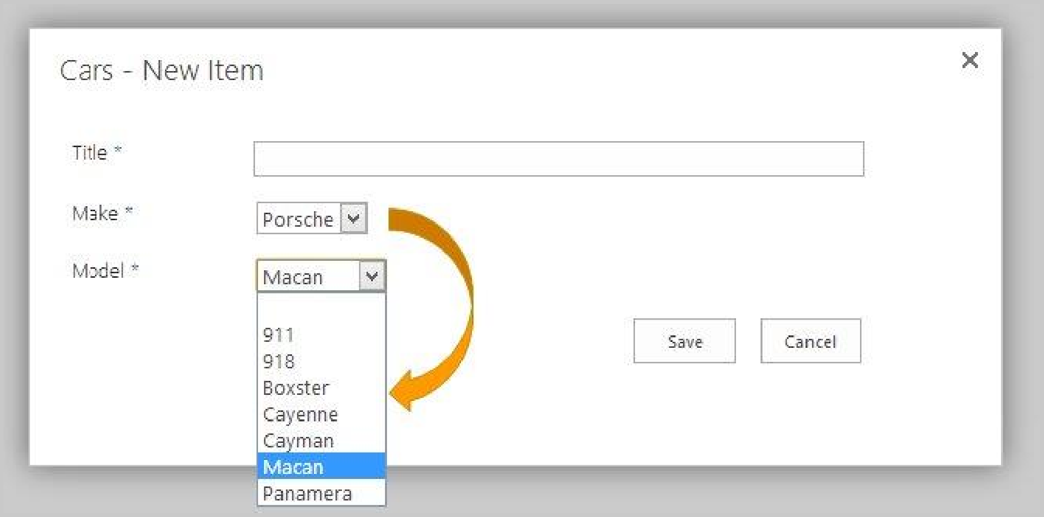
SPELL Cascading Selects

Evaluation Version



Cars - New Item

Title *

Make * Porsche

Model * Macan

SPELL Cascading Select 1.1.0

Evaluation Version

November 2013

Author:

Christophe HUMBERT

User Managed Solutions LLC

Table of Contents

About Cascading Selects

In a form, the term **cascading selects** refers to a hierarchical relationship between two select fields. Once an option is selected in the parent field, the child field displays a list of options based on the parent scope.

Some examples:

- Car: Make and Model
- Project Management: Client and Project
- Maintenance: Site and Asset
- Address: State and City
- Etc.

This document describes how cascading selects are implemented in SPELL, and provides instructions for the evaluation version. SPELL Cascading Selects work in SharePoint 2007, 2010, 2013 and the current version of Office 365 (SharePoint Online).

SPELL is a program dedicated to the deployment of User Managed Solutions in SharePoint environments.

An Example: Project Management

Explaining how cascading selects work in the context of SharePoint will be easier with an example.

Let’s consider the following scenario: we want to track various activities related to external organizations – they could be customers, vendors, partners, etc. For each organization, the work is structured in projects, each project being divided into Tasks.

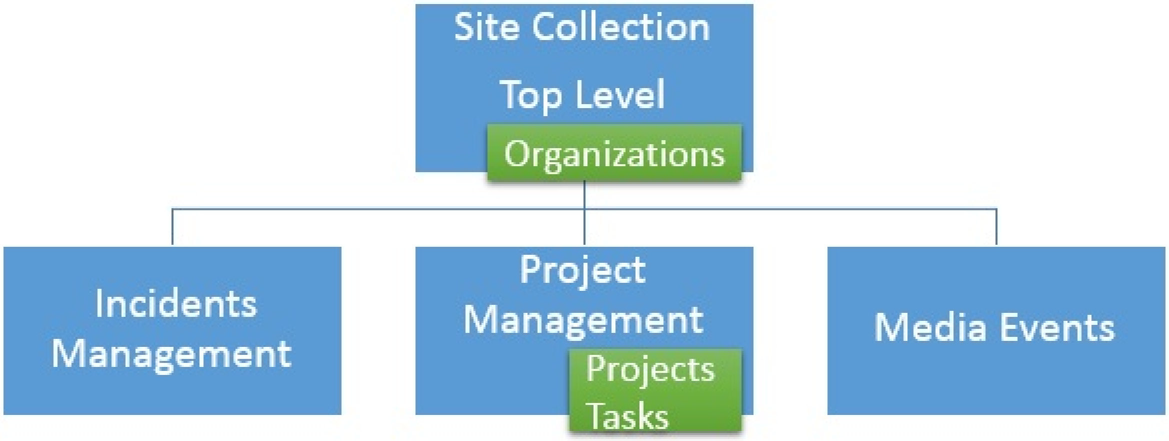


Figure 1: site collection (top level and sub-sites)

The Information Architecture for this scenario is pretty straightforward: we are naturally going to create 3 lists, called Organizations, Projects and Tasks, and bind them via lookup columns.

The **Projects** and **Tasks** reside in a sub-site called Project Management (figure 1). As for the **Organizations** list, I placed it at the top level because I plan to reuse it in other sub-sites (this choice doesn't affect the cascading selects solution).

Note: *This architecture assumes that everybody in the team has contributor access to all tasks. Things might be different if we needed more granular permissions.*

Organizations and **Projects** are based on a "Custom List" template, while the **Tasks** list is based on the "Tasks" template.

The columns are set as follows (figure 2):

- Each list includes the mandatory **Title** (renamed to clarify its purpose).
- **Organization** and **Organization Name** are lookup columns that point to the **Name** field in the Organizations list.
- **Project Codename** is a lookup that points to the **Codename** field in Projects.

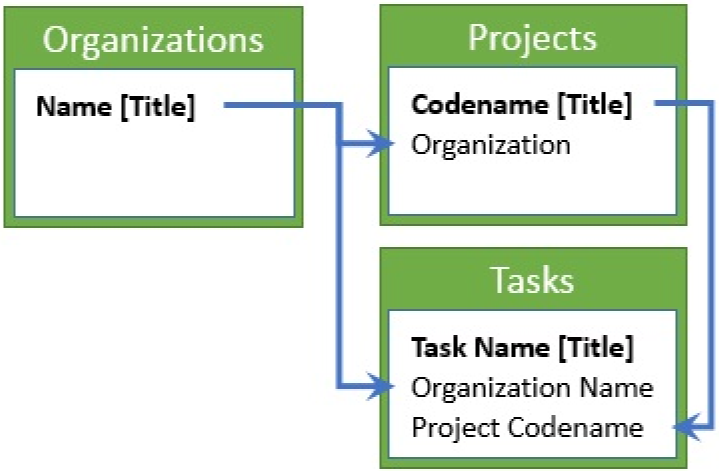
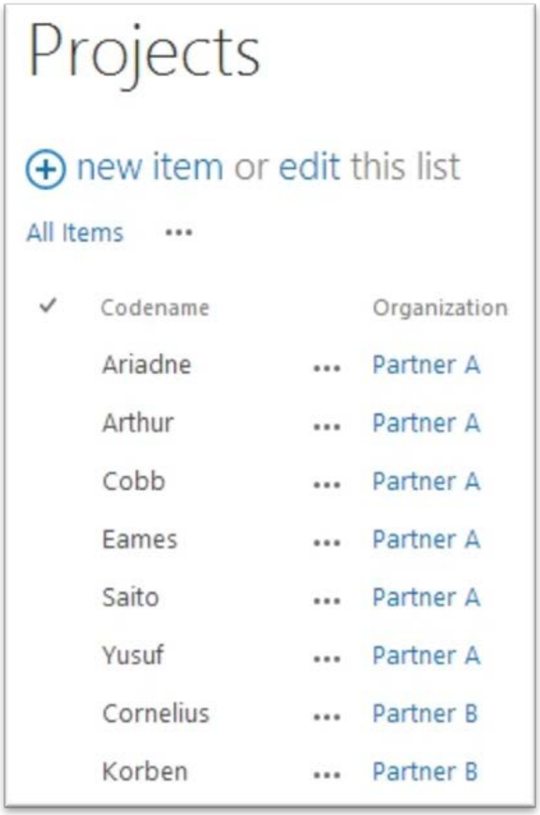


Figure 2: lookup columns

The Projects list is the cornerstone of the cascading implementation, as it establishes the relationship between the project and the organization (figure 3). In our solution, we'll refer to this list as the "**source**".



Projects

⊕ new item or edit this list

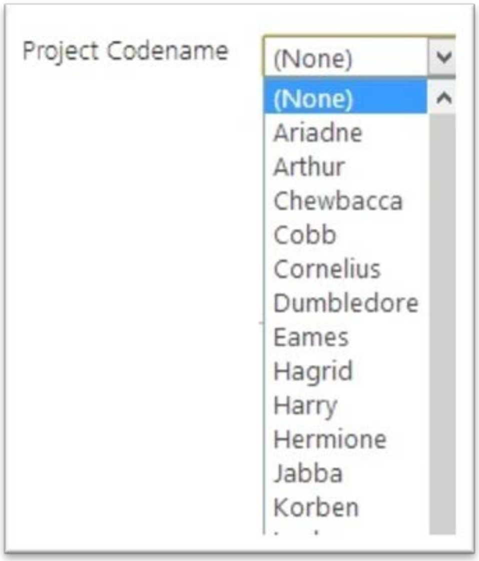
All Items •••

✓	Codename		Organization
	Ariadne	•••	Partner A
	Arthur	•••	Partner A
	Cobb	•••	Partner A
	Eames	•••	Partner A
	Saito	•••	Partner A
	Yusuf	•••	Partner A
	Cornelius	•••	Partner B
	Korben	•••	Partner B

Figure 3: project-organization relationship

At this point, all is good: team members can start adding tasks, and associate them with external partners and projects.

However, after some time, as the list of projects keeps growing, it becomes tedious to fill out the Tasks new form. The codenames are all mixed up, and users have to scroll down to find the project they are working on (figure 4).

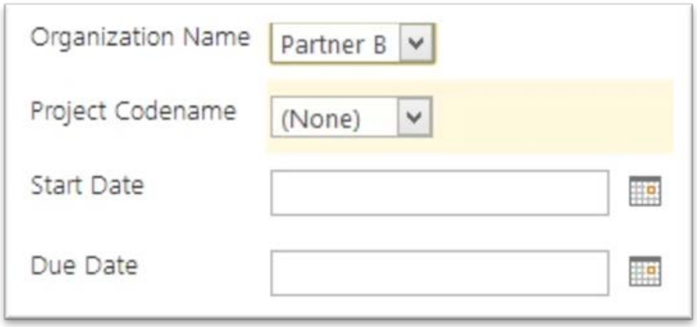


Project Codename: (None)

Figure 4: mixed codename list

After a user selects an organization in the form, it would be nice if the form could help him/her filter out irrelevant projects. That's what the SPELL Cascading Selects solution is here for!

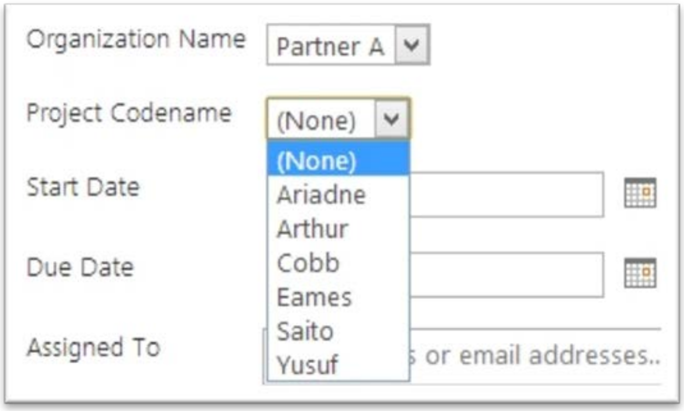
After the Cascading Selects solution is added to the form, the Project Codename options will be filtered based on the chosen organization. To tip you about the update, SPELL will paint the select background in light salmon (while a light red background would indicate that an issue has occurred).



Organization Name	Partner B
Project Codename	(None)
Start Date	
Due Date	

Figure 5: background color after programmatic change

The screenshots below show how the Project Codename field renders based on the selection in the Organization Name field.



Organization Name: Partner A

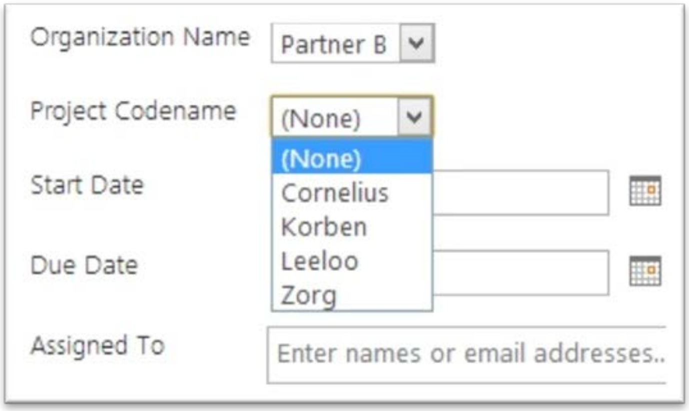
Project Codename: (None)

Start Date

Due Date

Assigned To

Figure 5a: codenames filtered for Partner A



Organization Name: Partner B

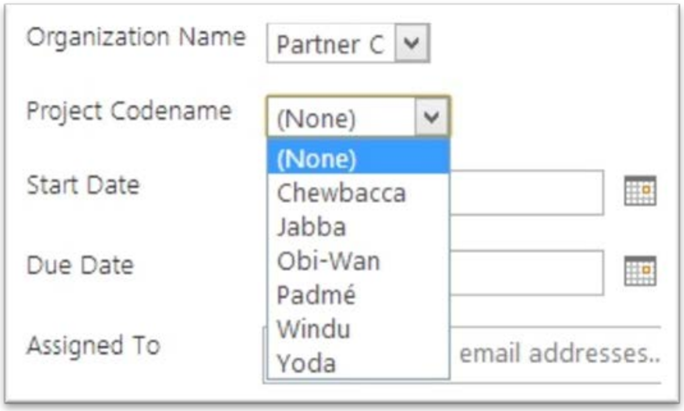
Project Codename: (None)

Start Date

Due Date

Assigned To: Enter names or email addresses..

Figure 5b: codenames filtered for Partner B



Organization Name: Partner C

Project Codename: (None)

Start Date

Due Date

Assigned To

Figure 5c: codenames filtered for Partner C

Deployment Instructions

Prerequisites

To implement the cascading selects, you first need to build an Information Architecture similar to the one presented in the Project Management example. Remember the key role of the source, which stores the relationship between the two selects.

In this section we are going to continue working with our Project Management example:

- "**form**" refers to the form where the cascading behavior is implemented. In our case it is the Tasks new form or edit form. The select fields are labeled "Organization Name" and "Project Codename".
- "**source**" refers to the list that establishes the relationship, in our case the Projects list. Within this list, the first column is called Codename (this doesn't matter for our cascading setup), and the second column is called "Organization".

Browser requirements

For modern browsers, all the required code is provided in a single page called **CascadingSelects1.1.0Evaluation.aspx**.

For older browsers (like IE 7), additional code is required to supplement missing capabilities. The evaluation version will try and retrieve it from a public Content Delivery Network (CDN).

SPELL doesn't support Internet Explorer version 6 or before.

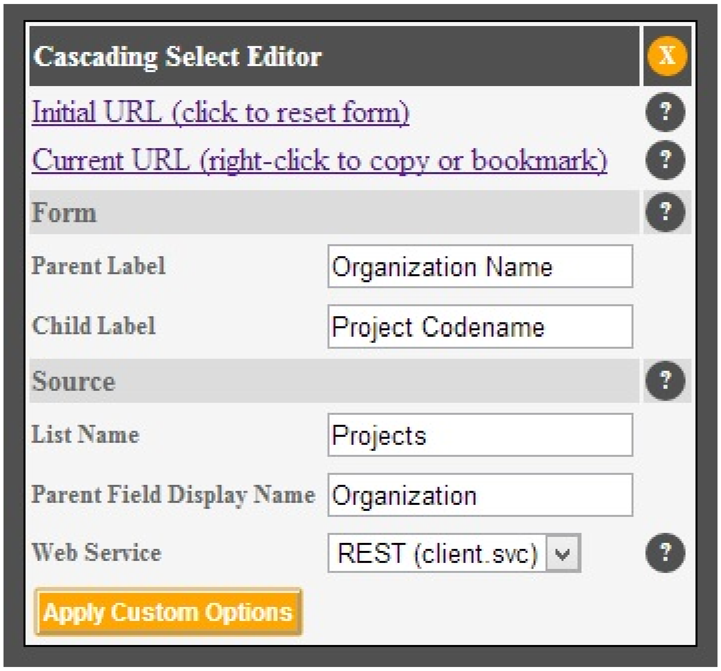
Installation

The installation process is extremely simple: just upload the **CascadingSelects1.1.0Evaluation.aspx** file to a document library and you're good to go! You can for example use a Site Assets library (present by default in all SharePoint team sites). In our example, we'll use the Site Assets library of the top level site, next to the Organizations list.

***Remember**: the same SPELL file can be used across sites and even across site collections, within a same domain. If for example your file is located in http://divisionOne.myCompany.com/SiteAssets/SPELL/ , you can use it on any page in http://divisionOne.myCompany.com, but not on pages located in http://divisionTwo.myCompany.com.*

Settings

Go to the library where you stored **CascadingSelects1.1.0Evaluation.aspx**, and click the file name. The standalone page opens and automatically turns to editor mode (figure 6).



Cascading Select Editor
X
Initial URL (click to reset form)
?
Current URL (right-click to copy or bookmark)
?
Form
?
Source
?
?
Apply Custom Options

Parent Label: Organization Name

Child Label: Project Codename

List Name: Projects

Parent Field Display Name: Organization

Web Service: REST (client.svc)

Figure 6: Cascading Selects – Edit mode

Fill out the form, and then click on the “Apply Custom Options” button. Hover over the (?) signs for contextual help.

***Important!** Your options are not saved; they are just added to the URL. Remember to keep a copy of the current URL when your options are final.*

After clicking the button, the form URL looks like this:

/SiteAssets/CascadingSelect1.1.0Evaluation.aspx#form.parentLabel=**Organization%20Name**; form.childLabel=**Project%20Codename**;source.listName=**Projects**; source.parentField.DisplayName=**Organization**;api=**client**

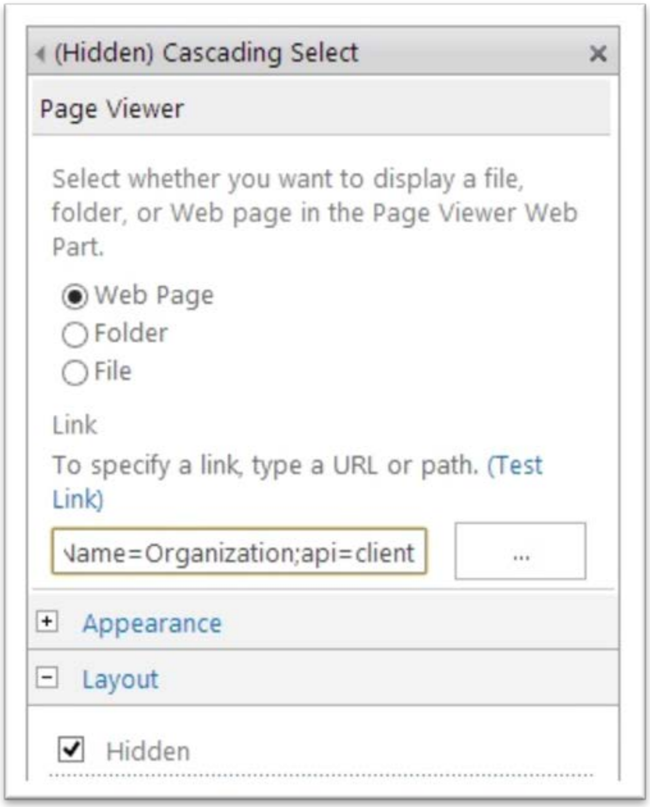
***Remember**: by passing the options in the hash part of the URL, we are ensuring that **the exact same page (CascadingSelects1.0.0Evaluation.aspx)** is leveraged across sites, and even across site collections within a same domain.*

Adding Cascading Selects to a form

To enable cascading selects on a new or edit form (for Tasks in our example):

1. Go to your SharePoint form and switch to edit mode (Site Actions > Edit page).
2. Add a **Page Viewer Web Part**, and link it to the URL you saved in the previous step (figure 7).
3. Set the Web Part to Hidden.
4. Save the page and exit the edit mode.

The next time you open the form, you should see the child select with a light salmon background, proof that the customization has been applied.



(Hidden) Cascading Select ×

Page Viewer

Select whether you want to display a file, folder, or Web page in the Page Viewer Web Part.

● Web Page
○ Folder
○ File

Link

To specify a link, type a URL or path. (Test Link)

Name=Organization;api=client ...

⊞ Appearance

⊟ Layout

☑ Hidden

Figure 7: Page Viewer Web Part settings

To learn more about code insertion via a Page Viewer Web Part, see this blog post.

Programmatic approach

For the record, in the full version of SPELL Cascading Selects, the options can also be entered directly in code.

This is what the code would look like:

```
SPELL.form.cascadeSelect({
        form:{
                parentLabel: “Organization Name”,
                childLabel: “Project Codename”
        },
        source:{
                listName: “Projects”,
                parentField:{DisplayName: “Organization”}
        },
        api: “client”
});
```

Can I do more with the SPELL Cascading Selects?

Definitely!

The user interface provided with the evaluation version gives you access to the standard options, for one cascading relationship. With a custom interface, or if you switch to code, you’ll be able to set up multiple cascading behaviors on a form, add filtering rules, and/or tweak the look.

Beyond cascading, this solution is part of **SPELL Form**, a module that allows various form customizations:

- Hide fields
- Render fields as read only
- Pre-populate fields
- Add a tabbed interface
- Etc.

You can find an example of tabbed interface in this blog post.

To get in touch with us and learn more about SPELL, see the contact information at the end of this document.

Known issues and limitations

Multi-select

In SharePoint, a lookup column can be set up as a single select, or a multi-select by checking the box "Allow multiple values".

For Cascading Selects, the parent has to be a single select. The child can be either single or multi-select.

Cascading Selects: what are my other options?

Cascading Selects are very popular in Web design in general, and in the SharePoint world in particular. An internet search for SharePoint and "cascading select" or "cascading lookup" or "cascading dropdown" will return many results.

InfoPath 2010 and 2013 allow you to build cascading dropdowns. Note that the future of InfoPath is unclear at this point, and Microsoft hasn't provided any roadmap.

The most popular option today seems to be client side scripting. Besides SPELL, a few other libraries also offer cascading selects.

SPServices is the most popular, and very well documented. Many other solutions advertised on the Web actually use SPServices under the hood. Its drawbacks: a strong dependency on jQuery and on the deprecated SharePoint SOAP services.

SharePoint Cascaded Lookups is another solution hosted on Codeplex.

The devil is in the detail! When selecting a cascading solution (no pun intended), go beyond the wow factor and test it thoroughly against your own scenarios. Some points to consider:

- Compatibility with multi-selects
- Compatibility with new browsers like IE 11
- Compatibility with SharePoint 2013 and in particular the Minimal Download Strategy (MDS)
- User friendly editor
- Does it break any existing feature? For example, do edits to a parent item propagate to the children lookups?
- Easy to uninstall or migrate?
- Size and performance

Terms of Use

CascadingSelects1.1.0Evaluation.aspx is provided for evaluation purposes and should only be used in test environments.

The full version comes with an End User License Agreement. Please read it. If you do not accept the terms, you must not install or use the application.

Support

User Managed Solutions offers support - webinars, forum, consulting hours - via the SPELL program. For more details, see the SPELL program page, or contact us (see information below).

Contact Information

Christophe HUMBERT

SPELL Program - User Managed Solutions LLC

E-mail	Christophe@UserManagedSolutions.com
Phone	+1-858-999-6321
Skype	ChristopheHumbert
Twitter	@UserManaged, @Path2SharePoint
Website	http://UserManagedSolutions.com/SPELL
Blog	http://blog.pathtosharepoint.com