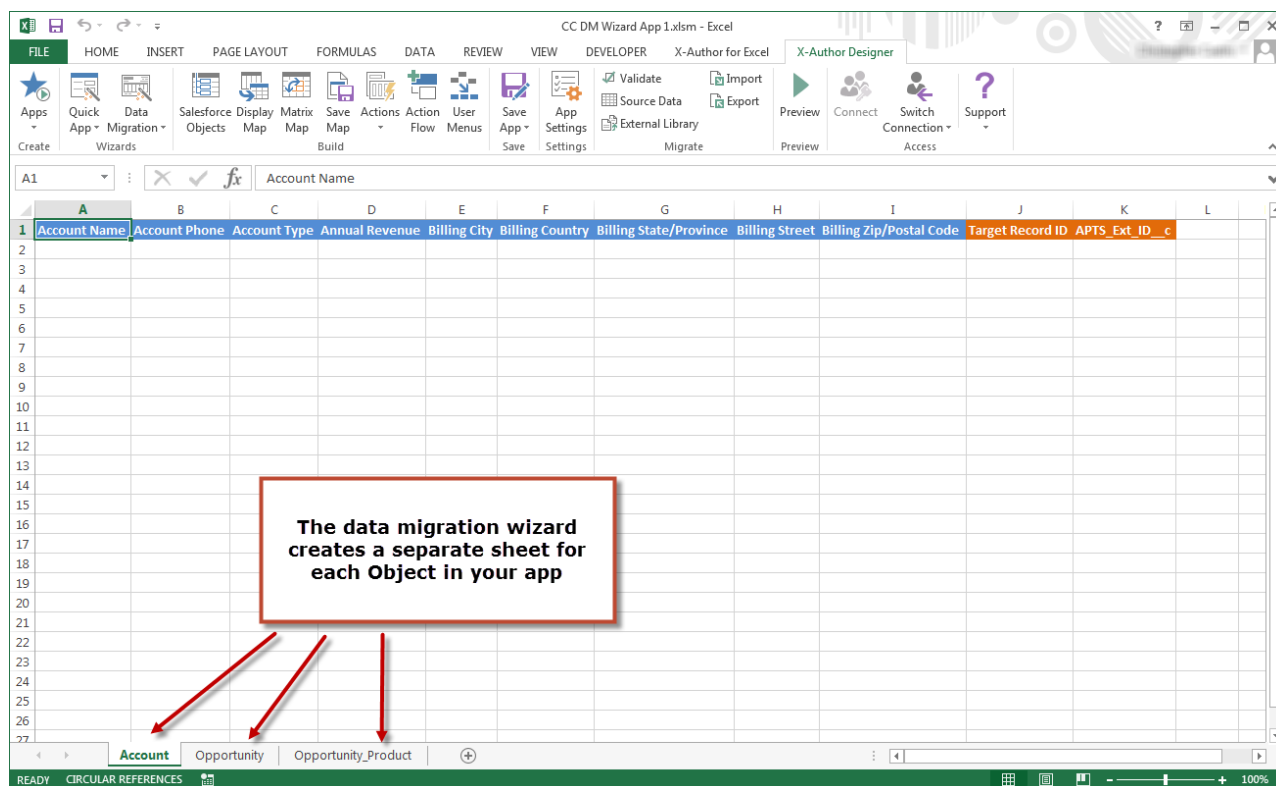


Setting	Description	Guidance
Export All Suffix	Enter custom text to replace "All" for the "Export All" button of the app.	Limit the suffix to fit text in the X-Author ribbon.
Export Selective Suffix	Enter custom text to replace "Selective" for the "Export Selective" button of the app.	Limit the suffix to fit text in the X-Author ribbon. Suffix should represent filters defined for your Objects.

3. Click **OK** to apply your settings.

## Finalizing your Data Migration App

When you are finished creating your Data Migration App, return to the Data Migration screen and click **Create App** (or **Update App** if you are editing). The Wizard creates or updates the app based on your specifications and displays the resulting app as shown in the example below.



When you create or update a Data Migration app it is automatically saved once you click Create App/Update App. Please remember that if you make any additional changes to the app, you must save the app in X-Author.

## Making Changes to your Data Migration App

Since the Data Migration app is just another X-Author app, created using the Wizard instead of the individual designer tools, most elements of the app can still be modified by the designer before the app is finalized. Click on any of the buttons in the Designer tab to see the Display Maps, Save Maps, Actions and User Menus created by the Data Migration Wizard.



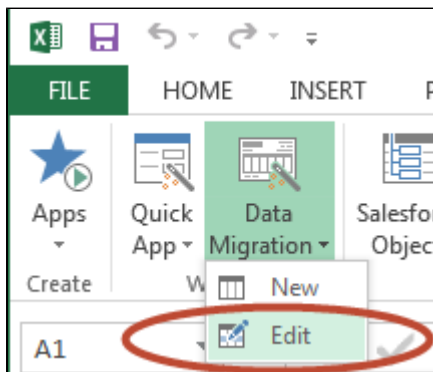
Be especially careful when modifying a Data Migration app after you create it with the Wizard. Because the app is assembled using the Wizard, you may inadvertently break the functionality if you modify maps or remove other components. If you want to add or

modify any Objects or Fields, it is recommended to **Edit** the app using the Data Migration Wizard (see below), rather than editing Display/Save Maps and other elements of the app directly.

Add Custom Actions or User Menus to your app as required to finalize your Data Migration App.

## Editing your Data Migration App Using the Wizard

When you want to modify a Data Migration App to add, remove or change Objects, Fields, Lookups, or change Filters and other settings, go to the Data Migration Wizard and select **Edit**.



**i** If the changes you want to make to the Data Migration App are significant enough, it is recommended to create a new app rather than modify an existing one.

Please note that you cannot change the following components of an app when you are editing it:

- App Name
- User Actions
- External ID

If you want to make changes to any of the above, you can still remove an Object from the app and add it back again, making any changes as required.

## Performing the Data Migration

This page provides instructions for performing a Data Migration from one Salesforce org to another using an app created with the [Data Migration Wizard](#). Instructions may vary slightly depending on if any customizations were made to the app outside of the Wizard. For help with specific user actions or action flows, refer to the appropriate section in this guide.

### Prerequisites

- The app to be activated for use by the user doing the migration.
- The user must also have access to the X-Author for Excel Designer tab.

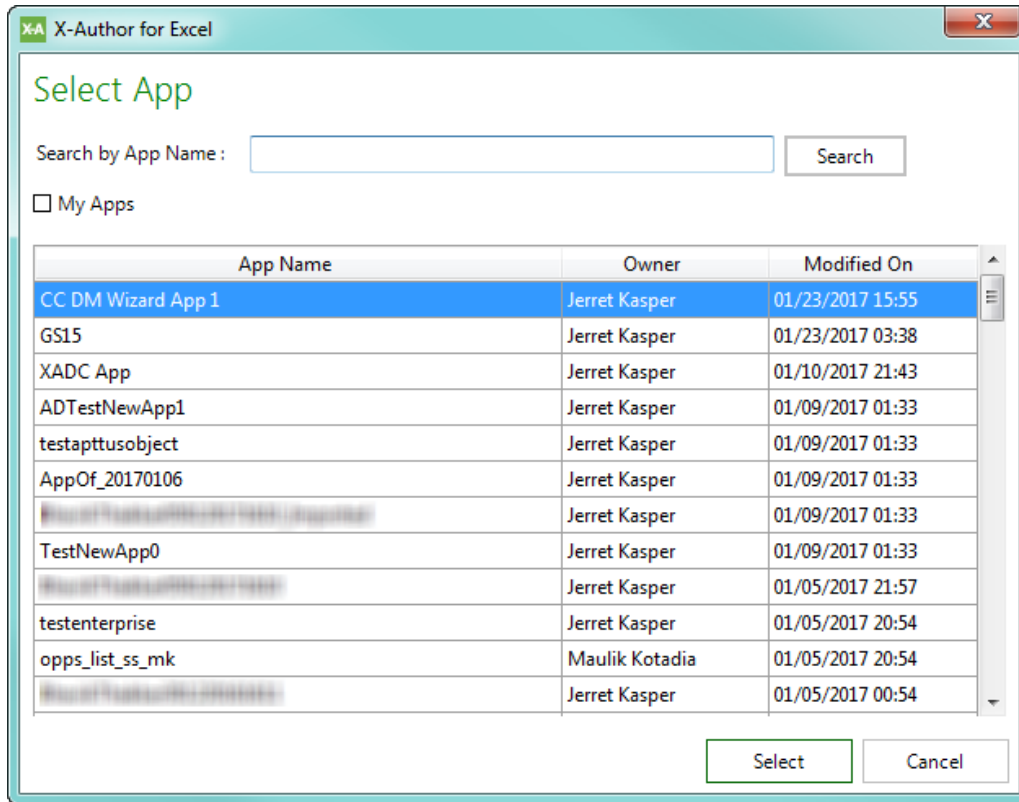
### To perform a data migration

1. From the X-Author Designer tab, export the app to your computer and Import the app to the Salesforce org where your source data resides. Follow the instructions under [Importing and Exporting an Application](#), then move to step 2 on this page.

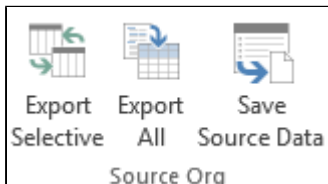
#### **Import Best Practice**

To avoid errors using the app in the new org, make sure to **Validate** the app after the import is complete. This will identify any existing mismatches in objects and fields defined in the app. If you do encounter errors upon validation, check the objects and fields affected, make changes to your app or in your org as needed, and perform the import process again.

2. Click on the **X-Author for Excel** tab to switch to the runtime view.
3. Select **App > (your Data Migration App Name)** to open the app.



4. Populate the app by choosing one of two options:



- **Export Selective** - Choose this action if you want to search for and select specific records to export from the source org for each Object in the data migration app. After making your selections, click **Next** to retrieve data into the app.



Use this option when there are Filter criteria defined for any of the Objects in the app. User actions defined for Objects in the app (like Search & Select) only execute when this option is chosen.



**Search Account**

Search Across Filters :    [Hide Filters](#)

Account Type :

☐ Select All 13 Account record(s) retrieved

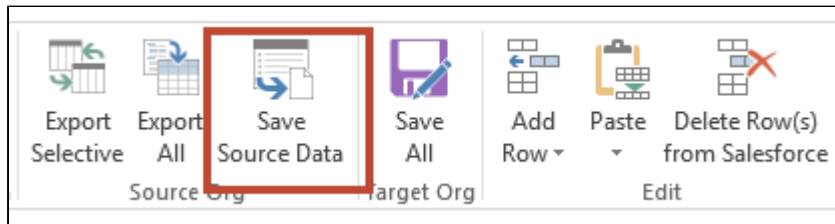
	Account Name
<input checked="" type="checkbox"/>	Global Media
<input type="checkbox"/>	ABC Corporation12
<input type="checkbox"/>	ABC Food 1231
<input type="checkbox"/>	ABC Corporation, Japan
<input type="checkbox"/>	ABC Corporation International
<input type="checkbox"/>	ABC Corporation, UK
<input type="checkbox"/>	ABC Food 123
<input type="checkbox"/>	Global Media
<input type="checkbox"/>	ABC Corporation12
<input type="checkbox"/>	ABC Food 123
<input type="checkbox"/>	ABC Corporation, Japan
<input type="checkbox"/>	ABC Corporation, UK
<input type="checkbox"/>	ABC Corporation, UK

- **Export All** - Choose this action if you want to export ALL records from the source org for each Object in the data migration app. No filters are applied.

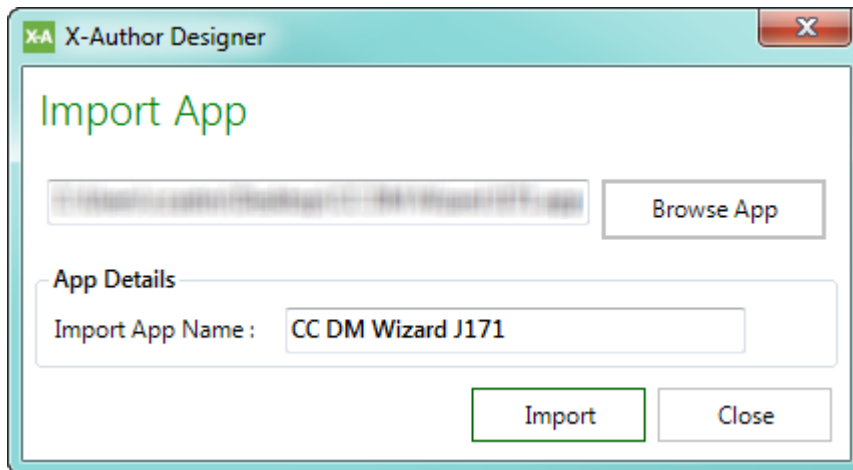
**Note**

This option may take an unusually long amount of time to retrieve data if there are a large number of records in your Org.

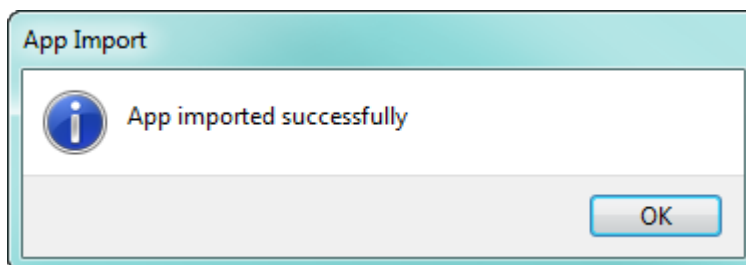
5. From the Source Org section of the X-Author Excel tab, click **Save Source Data**.
6. Browse to a location on your system to save the workbook. Name the file and click **Save**. All data retrieved by the Export action is saved to the app.



7. Close the app.
8. From the X-Author Designer tab, click on **Switch Connection** and select the target org for data migration. Enter your credentials and connect.
9. You must now import the Data Migration app you exported in Step 1 into the target org. Select **Migrate > Import** and browse to the app file.

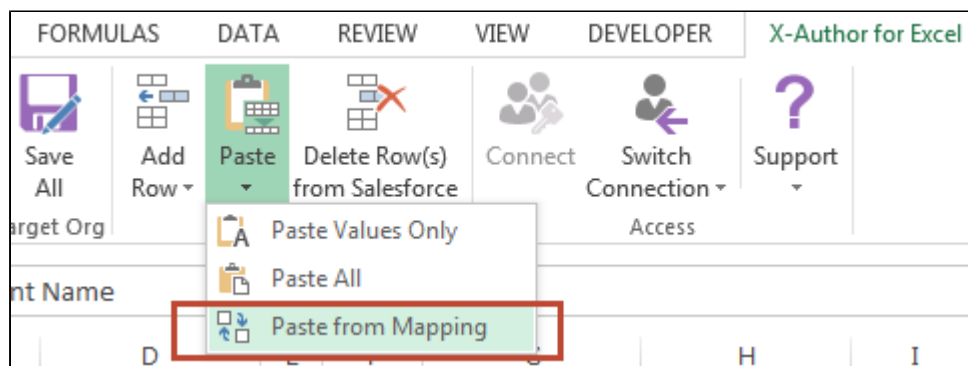


10. Click **Import**. Verify the app imported successfully and click **OK**.

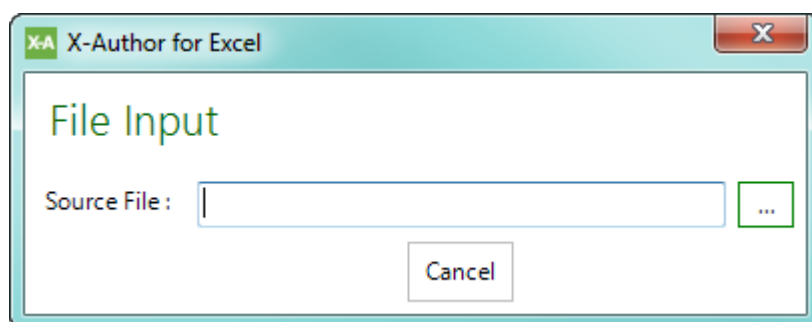


11. Click on the **X-Author for Excel** tab to switch to the runtime view.
12. Connect to the target org.

13. Click on **Apps > Open App** and open the imported app.
14. From the Edit section, click on Paste and select **Paste from Mapping**.



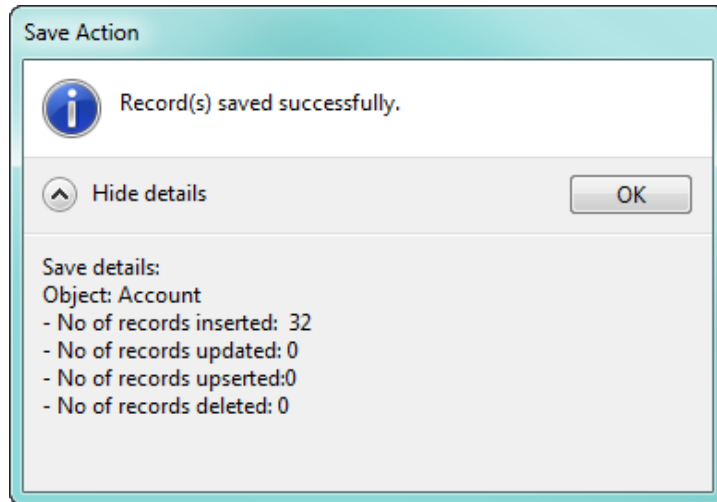
15. Browse to the app file save in Step 5 (saved source data) and click **Open**.



### **Note**

Source data that you exported is pasted into the app without **Target Record ID**. Record IDs will be assigned to all new records that are created when you migrate that data to the Target Org (but will not be represented in the app).

16. From the Target Org section of the ribbon, click **Save All**. The exported records from the source org are migrated to the target org. Depending on how much data you are migrating, this can take some time. When data migration is complete, a separate success message is shown for each Object being migrated. Click on **Show details** to view details.



### **Note**

The success message may not be shown if settings have been changed in X-Author Excel to suppress messages. Refer to [Application Settings](#) for more information.

## Updating Records during Migration

As part of the migration, you may choose to update (or upsert) existing records in the target org with updated data from the source org. In this case, you are updating or upserting data from the source org records to target org records. You do this by modifying the exported data from the source org prior to migration. You can also update/upsert records following the initial migration by reusing source data in a subsequent migration. The following use case shows an example of this.

### **Use Case: Updating Records after Migration**

Diana Designer has performed an initial migration of data from a source production org to a target production org. After the migration, she decides that she would like to perform some updates to some Opportunities in the migration. Diana can use X-Author functionality to update (or upsert) records by modifying the same source data she used during the initial migration.

She performs the following steps to update the migration data:

1. Diana connects to the **Target Org** and reopens the Data Migration app in **X-Author for Excel Runtime**.
2. She selects **Edit > Paste from Mapping**. Diana browses to the source data workbook from the previous migration and opens it in the app. The **Account** worksheet is displayed.

	A	B	C	D	E	F	G	H
1	Account Name	Account Phone	Account Type	City	Ann	Target Record ID	APTS Ext ID c	
3	ABC Corporatio	(650) 555 3338	Customer		##		001i000000Q9c8kAAB	
4	Acme1	(650) 555 3334	Customer	New	##		001i000000Q9c8IAAB	
5	Cisco Inc 1	(650) 555 3344	Customer	Wate	##		001i000000Q9c8rAAB	
6	Salesforce	(650) 555 3336	Customer	Sura	##		001i000000Q9c8tAAB	
7	GS1	123456789	Customer		##		001i000000Q9c9GAAR	
8	Zipcar & co	(650) 555 3350	Customer		##		001i000000Q9c9YAAR	
9	Venture Industi	(650) 555 3381	Customer		##		001i000000Q9c9aAAB	
10	ABC Corporatio	(650) 555 3639	Customer		##		001i000000xsjhoAAA	
11	Cisco Inc	(650) 555 3644	Customer		##		001i000000xsjhuAAA	
12	Zipcar & co	(650) 555 3350	Customer		##		001i000001gZoG7AAK	
13								
14								

Account Opportunity Opportunity\_Product (+)

3. Diana chooses to update some Account phone numbers.

	A	B	C	D	E	F	G	H
1	Account Name	Account Phone	Account Type	City	Ann	Target Record ID	APTS Ext ID c	
3	ABC Corporatio	(650) 555 3338	Customer		##		001i000000Q9c8kAAB	
4	Acme1	(650) 555 8334	Customer	New	##		001i000000Q9c8IAAB	
5	Cisco Inc 1	(650) 555 3344	Customer	Wate	##		001i000000Q9c8rAAB	
6	Salesforce	(650) 555 3336	Customer	Sura	##		001i000000Q9c8tAAB	
7	GS1	(888) 555 1234	Customer		##		001i000000Q9c9GAAR	
8	Zipcar & co	(650) 555 3350	Customer		##		001i000000Q9c9YAAR	
9	Venture Industi	(650) 555 3381	Customer		##		001i000000Q9c9aAAB	
10	ABC Corporatio	(650) 555 3639	Customer		##		001i000000xsjhoAAA	
11	Cisco Inc	(650) 555 3644	Customer		##		001i000000xsjhuAAA	
12	Zipcar & co	(650) 555 3350	Customer		##		001i000001gZoG7AAK	
13								
14								

Account Opportunity Opportunity\_Product (+)

4. She then moves to the Opportunity worksheet.

	A	B	C	D	E	F	G	H	I
1	Name	Amount	Close Date	Opportunity Type	Stage	Target Record ID	Account	APTS Ext_ID_c	Account
3	Opp ch	#####	11/22/2016	New Business	Test Stage 1		001i0000	006i000000f34Ed	0
4	Salesf	#####	6/8/2015	New Business	Perception Analysis		001i0000	006i000000gZd79	0
5	North	#####	1/13/2016	Existing Business	Value Proposition		001i0000	006i000000geQu	0
6	Acme	#####	5/30/2015	Existing Business	Needs Analysis		001i0000	006i000000ZRDO	0
7	Partne	#####	6/4/2015	New Business	Needs Analysis		001i0000	006i000000bkMn	0
8	Ventu	#####	2/1/2015	Existing Business	Id. Decision Makers		001i0000	006i000000GcJUu	0
9	SaveP	#####	1/7/2013	Existing Business	Qualification		001i0000	006i000000LpbirA	0
10									
11									
12									
13									
14									

Account Opportunity Opportunity\_Product (+)

5. Diana modifies some Opportunity amounts.

	A	B	C	D	E	F	G	H	I
1	Name	Amount	Close Date	Opportunity Type	Stage	Target Record ID	Account	APTS Ext_ID_c	Account
3	Opp ch	30,000.00	11/22/2016	New Business	Test Stage 1		001i0000	006i000000f34Ed	0
4	Salesf	15,000.00	6/8/2015	New Business	Perception Analysis		001i0000	006i000000gZd79	0
5	North	15,000.00	1/13/2016	Existing Business	Value Proposition		001i0000	006i000000geQu	0
6	Acme	15,000.00	5/30/2015	Existing Business	Needs Analysis		001i0000	006i000000ZRDO	0
7	Partne	15,000.00	6/4/2015	New Business	Needs Analysis		001i0000	006i000000bkMn	0
8	Ventu	12,000.00	2/1/2015	Existing Business	Id. Decision Makers		001i0000	006i000000GcJUu	0
9	SaveP	122,232.00	1/7/2013	Existing Business	Qualification		001i0000	006i000000LpbirA	0
10									
11									
12									
13									
14									

Account Opportunity Opportunity\_Product (+)

6. Diana selects **Target Org > Save All**. She has successfully upserted the migrated records.

### Note

You can also add rows using the method to Insert new records. Because the workbook is protected, you must copy an existing row, use the "Paste Values Only" action from the X-Author ribbon and make changes (remember to remove the Target Record ID if one is displayed).



# Using Advanced App Design

The following sections provide step-by-step instructions for advanced app design features.

## Creating External Actions

Over and above the standard actions, Designers can include their own actions through the External Library capabilities. This requires the use of the X-Author SDK libraries to create custom DLLs. For example, you could create an external action for something as simple as record data cleansing, or something more complex such as invoking Oracle or SAP web services.

### Prerequisites for creating Actions

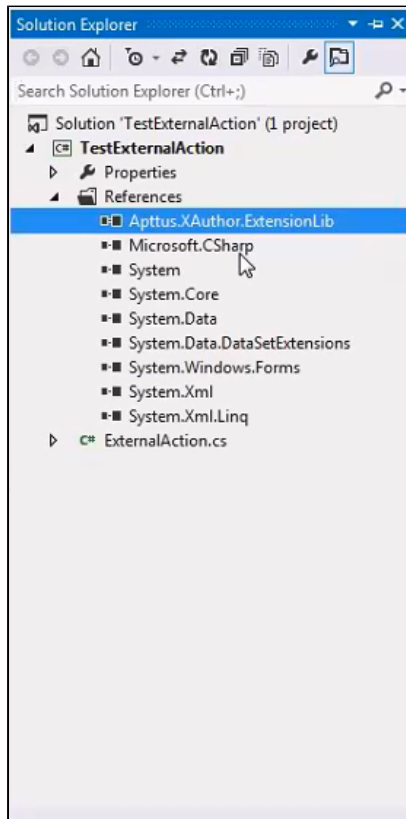
- An Integrated Development Environment that enables you to build DLLs. In the example in this procedure, Microsoft Visual Studio is used to describe the setup process for External Actions.

### How to design an External Action

1. Create a Visual Studio Class Library Project. The X-Author client installer includes a sample C# file.

The code sample shows the functions to use and it will pop up a message box in runtime. You can extend the *Execute()* function to write your custom logic using the full power of Visual Studio and .NET.





2. Add a reference of the Apttus.XAuthor.ExtensionLib module into the project. The Apttus.XAuthor.ExtensionLib action class which represents an action is available for external use in this SDK Library. Any user who designs a new action, needs to add a reference of this library in their Visual Studio project. This library extends the following:
  - Action Class extends the capabilities of the Base class capabilities to the custom .cs file.
  - IActionRuntime interface invokes the execute method defined in a custom .cs file.
3. Provide *Name* and *Type* for the external action and provide custom logic for the *Execute()* function.
4. Compile and link the assembly to generate the output .NET assembly. The .NET assembly should be a .dll file.

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Apttus.XAuthor.Core;
using System.Xml.Serialization;
using System.Runtime.InteropServices;
namespace TestExternalAction

//Apttus.XAuthor.Core.Action is the base class of the module
//for which the capabilities are extended. Using the
//IActionRuntime interface invokes the custom execute method.
{
    public class FirstExternalAction : Apttus.XAuthor.Core.
    Action, IActionRuntime
    {
        //Custom class. You can define any object that you
        //choose in your class.
        public FirstExternalAction()
        {
            Name = "First External Action";
            Type = "External Message Action";
        }
        //Defining the logic on execution of the external
        action.
        public ActionResponse Execute(ActionRequest request)
        {
            ActionResponse r = new ActionResponse();
            System.Windows.Forms.MessageBox.Show("External
Action : Executed");
            r.Status = ActionResultStatus.Success;
            return r;
        }
    }
}

```



The sample \*.cs file can contain one or more external actions.

## How to link an External Library into X-Author

After developing an External Action that is part of a library (.NET assembly), load the library into X-Author Designer using the following steps.

1. Navigate to the X-Author for Designer Tab.
2. Click External Library, load the required .dll file and click OK.



After you load the selected assembly, X-Author detects all the external actions within that assembly.

3. Invoke the newly created external actions within an Action Flow just like built-in actions.

## To update an existing external library

To update and link a new version of the library into X-Author, ensure that any external actions within that library are not in use in any Action Flow. You can update an existing external library only under certain conditions.

## How to remove an existing external library

A user can remove an already linked external library. To remove an existing library, open the existing library and select a row and click **Unlink Library**. An External Library can only be removed if any of the External Actions exported in the library is not used in an Action Flow.

## Creating Look Aheads

Look Aheads allow designers to create field lookups for runtime users. Look Aheads allow users to replace existing records add new records with the values chosen from Search & Select action results or cells from the workbook. For example, a user can look up the account name when they are adding a new opportunity to the app. You can design two types of Look Aheads:

- **Online:** these lookups are real-time Salesforce data and are more flexible than offline. See the procedure for details.


- **Offline:** these lookups are for data that is already available in Excel (it might have been placed there from any source). For example, from a data invoking action (Query or Search and Select), data that was already in the template, or data that is generated from a macro into a list and made available to the offline lookup feature.

### **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 the magnify icon  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**.

**Look Ahead**

Excel **Search And Select Action**

**Configure a Search and Select Data Source**

Search and Select Actions

Account Action

☒ Is Active ☐ Enable Field Mapping

At runtime, the mapped field below will be populated with the record ID from the above Search and Select Action

Mapped Record ID field Account.Account ID

Save Remove Close

8. Click **Close**.

To create an online lookup where new rows are added for each record selected in the lookup

The Enable Field Mapping check box allows you to design a lookup where the data is fetched from the Search and Select Action. The check box also enables you to add new rows and data for each record selected by the user in the record search.

1. Follow the steps to create an online lookup and also click Enable Field Mapping.

**Look Ahead**

Excel Search And Select Action

**Configure a Search and Select Data Source**

Search and Select Actions

Search Agreements

☒ Is Active ☒ Enable Field Mapping

At runtime, the mapped field below will be populated with the record ID from the above Search and Select Action

Mapped Record ID field Account.Opportunity.Agreement.Record ID

Field Mappings: Agreement

	Source Fields		Target Fields
1	Agreement Name	->	Account.Opportunity.Agreement.Agrei ▼
2	Agreement End Date	->	Account.Opportunity.Agreement.Agrei ▼
3	Agreement Start Date	->	Account.Opportunity.Agreement.Agrei ▼

Save Remove Close

- For each field in the Search and Select Action, select a corresponding Target Field from the Display Map.










The Enable Field Mapping check box in a look ahead dialog allows you to design a look ahead where the data is fetched from the Search and Select Action. The check box enables you to add or replace new rows and data for each record selected by the user in the record search.

In the Result Selection of the Search and Select action, 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.). You can set any of the following options:

- Single: Replaces the selected record.
- Multiple: Creates a new record.

## 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 the magnify icon  to open lookup properties for that field. The X-Author Designer window opens.

	Name	Location	Type	
<input type="checkbox"/>	Account.Account ID	Sheet1!\$A\$5	Individual	
<input checked="" type="checkbox"/>	Account.Account Name	Sheet1!\$A\$6	Individual	
<input type="checkbox"/>	Opportunity.Name	Sheet1!\$A\$8	List	
<input type="checkbox"/>	Opportunity.Opportunity ID	Sheet1!\$B\$8	List	
<input type="checkbox"/>	History: Agreement.Created Date	Sheet1!\$A\$15	List	
<input type="checkbox"/>	History: Agreement.Entity History ID	Sheet1!\$B\$15	List	

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



The single-column option must 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**.



**Look Ahead**

**Excel** Search And Select Action

**Configure an Excel Data Source**

Mapped Field : Account.Account Name

**Column Mapping**

☒ Single Column ☐ Multi Column

Selected Excel Columns

	Primary Return Column	
	Secondary Return Column	
	Target Field	

Current Selected Range :

Highlight the Excel cell range and click on Get Current Selection


New Range : Sheet1!\$A\$5:\$A\$6

☒ Refresh data on every load ☒ Is Active

Get Current Selection

**Save** Remove Close

## 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 the magnify icon  to open lookup properties for that field.

Mapped Details			
	Name	Location	Type
<input type="checkbox"/>	Account.Account ID	Sheet1!\$A\$5	Individual
<input checked="" type="checkbox"/>	Account.Account Name	Sheet1!\$A\$6	Individual
<input type="checkbox"/>	Opportunity.Name	Sheet1!\$A\$8	List
<input type="checkbox"/>	Opportunity.Opportunity ID	Sheet1!\$B\$8	List
<input type="checkbox"/>	History: Agreement.Created Date	Sheet1!\$A\$15	List
<input type="checkbox"/>	History: Agreement Entity History ID	Sheet1!\$B\$15	List

- Click the **Excel** tab and, under the Column Mapping section, select **Multi-Column**.

### **Note**

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

- 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 must always include the record ID of the lookup value (for example, Account ID). The column identifiers appear in the display columns.

- Select the **Primary Return Column**. This is the column that matches the lookup value (for example, Account Name).
- Select the **Secondary Return Column**. This must always be the column that has the record ID of the lookup value.
- Select the **Target Field**. This must be the record ID field for the lookup name.
- 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.
- 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, refresh the lookup fields every time you load the lookup by clicking the **Refresh on Every Data Load** check box.

## Mapping Records into Columns and Rows

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	

Term	Definition
	A data mapping is created when you drag a field to the worksheet from a List object. It must be created <b>after</b> 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.
- 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.

## Matrix Map 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	Actual	Variance	Forecast	Actual
	Jan	Jan	Jan	Feb	Feb
<b>Revenue</b>	5,000,000	5,500,000	500,000	6,000,000	6,200,000
<b>Cost of Revenue</b>	3,000,000	3,100,000	100,000	3,600,000	3,655,000
<b>Gross Margin</b>	2,000,000	2,400,000	400,000	2,400,000	2,545,000
<b>% of Revenue</b>	40.0%	43.6%	3.6%	40.0%	41.0%
			0		
<b>R&amp;D Expense</b>	400,200	485,000	84,800	549,800	539,000
<b>% Revenue</b>	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

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.

- [Creating Sections](#)
- [Extending Sections](#)
- [Grouping Data in Columns](#)

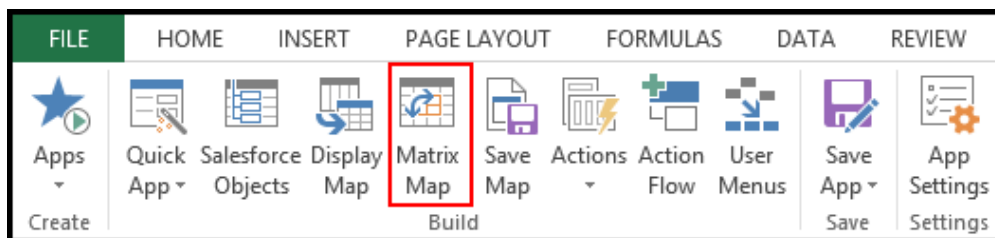
## Creating Sections

This procedure enables you to create a section

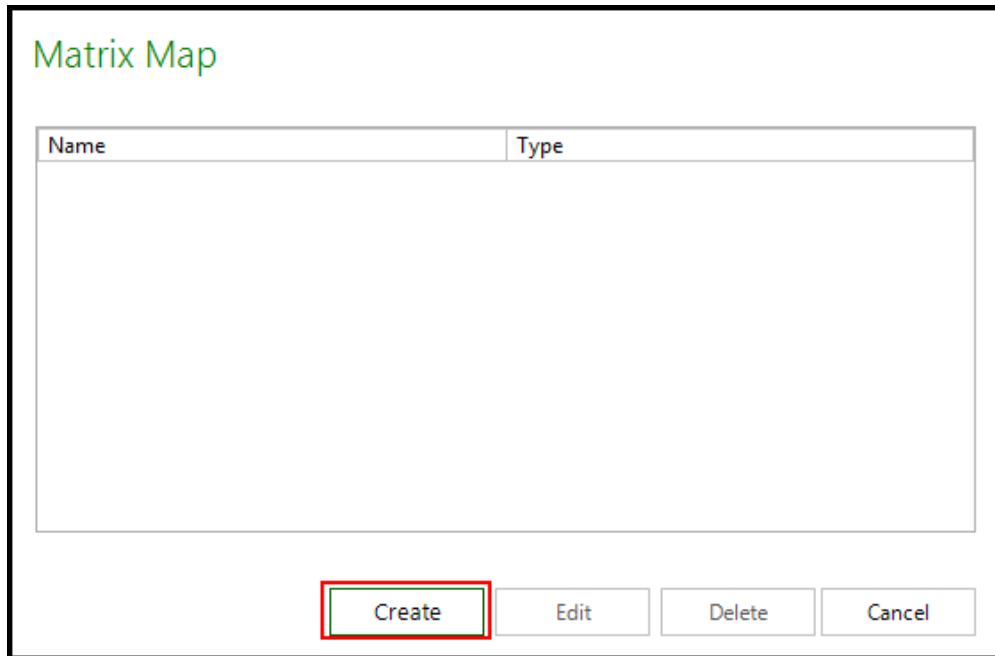
### To create sections

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

1. From the X-Author Designer ribbon, click **Matrix Map**.



2. Click **Create**.



The image shows a dialog box titled "Matrix Map" in green text. Inside the dialog, there is a table with two columns: "Name" and "Type". The table is currently empty. Below the table, there are four buttons: "Create", "Edit", "Delete", and "Cancel". The "Create" button is highlighted with a red rectangular border.

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

3. Type a Name for the Matrix Map.

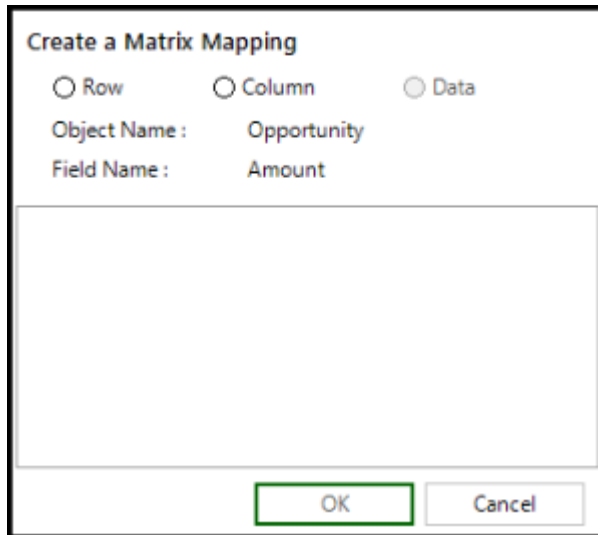
If you are creating one Matrix Map, you can use *Map* or any other name. If you are using multiple Matrix Maps, then you need a more descriptive name to identify each one.

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.

4. Create a Row and Column Mapping for each section.

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





The dialog box is titled "Create a Matrix Mapping". It contains three radio buttons: "Row", "Column", and "Data". Below these, there are two labels: "Object Name:" followed by "Opportunity" and "Field Name:" followed by "Amount". At the bottom, there are "OK" and "Cancel" buttons. The "OK" button is highlighted with a green border.

- b. Select either 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. Click **OK** to complete the settings for that field.

Create a Matrix Mapping

☒ Row    ☐ Column    ☐ Data

Object Name :    Opportunity

Field Name :    Amount

Rendering Type	Static
Value Type	Field Label
Sort field	

OK    Cancel

5. 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. Select a Row Lookup.
  - d. Select a Column Lookup.
  - e. Click **OK**.
6. Click **Save and Close**.
7. Click **Save App**.

## 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.

## Grouping Data in Columns

You can use matrix grouping features to display data in grouped columns within a matrix.

If you want to group pricing data for a region, for a specific devicetype for a particular processor, configure the following

1. Set the **Device Type** field as a regular Dynamic Matrix Column.
2. Set the **Region** field as a Grouped Column on which the data is grouped.
3. Add **Processor Name** as Matrix Row.
4. Add **Price as a Data Field** to the Matrix.
5. You can add one or more grouped columns on top of the Region Grouped Column field.

	Region (Grouped Column)
	Device Type (Column)
Processor Name (Row)	Price (Data Field)

In this use case all the fields belong to the same object. While grouping data in columns ensure that the fields belong to the same object or are associated with each other via a lookup.

For every Individual Matrix Column of Rendering Type – Dynamic, one or multiple grouped columns may exist.

If the same record exists across multiple objects, you can group the data in the cells of a matrix based on the objects the data exists in. For Matrix grouping to work, the data has to be structured in such a way where the data for a particular field is common. Not every List can be transformed into a matrix grouping use-case.

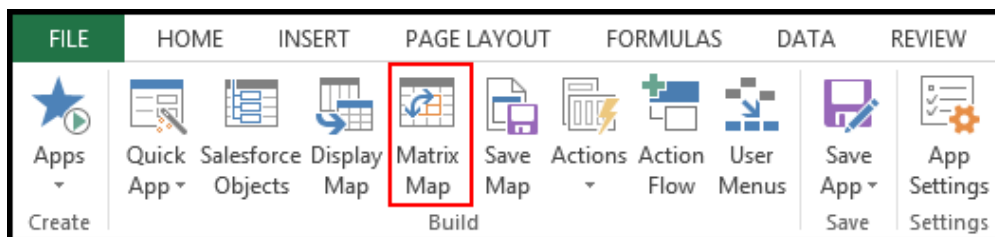
Matrix Grouping is applicable only on columns with Rendering Type as Dynamic. For matrix grouping to work, there must be a valid Dynamic matrix column. A user can drag a matrix field on top of an already created dynamic Matrix Column. X-Author for Excel automatically detects whether a dragged matrix field is a matrix grouped field or not. If it is a grouped field, the Grouped Field property is set to true automatically and within the elements of the matrix, the column categorized as a Grouped Column.

**Create a Matrix Mapping**  
☐ Row ☒ Column ☐ Data  
Object Name : ProcessorPricing  
Field Name : Region  

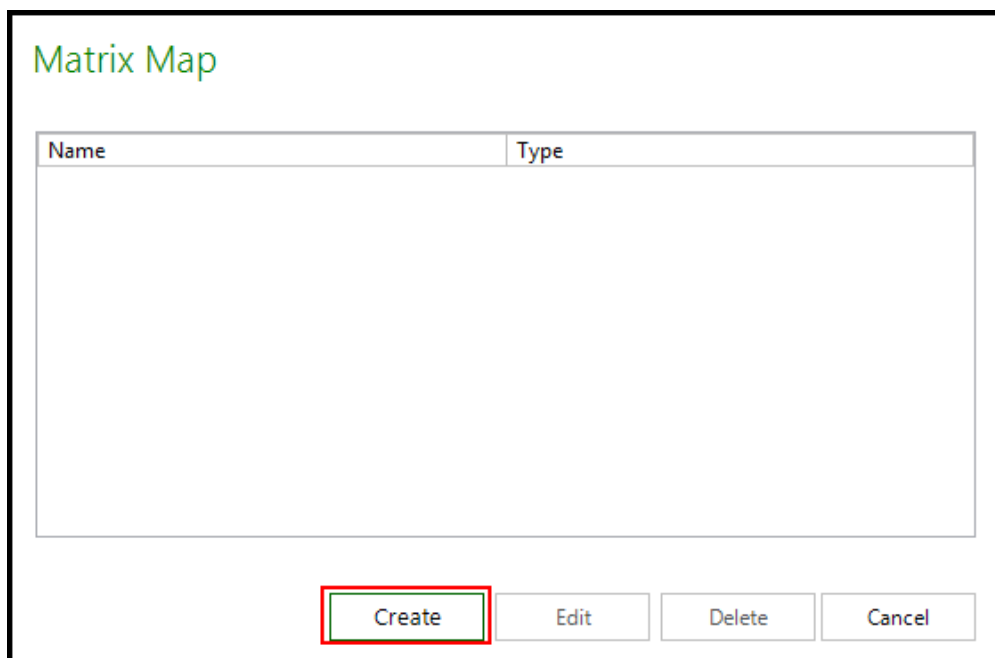
Rendering Type	Dynamic
Value Type	Field Value
Sort field	
Grouped Field	True

## To create Grouped Matrices

1. Click **Matrix Map**.



2. Click **Create**.



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

3. 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 (**List**). List fields are displayed by default.

4. In an existing Matrix Map if you want to add a grouped column, create a Column Mapping for each section.

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

**Create a Matrix Mapping**

☐ Row ☒ Column ☐ Data

Object Name : ProcessorPricing

Field Name : Region

Rendering Type	Dynamic
Value Type	Field Value
Sort field	
Grouped Field	True

**OK** **Cancel**

- b. Click the Column radio button. The Rendering Type, Value Type, Grouped Field and Sort options are displayed. All of these work together to determine how column values are grouped.



Ensure that the Grouped Field column is True. If this field is set it to false, the column added to the Matrix is not a part of other objects.

- c. Select a Rendering Type.
    - Dynamic: Field data is retrieved automatically based on the data getter actions.
  - d. Click **OK** to complete the settings for that field.
5. 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. After 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. Select a Row Lookup.
  - d. Select a Column Lookup. Click OK to complete the settings for that field.
6. Click **Save and Close**.
  7. Click **Save App**.

## Behavior of Grouped Matrices in Runtime

At runtime, the end-user doesn't need to do anything. Based on the matrix grouping design, the data will be filtered from top most Grouped Column to last regular Matrix Column and finally the data field is rendered.

## Saving Matrix Grouping

If a Matrix contains Columns that are grouped, only the update operation can be performed. The Insert operation cannot be performed.

## Matrix Limitations

1. Currently grouping based on top of matrix columns is supported while grouping for matrix rows is not supported.
2. If the data is grouped, you cannot add a row or a column to the matrix.

## 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 existing Section.

## To add a section filter

1. Click **Matrix Map**.
2. Select the desired Matrix Map and click **Edit**.
3. Click the Sections tab.
4. Click the lookup 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 App**.



## 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.

## Uploading or Downloading File Attachments

You can 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. 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.

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	006i000000LoHFKA3	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.

# 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 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.

To implement this feature, perform the following steps in order.

1. Adjust your App slightly so that it can be launched 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.

You can launch a specific Action Flow of a specific app from Salesforce. To do so, you must mention the names of the app and action flow in the code of the button that you add. When an action flow is mentioned in the launch button code, it gets executed when launched from Salesforce. When an action flow is marked as Auto Execute (Launch from Excel) and you mention another action flow in the launch button code, the latter action flow gets executed when launched from Salesforce.

## Making an App Launch-Ready from Salesforce

There are only two differences in an app launched from Salesforce versus 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 X-Author for Excel ribbon.
2. Select an Action Flow for auto-execution by checking the box in the Auto-Execute (Launch from Salesforce) 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, 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.

### To add an External ID to a query

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.

	Salesforce Object and Field		Value Type	Operator	Value	
1	Account ID		System Variab	equals	ExportRecordId	

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.



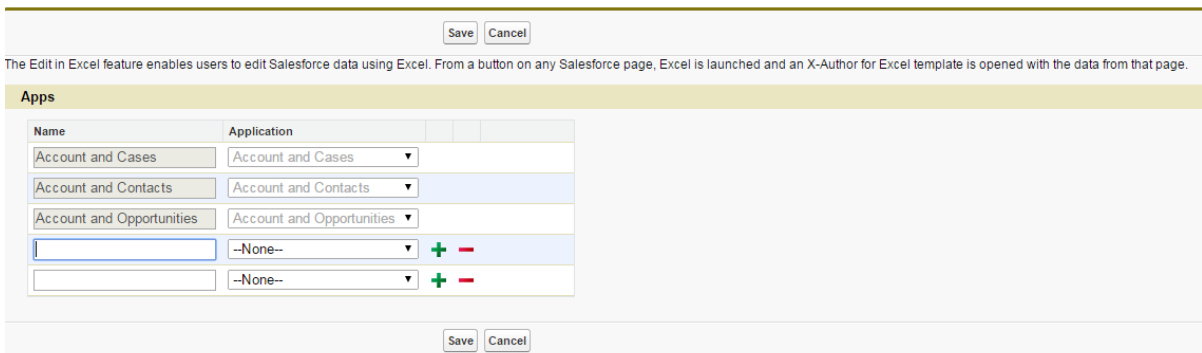
2. Click on Edit in Excel Setup.

### Note

If the **Edit in Excel** button is not visible, you need to add it to your App page layout. Refer to [Completing Setup for Designers](#) for instructions.

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.

### Setup Edit In Excel Applications



Name	Application
Account and Cases	Account and Cases
Account and Contacts	Account and Contacts
Account and Opportunities	Account and Opportunities
	--None--
	--None--

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 drop-down, select the Application 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.

### To create a button on an object

You can create this button on standard as well as custom objects. This example shows how to create the Edit In Excel button on Opportunity related list on Account object.

1. Go to Setup > App Setup > Customize > Opportunities and select Buttons, Links and Actions.
2. Click New Button or Link.
3. Enter the Label. Name is auto-populated.
4. Enter the Description of the button to help users understand the button.
5. Select a Display Type for the button.

Display Type	Description
Detail Link Page	To display the custom button as a link
Detail Page Button	To display the custom button as a detail button. For example, to display the custom button with Edit, Delete, Cancel, and Clone buttons.
List Button	To display the custom button as a list button. For example, to display the custom button with Merge Contacts, New, Add to Categories and so on.

6. Select a Behavior. Click View Behavior Options to view the options of behavior.
7. Select Custom Source.
8. If you are executing a specific Action Flow, then paste the following code in the Formula Editor:

```
/apex/Apttus_XApps__EditInExcelLaunch?parentRecordId={!Account.  
Id}&"&appName=Account and  
Opportunities&actionflowName=retrieveOpp"
```

### Important

Click Check Syntax to verify the syntax of formula you have written. You can choose to configure an App with a named action flow by passing actionflowName as a parameter and the name of the actionflow that you want to call as a value of the parameter. This feature is supported only on the X-Author for Excel 4.72 Salesforce package only

9. If you are using an auto-execute Action Flow, enter the following code into the Formula Editor:

```
/apex/Apttus_XApps__EditInExcelLaunch?parentRecordId={!Account.  
Id}&"&appName=3_8 AC_Opps ASC"
```




10. Select the Link Encoding. Default value is Unicode (UTF-8).
11. Click Save.

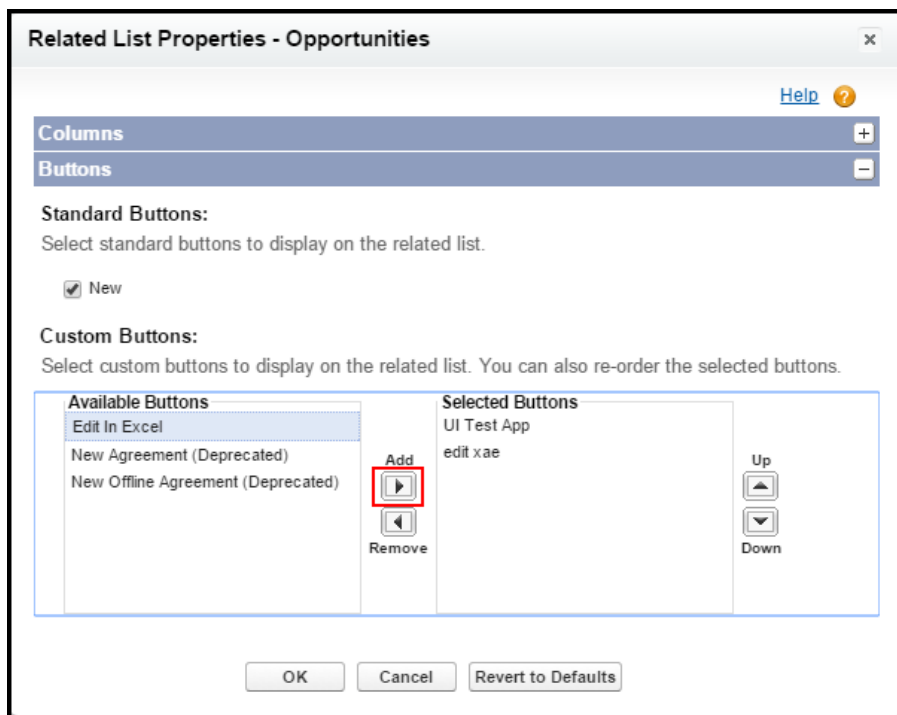
## To add a button in a Salesforce page

You can add this button on standard as well as custom objects. For example, add the **Edit In Excel** button on Opportunity related list on Account object.

1. Click the Accounts tab and select an account.



2. Click Edit Layout.
3. Click Related Lists from the left pane of the Edit Layout box.
4. For an Opportunities-related list, click the wrench  icon.
5. For Buttons, click the plus  icon.
6. Select Edit In Excel and click the right arrow  icon.



7. Click Ok.

If your Launch from Salesforce feature is not working as expected, you might not have set the correct user permissions to access.

# X-Author for Excel Runtime Functions

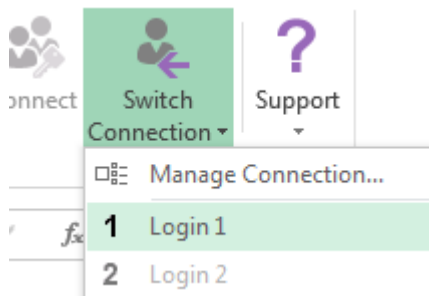
The following runtime functions are available for X-Author for Excel apps.

- [Switch Connection](#)
- [Opening Apps](#)
- [Adding Rows or Records to a List](#)
- [Adding Rows and Columns to a Matrix](#)
- [Paste from Mapping](#)
- [Deleting Records from Salesforce](#)
- [Adding and Deleting Records](#)
- [Chatter](#)
- [Rich Text](#)

## Switch Connection

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

1. Click **Connect** to display the Manage Connections window.
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, type the custom domain name.
4. Click **Authorize**.
5. Open an App.

## Opening Apps

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

### To open an app

1. Click **Apps** and then click Open Application to view the available Apps to open.

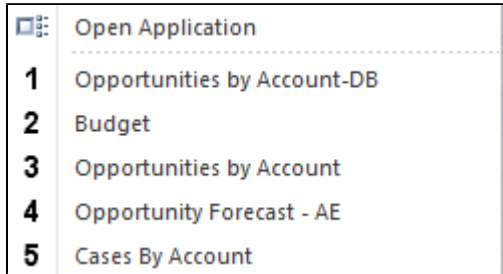


2. Either scroll down to find an App or enter in the App name in Search by Application Name and click **Search**. Then either double-click on the App or click **Select**.

Search by Application Name : <input type="text"/>			<input type="button" value="Search"/>
<input type="checkbox"/> My Apps			
Application Name	Owner	Modified On	
Sony Test 2	Anton Weber	05/28/2014 06:30	
Sony Test	Anton Weber	05/28/2014 06:02	
Opportunities by Account-DB	Anton Weber	05/27/2014 17:52	
Budget	Anton Weber	05/27/2014 11:45	
Opportunities by Account	Anton Weber	05/27/2014 11:45	
Opportunity Forecast - AE	Anton Weber	05/27/2014 11:45	
Cases By Account	Anton Weber	05/27/2014 10:48	
Cases	Anton Weber	05/27/2014 09:58	

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

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



After the App loads into Excel, you will see custom menus created specifically for the chosen app, such as 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 in one of two ways: either by using the Add Row button or by using one of the Paste functions from the Paste drop-down 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.



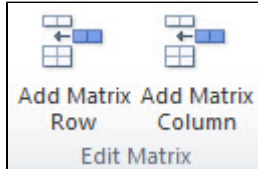
There are three variations of the Paste function: Paste Values Only, Paste All, and Paste from Mapping.

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 must 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, then 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

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

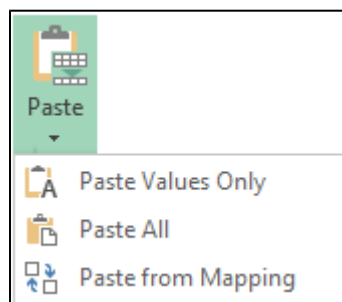
Matrix Configuration		Add Row?	Add Column?
	Column: Static - Field Value	YES	
Row: Dynamic - Field Value	#####Data#####		
	Column: Dynamic - Field Value	YES	YES
Row: Dynamic - Field Value	#####Data#####		

## Paste from Mapping

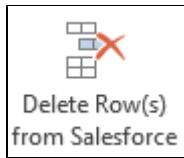
This function is used during data migration between Salesforce orgs and is specifically for copying data between source and target workbooks. Refer to [Migrating Data Between Salesforce Orgs](#) for more details.

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**.

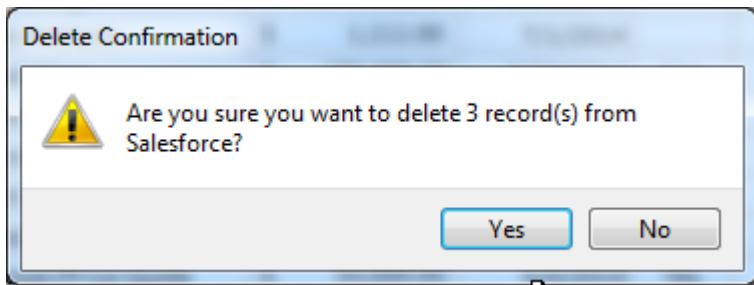
Data is transferred from the source file to the Target App.



## 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. After 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.

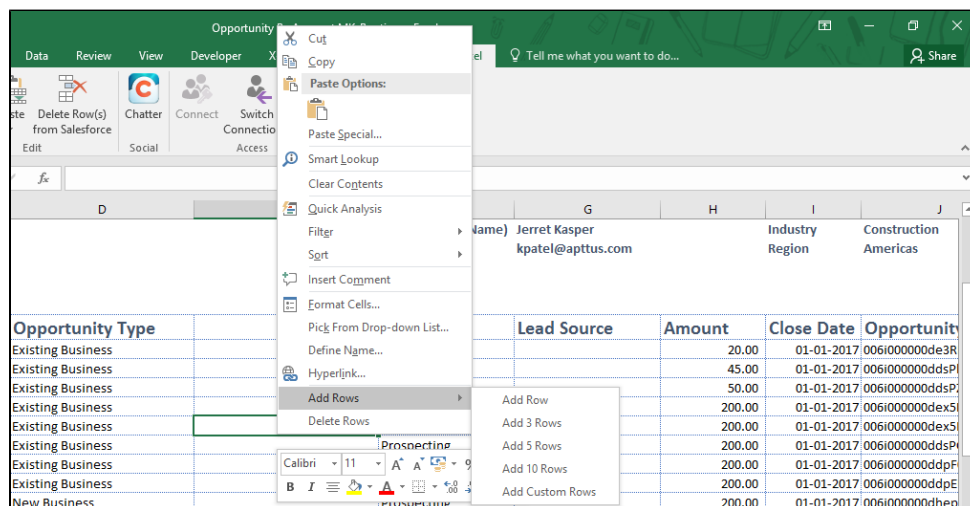


## Adding and Deleting Records

Whenever a user right-clicks a cell, if the cell is a part of a Repeating range or is of type "Save Field Only," then the X-Author add row menu and remove row menu is displayed in the context menu. The native Insert or Delete function disappears. The Add or Remove menu is enabled.

The menu functionally behaves the same as Current Add Row menu.





## Limitations of the Add Rows and Delete Rows functions

The user cannot access shortcut keys (Ctrl + and Ctrl -) to insert or delete rows.

## Chatter



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.

## Configuring Presto Apps

The procedure below outlines the custom settings required to configure Presto Apps.

1. Go to **Setup > Develop > Custom Setting**. Look for DocGen System Properties. Click **Manage**.
2. Under this setting one entry must be created - If there are no records then click **New**. After clicking New, fill in the following details:

<b>Name</b>	System Properties
<b>Merge Call Timeout Millis</b>	60,000
<b>Merge Webservice Endpoint</b>	<a href="https://mergewebs.apttus.net/cgi-bin/MergeServer/Bin/MMCGI.exe">https://mergewebs.apttus.net/cgi-bin/MergeServer/Bin/MMCGI.exe</a> (Production)

3. Go to **Setup > Develop > Custom Setting**. Look for X-Author For Excel System Properties. Click **Manage**.
4. Under this setting one entry needs to be created - If there are no records then click on **New**. Fill in the details as shown in the table below:

<b>Name</b>	System Properties
<b>Instance Url</b>	<Instance Url for your Salesforce Org> For example: <a href="https://na16.salesforce.com">https://na16.salesforce.com</a>



*If you see more fields under System Properties, keep them unchanged.*

5. Go to **Setup > Security Controls > Remote Site Settings**.

- Under this setting one entry needs to be created - if it doesn't exist. Below are the details:

<b>Remote Site Name:</b>	mergews
<b>Remote Site URL:</b>	<a href="https://mergews.apttus.net">https://mergews.apttus.net</a>
<b>Disable Protocol Security</b>	Unchecked
<b>Active</b>	Yes

## Configuring a Presto App

- Create a new detail page button on your target object and configure the button setting as shown in the illustration below:

Text for a Button or Link URL must be set. Copy the code below into the text area.

```
/apex/Apttus_XApps__EditInExcelLaunch?selectedRecordId={!
Account.Id}
```

```
&"&appName=Quote Calculator&mode=touchless&outputType=None"
```




You can also set the information on a button, link or VF page by using the following API in a class: `Apttus_XApps.EditInExcelLaunchController` `editLaunch = new Apttus_XApps.EditInExcelLaunchController('TestPresto2', recId,'XLSX');` where the first parameter is the App name, second one is the record ID, and third one is the output format. This API returns a boolean value. On successful execution, the API returns true otherwise false.

To access the package for Presto as an API contact the Apttus Customer Excellence by raising a case with them or Apttus Release Management Team.

2. Replace the following section of the code:

- **selectedRecordId** is the record id of the object that will trigger X-Author for Excel. For example if a button is created on the Account page, you will set the value as `Account.Id`.
- **appName** is the name specified in Edit in Excel setup. This is NOT the actual name of the X-Author for Excel app.

For example, as seen in the picture below, the *appName will be Quote Calculator as specified in Edit in Excel setup* and NOT the actual name of X-Author for Excel app, which in this case is SFDC - Quote Calculator.



## Setup Edit In Excel Applications

The Edit in Excel feature enables users to edit Salesforce data using Excel. From a button on any Salesforce page, is opened with the data from that page.

### Apps

Name	Application		
Account and Cases	Account and Cases	▼	
Account and Contacts	Account and Contacts	▼	
Account and Opportunities	Account and Opportunities	▼	
Account and Agreements	SFDC - Account Agreements	▼	—
Quote Calculator	SFDC - Quote Calculator	▼	+ —

- **outputType** can either be None, EXCEL or PDF. When it is set as Excel or PDF, Presto will attach the filled app template as an attachment to the selected record. The filename will be Template.xlsx or Template.pdf when output type is Excel or PDF respectively.

# Apttus Copyright Disclaimer

Copyright © 2017 Apttus Corporation (“Apttus”) and/or its affiliates. All rights reserved.

No part of this document, or any information linked to or referenced herein, may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written consent of Apttus. All information contained herein is subject to change without notice and is not warranted to be error free.

This document may describe certain features and functionality of software that Apttus makes available for use under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not, in any form, or by any means, use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part of the software. Reverse engineering, disassembly, decompilation of, or the creation of derivative work(s) from, the software is strictly prohibited. Additionally, this document may contain descriptions of software modules that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document.

**U.S. GOVERNMENT END USERS:** Apttus software, including any operating system(s), integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Neither the software nor the documentation were developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Apttus and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Apttus and X-Author are registered trademarks of Apttus and/or its affiliates.

The documentation and/or software may provide links to Web sites and access to content, products, and services from third parties. Apttus is not responsible for the availability of, or any content provided by third parties. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Apttus is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Apttus is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

For additional resources and support, please visit <https://apptusportal.force.com/community>.

Copyright ©2017 Apttus. All Rights Reserved.