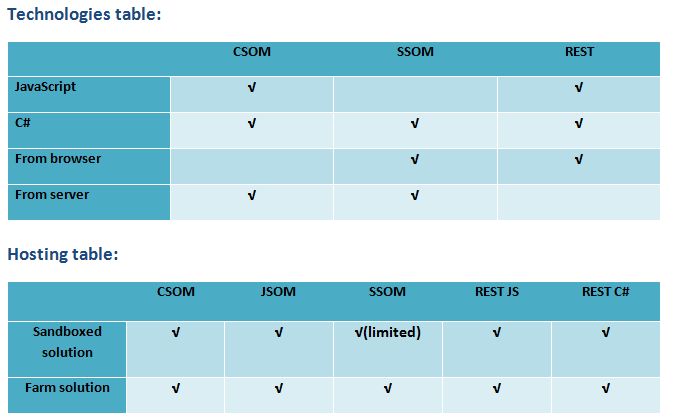
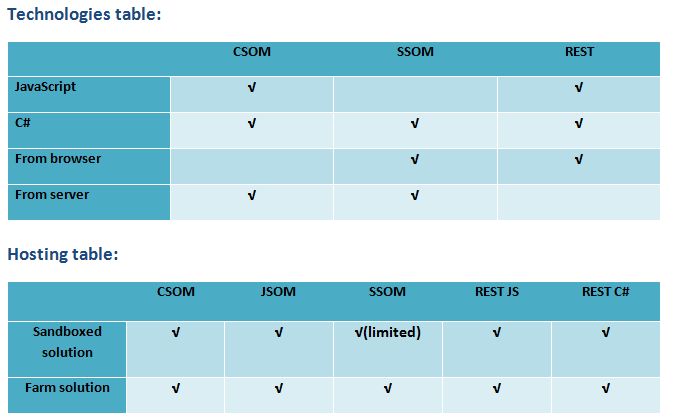
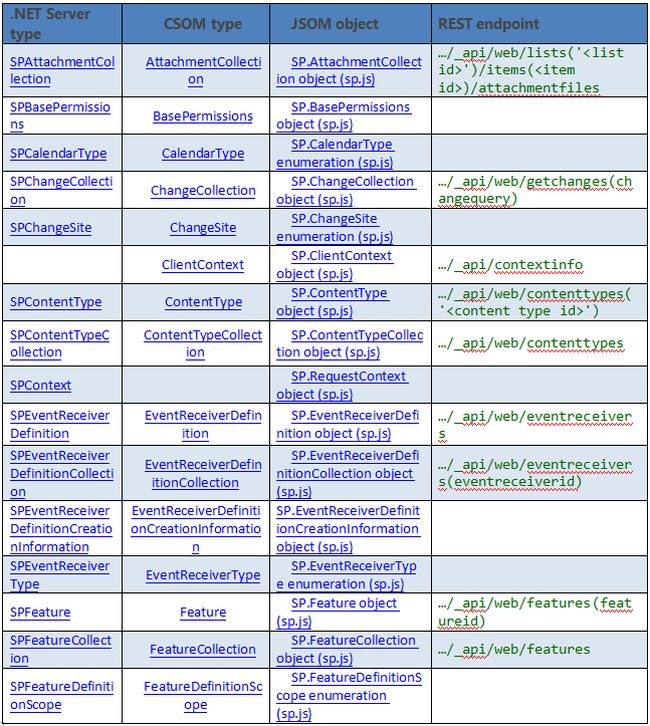
**Introduction**  
In this article we will explore different hosting types and technology abbreviations used in SharePoint 2010/2013 and elsewhere too. Useful API reference of the most frequently used types and objects implemented in Server object model (SSOM) and at least one client programming model: .NET client-side object model (CSOM), JavaScript object model (JSOM), and/or REST. Here's the difference between [CSOM/ JSOM/ SSOM/ REST](http://www.c-sharpcorner.com/blogs/sharepoint-2013-net-server-csom-jsom-and-rest-api-index).  
  
image  
  
**Model Types**

* **CSOM:**CSOM core assembly is Microsoft.SharePoint.Client.Runtime.dll and Microsoft.SharePoint.Client.dll, which stands for Client-Side Object Model. It is a web service based API of SharePoint. It allows access to SharePoint data and features from remote clients. CSOM was introduced in SharePoint 2010 and greatly enhanced in SharePoint 2013.  
    
  SP.ClientContext.get\_current() for normal use. New SP.ClientContext('url...') for specific SPWeb.
* **SSOM:**   
    
  Server Object Model core assembly is Microsoft.SharePoint.dll which is installed in the Global Assembly Cache. The Server Object Model will be executed in the server side & it provides rich set of classes in representing & manipulating SharePoint objects. Must be deployed on same farm Server-side object model. C# (or Visual Basic) uses only Microsoft.SharePoint (14) DLL. Must be deployed on same farm.
* **REST:**   
    
  The SharePoint REST interface is based on the REST-based Open Data protocol (OData) which is a platform-independent open standard. REST in 2010 is only for ListData.svc. Note that SPServices (jQuery plugin) utilizes the.asmx services in \_/vti/bin , such as /\_vti\_bin/Webs.asmx.
*   
    
  image  
    
  **Model Types**  
  **CSOM:**   
    
  CSOM, which stands for Client-Side Object Model, is a web services-based API of SharePoint. It allows access to SharePoint data and features from remote clients. CSOM was introduced in SharePoint 2010 and greatly enhanced in SharePoint 2013.  
    
  **JSOM:**Access information in the host web using the Javascript Object Model, you need to use an SP.ClientContext.get\_current() for normal use. New SP.ClientContext('url...') for specific SPSite. Note this works cross-SPSite in 2013.  
   **SSOM:**   
    
  Server Object Model is the most extensive API set available for SharePoint 2013. The core assembly is Microsoft.SharePoint.dll which is installed in the Global Assembly Cache. The Server Object Model will be executed in the server side & it provides rich set of classes in representing & manipulating SharePoint objects. Must be deployed on same farm.  
    
  **REST:**   
    
  The REST Services in SharePoint offer nearly the same functionality like JSOM. Furthermore, it is easier to use for a developer not coming from the SharePoint world since REST interfaces are standardized. This makes the choice of technology easier than for JSOM.  
    
  **Hosting**  
  **SharePoint hosted:**  
    
  A SharePoint-hosted app may provision basic resources into its app web such as HTML/CSS/JS files, site column/content type/list definitions, etc. Under no circumstances can server-side code run within a SharePoint-hosted app.  
    
  **Provider hosted:**Provider-hosted apps for SharePoint include components that are deployed and hosted outside the SharePoint farm. They are installed to the host web, but their remote components are hosted on another server, but is meant to be hosted more manually (i.e. On Azure). So you fire up a web application, on any server, and then use S2S to connect to SharePoint.  
    
  **Sandboxed:**   
    
  Remnant from 2010, can be uploaded to Solution gallery. Support for limited subset of the SSOM. No file access, so can’t deploy anything to \_layouts folder. If you use the limited SSOM-part (C#) your solution will be considered deprecated. Support for templates such as list templates and content types, deployed directly to host web.

**Farm solution:** Full access, deploy through PowerShell.  
  
  
  
**Note:**   
  
Microsoft officially announced the discontinuation of [Autohosted apps](http://blogs.office.com/2014/05/16/update-on-autohosted-apps-preview-program) model in SharePoint Online/Office 365 after it has been tested for the last few months.





**SharePoint 2013 Vs SharePoint 2010**

|  |  |
| --- | --- |
| **What is SharePoint 2013 (Preview) –** A new version of Microsoft famous Collaboration portal called SharePoint. The version adds few new exciting features such as Social Feed,SharePoint Apps and cross-site publishing. | **What is SharePoint 2010 –** It is a previous or I should say current version of SharePoint that was released in year 2010. |
| **Development Changes –**   * In SharePoint 2013 Microsoft Introduced a new Cloud App Model for designing Apps for SharePoint. Apps for SharePoint are self-contained pieces of functionality that extend the capabilities of a SharePoint website. You can use HTML, CSS, JavaScript and protocols like the Open Data protocol (OData), and OAuth to communicate with SharePoint using Apps. * Tools – SharePoint 2013 has Introduced new Tools for App development. Visual Studio 2012 now lets you develop apps for SharePoint and apps for Office. In addition a new web-based tools called “Napa” Office 365 Development Tools were introduced for developing apps. * No more Sandbox solutions. SharePoint 2013 sandboxed solutions are deprecated. So all we got is the New App model and the Old SharePoint Farm solutions. check out [SharePoint 2013 – Apps Vs Farm solutions](http://www.learningsharepoint.com/2012/07/20/sharepoint-2013-apps-vs-farm-solutions/) | **Development Changes –**   * SharePoint 2010 Introduced Sandbox solutions to help developers deploy code that did not effect the whole farm. * In SharePoint 2010 you could use Server Object model and Client Object model (.Net Managed, ECMASCRIPT and silverlight) to extract data from SharePoint. * In SharePoint 2010 developers were also developing Farm solutions as they did with the previous SharePoint 2007 version. |
| **Social and Collaboration features –** Microsoft in SharePoint 2013 Introduced new Social capabilities for better collaboration in the company.New Features added are –   * Interactive feed * Community Site * Follow people * Follow Sites | **Social and Collaboration features –**SharePoint 2010 had very few social capabilities.   * My sites * Tags and Tag profile pages * Notes |
| **Search –** SharePoint 2013 includes several enhancements, custom content processing with the Content Enrichment web service, and a new framework for presenting search result types. Some of the features added are –   * Consolidated Search Results * Rich Results Framework * keyword query language (KQL) enhancements | **Search –**SharePoint 2010 had Introduced Integrated FAST search as an Enterprise search. In addition to this build-in SharePoint search is still widely used in companies. |
| **Enterprise Content Management (ECM) –** SharePoint 2013 added some of the best capabilities of an ECM software. The newly added stuff is   * Design Manager * Managed Navigation * Cross-site Publishing * EDiscovery | **Enterprise Content Management (ECM) –** SharePoint 2010 on the other hand had Introduced Managed metadata and taxonomy as a part of new ECM benefits for SP 2010. This version did not had Managed Navigation and Cross-site Publishing. SharePoint designer was a primary tool to modify Master pages instead of the new Design Manager. |

## **Features deprecated in SharePoint 2013**

The following features and functionality have been deprecated or changed in SharePoint 2013.

### **Visual upgrade**

**Description:** The visual upgrade feature in SharePoint Server 2010 is not available in SharePoint 2013. For the upgrade from Office SharePoint Server 2007 to SharePoint Server 2010, you could choose to use the visual upgrade feature to give site collection owners and site owners the opportunity to preserve the previous user interface temporarily while still upgrading the infrastructure and databases, site collections, and features to the latest version. This allowed site collection owners and site owners to update customizations to work in the new user interface. Once the database and site collection upgrade was complete, the user had the option to upgrade the user interface on a more granular level of the website (SPWeb object).

**Reason for change:** The visual upgrade feature is replaced with deferred site collection upgrade. The site collection upgrade process is not reversible. The deferred site collection upgrade is a more comprehensive upgrade process than visual upgrade.

Visual upgrade preserved only the old master pages, CSS files, and HTML files. Deferred site collection upgrade preserves much more, including SPFeature functionality. To achieve the deferred site collection upgrade, major changes in the architecture were required, including the removal of visual upgrade.

With deferred site collection upgrade, you can continue to use the UI from the previous version (SharePoint Server 2010) more seamlessly than is possible with visual upgrade. The master page, CSS, JScript, and SPFeatures will remain in SharePoint Server 2010 mode. One key difference is that the granularity of upgrading the user interface is per site collection (SPSite) instead of site (SPWeb). Users can still preview their site in the new SharePoint 2013 user interface before committing. However, this is accomplished by creating and upgrading a temporary copy of their site collection instead of a preview in the existing instance of the site collection. The reason for previewing a copy of the site collection is because of the complexity of what occurs during site collection upgrade. Once a site collection is upgraded, it cannot be rolled back. Therefore, performing a preview would not be possible except in a copy of the site collection.

**Migration path:** Site collection administrators who are using visual upgrade to continue to use SharePoint Server 2007 must move to the SharePoint Server 2010 user interface before upgrading to SharePoint 2013. After the content database is upgraded, users can use deferred site collection upgrade to continue to use the SharePoint Server 2010 experience for their site collections. Site collection administrators can be notified by their farm administrator when a site collection is ready for upgrade and the site collection administrators can then choose to either perform the upgrade of their site collection or optionally first preview the new functionality in a temporary copy of their site collection.

Any SharePoint user interface might have dependencies on visual upgrade. The main dependency was getting the user interface version and then outputting the correct user interface (new or legacy). The visual upgrade API feature is updated so that the user interface version is remapped to the new **site collection compatibility level** property. This returns the same information about which version the site uses as before. Therefore, dependent code does not need to change.

### **Document Workspace site template**

**Description:** When you create a site in SharePoint 2013, the Document Workspace site template is not available.

**Reason for change:** The scenario of collaborating on a document is now provided by the Team Site site template. The Document Workspace site template was removed from SharePoint 2013 to simplify the list of templates that are available when a user creates a new site collection.

**Migration path:** Existing sites that were created by using the Document Workspace site template will continue to operate in SharePoint 2013. The Document Workspace site template will be removed completely from the next major release of SharePoint and sites that were created by using the Document Workspace site template will not be supported.

### **Personalization Site site template**

**Description:** When you create a site in SharePoint 2013, the Personalization Site site template is not available.

**Reason for change:** The Personalization Site site template was not a widely used site template. The Personalization Site site template was removed from SharePoint 2013 to simplify the list of templates that are available when a user creates a new site collection.

**Migration path:** Existing sites that were created by using the Personalization Site site template will continue to operate in SharePoint 2013. The Personalization Site site template will be removed completely from the next major release of SharePoint and sites that were created by using the Personalization Site site template will not be supported.

### **Meeting Workspace site templates**

**Description:** When you create a site in SharePoint 2013, all five of the Meeting Workspace site templates are not available. This includes the Basic Meeting Workspace, Blank Meeting Workspace, Decision Meeting Workspace, Social Meeting Workspace, and Multipage Meeting Workspace. In addition, the integration with Meeting Workspaces has been removed from Outlook 2013, and the commands to create a Meeting Workspace in Outlook 2013 have been removed from the Quick Access Toolbar and the Ribbon.

**Reason for change:** SharePoint 2013 and Office 2013 provide other features that support meetings and collaboration. For example, you can use Lync to conduct live meetings, OneNote to take notes during meetings, and a SharePoint team site or My Site to store shared meeting notes.

**Migration path:** Sites created using the Meeting Workspace site templates that are upgraded to the SharePoint 2013 user experience will no longer operate in SharePoint 2013. In the event continued use is necessary while a migration plan is determined, sites using the Meeting Workspace site template should be operated in SharePoint 2010 mode. Client integration features when operating in SharePoint 2010 mode require a 2010 version of the Microsoft Office client. The Meeting Workspace site templates will be removed completely from the next major release of SharePoint and sites that were created by using the Meeting Workspace site templates will not be supported.

### **Group Work site template and Group Work solution**

**Description:** When you create a site in SharePoint 2013, the Group Work site template is not available. This Group Work site template provides a groupware solution that teams can use to create, organize, and share information. The Group Work site template includes the Group Calendar, Circulation, Phone-Call Memo, document library, and other basic lists. The Group Work site template and the Group Work solution are discontinued and not available in SharePoint 2013.

**Reason for change:** The Group Work site template was not a widely used site template. The Group Work site template was removed from SharePoint 2013 to simplify the list of templates that are available when a user creates a new site collection.

**Migration path:** Existing sites that were created by using the Group Work site template will continue to operate in SharePoint 2013. The Group Work site template will be removed completely from the next major release of SharePoint and sites that were created by using the Group Work site template will not be supported.

### **Visio Process Repository site template**

**Description:** When you create a site in SharePoint 2013, the Visio Process Repository site template will continue to be available. However, the Visio Process Repository site template will be removed in the next major release of SharePoint.

**Reason for change:** The Visio Process Repository site template is not a widely used site template. The Visio Process Repository site template was removed from SharePoint 2013 to simplify the list of templates that are available when a user creates a new site collection.

**Migration path:** Not required. The Visio Process Repository site template is available in SharePoint 2013.

### **Unghosting and customizing CSS files**

**Description:** The following methods are included in SharePoint 2013, but will be removed from the next major

release of SharePoint:

* **Microsoft.SharePoint.SoapServer.Webs.CustomizeCss**
* **Microsoft.SharePoint.SoapServer.Webs.RevertCss**

The **Webs.CustomizeCss** method applies style sheet customization to a particular file.

The **Webs.RevertCss** method reverts style sheet customization of a file to the default style sheet.

These two methods are stored in Webs.asmx.cs and are defined in Webswsdl.asps.

**Reason for change:** The methods are outdated and are no longer needed.

**Migration path:** None.

### **Imaging Web service**

**Description:** The Imaging Web service provides functionality for creating and managing picture libraries. The Imaging Web service will be removed from the next major release of SharePoint. The Imaging Web service is included and supported in SharePoint 2013.

**Reason for change:** The Imaging Web service is not widely used. The only client application for the Imaging Web service, Office Picture Manager, is no longer included with SharePoint 2013. The Imaging Web service is being removed to reduce security vulnerabilities and to simplify the number of ways to connect to SharePoint 2013.

**Migration path:** All the functionality of the Imaging Web service is available through the client-side object model (CSOM). The CSOM provides client-side applications with access to a subset of the SharePoint Foundation server object model, including core objects such as site collections, sites, lists, and list items. Also, Web Distributed Authoring and Versioning (WebDAV) provides clients with key functionality of the Imaging Web service (for example, upload, download, and rename).

### **Excel Services — Can't edit workbooks in the browser that have external data connections**

**Description:** Workbooks with external data connections that use Windows authentication cannot be refreshed in the browser. Instead, you are prompted to open the workbook in the Excel client program. Workbooks that have database or Windows credentials stored either in the Secure Store Service or in the connection string can still be edited in the browser. This change applies only when Excel Web App in Office Web Apps Server is used to view workbooks, not when Excel Services in SharePoint Server 2013 is used.

**Reason for change:** This is a design limitation in SharePoint 2013.

**Migration path:** You can still refresh these workbooks in the Excel client program. Additionally, a service application administrator can configure that workbooks are viewed in SharePoint 2013 instead of Office Web Apps Server.

### **Web Analytics in SharePoint Server 2010**

**Description:** Web Analytics in SharePoint Server 2010 has been discontinued and is not available in SharePoint 2013. Analytics processing for SharePoint 2013 is now a component of the Search service.

**Reason for change:** A new analytics system was required for SharePoint 2013 that included improvements in scalability and performance, and that had an infrastructure that encompasses SharePoint Online. The Analytics Processing Component in SharePoint 2013 runs analytics jobs to analyze content in the search index and user actions that are performed on SharePoint sites.

SharePoint 2013 still logs every click in SharePoint sites and still provides a count of hits for every document. User data is made anonymous early in the logging process and the Analytics Processing Component is scalable to the service.

This analytics data is used in SharePoint 2013 to provide new item-to-item recommendation features, to show view counts that are embedded in SharePoint 2013 and Search Server user interface, to provide a report of the top items in a site and list, and to influence the relevancy algorithm of search.

**What happens to Web Analytics after upgrade:** The Web Analytics Service is not upgraded to the Analytics Processing Component in SharePoint 2013. When you upgrade to SharePoint 2013, the databases that contain the data from Web Analytics in SharePoint Server 2010 are not removed. These databases are not used by or maintained by the Analytics Processing Component in SharePoint 2013. This means that documents on sites in SharePoint Server 2010 that are upgraded will show a hit count of 0.

When you upgrade to SharePoint 2013, do not attach and upgrade the databases that contain the data from Web Analytics in SharePoint Server 2010. We recommend that you turn off Web Analytics in the SharePoint Server 2010 environment before you copy the content databases that you want to upgrade to SharePoint 2013.

Reports from Web Analytics for the top items in a site are carried forward. Reports that show browser traffic, top users of a site, and referring URL are not carried forward and are not used by the Analytics Processing Component in SharePoint 2013.

Administrative reports for the quota usage of site collections in the farm are not available in SharePoint 2013.

SharePoint 2013 does not support the Web Analytics Web Part. After a farm is upgraded to SharePoint 2013, all instances of a Web Analytics Web Part will not function. The page that includes the Analytics Web Part will render and a message appears that informs the user that the Web Part is no longer supported.

**Migration path:** None. Data collection for Analytics Processing in SharePoint 2013 starts immediately for sites, including SharePoint Server 2010 sites.

### **Managed code assemblies in sandboxed solutions**

UNRESOLVED\_TOKEN\_VAL(SandboxSolutions)

**Description:** While developing sandboxed solutions that contain only declarative markup and JavaScript -- which we call no-code sandboxed solutions (NCSS)-- is still viable, we have deprecated the use of custom managed code within the sandboxed solution.

**Reason for change:** We have introduced the new SharePoint app model as a replacement to those scenarios that required the use of managed code. All future investments will go to making the new SharePoint app model richer and more powerful. The app model decouples the SharePoint core product from the app runtime, and this enables much more flexibility and gives you the ability to run the code in the environment of your choice.

We realize that our customers have made investments in coded sandboxed solutions and we will phase them out responsibly.

**Migration path:** We recommend that all new development should use the new app model whenever possible. In scenarios where you have to develop a farm solution or coded sandboxed solution, we recommend that you design it so that it can easily evolve toward a more loosely coupled development model.

### **Blank site template**

**Description:** When you create a site in SharePoint Server 2013, the Blank Site site template is not available.

**Reason for change:** The same functionality is provided by the Team Site site template. The Blank Site site template was removed from SharePoint Server 2013 to simplify the list of templates that are available when a user creates a new site collection.

**Migration path:** Existing sites that were created by using the Blank Site site template will continue to operate in SharePoint Server 2013. The Blank Site site template will be removed completely from the next major release of SharePoint and sites that were created by using the Blank Site site template will not be supported.

## **Organization Profiles**

**Description:** The Organization Profiles feature is deprecated in SharePoint Server 2013. Organization Profiles contain detailed information about an organization such as teams, divisions, and other information that describes the organization’s hierarchy.

**Reason for change:** SharePoint features related to identities continue to evolve around the core concepts of users and groups, and SharePoint will not be investing further in OrgID.

**Migration path:** Existing solutions based on Organization Profiles will continue to operate in SharePoint 2013. The Organization Profiles feature will be removed completely from the next major release of SharePoint, and solutions created by using Organization Profiles will not be supported.

## **SharePoint Foundation 2010 deprecated search features**

The following functionality has changed in SharePoint Foundation search.

### **Search capabilities**

**Description:** The search capabilities of SharePoint Foundation 2013 have changed, and are now based on the same search implementation as SharePoint Server. This provides many improvements, but also means that the search configuration is very different.

**Reason for change:** Alignment of basic capabilities between SharePoint Server and SharePoint Foundation.

**Migration path:** No migration of search settings is supported.

## **SharePoint Server 2010 deprecated search features**

The following section provides details about the deprecated search features in SharePoint Server.

**Modifying the search topology using a web-based interface**

**Description:** SharePoint 2013 uses the web-based interface to show the current status of the topology. You change the topology by using Windows PowerShell. SharePoint Server 2010 also included a web-based option for changing the topology.

**Reason for change:** The core search architecture of SharePoint 2013 has a more complex and flexible topology that can be changed more efficiently by using Windows PowerShell.

**Migration path:** Use Windows PowerShell to modify the search topology.

### **Diacritic sensitivity element in the thesaurus**

**Description:** In SharePoint Server 2010, thesaurus files contain a <diacritics\_sensitive> element. This element determines whether diacritical marks such as accents should be ignored or applied by the search system when expanding a query with terms from the thesaurus. By default, the <diacritics\_sensitive> element is set to zero to ignore diacritical marks.

In SharePoint 2013, the <diacritics\_sensitive> element is not available. Instead, diacritical marks are always respected when matching query terms with terms in the thesaurus.

Diacritic variants are not automatically matched with query terms. Therefore, fewer query terms might be expanded by synonyms. For example, the thesaurus entry <munchen> is not matched with the query term <münchen>.

**Reason for change:** The feature has limited usage. The same behavior as in SharePoint Server 2010 can be achieved by adding diacritic variants in the thesaurus.

**Migration path:** Update the thesaurus dictionaries that are tagged as diacritic insensitive. To update thesaurus dictionaries, add diacritic variations of the relevant terms.

### **Replacement mode within the thesaurus**

**Description:** The thesaurus replacement mode is deprecated in SharePoint 2013.

In SharePoint Server 2010, you can classify entries in the thesaurus as expansions that are added to the query in addition to the original term. Likewise, you can classify entries as replacements of the original term in a query.

In SharePoint 2013, thesaurus replacements are no longer supported. All entries in the thesaurus are expansions, and the original term is not removed from the query. The original query term is always evaluated when you search the index. You cannot remove synonyms or words from the index.

**Reason for change:** The feature has limited usage, and may also have unwanted side-effects for relevance.

**Migration path:** No equivalent feature.

### **Search Query web service**

**Description:** The Search Query web service is deprecated in SharePoint 2013.

In SharePoint Server 2010, the Search Query web service exposes the SharePoint Enterprise Search capabilities to client applications. This enables you to access search results from client and web applications outside the context of a SharePoint site.

**Reason for change:** The Search Query web service is deprecated because the client object model (CSOM) and a new REST-based web service are available for developing Office-wide extensibility scenarios. The CSOM exposes the same functionality as the Search Query web service, and a larger set of functionality for stand-alone client applications.

**Migration path:** Change custom search solutions to use the CSOM or REST-based web service instead of using the Search Query web service.

### **Search RSS and search from Windows**

**Description:** The search RSS feature is deprecated in SharePoint 2013. The functionality for performing enterprise searches from Windows 7 depends on search RSS and this element has also been deprecated in SharePoint 2013.

The RSS link no longer appears on the results page. This link is replaced by the Search Alerts link.

Before upgrading site collections to SharePoint 2013, you can continue to use RSS in the SharePoint 2010 version of the Search Center. However, after you upgrade the Search Center to SharePoint 2013, the RSS is no longer available. In SharePoint 2013, you can create custom RSS feeds that use the client object model (CSOM), which targets the needs of your particular application and the RSS readers.

**Reason for change:** Most RSS readers that are available do not support claims authentication. In SharePoint 2013, claims authentication is the default authentication model. By using claims authentication, RSS readers work while the authentication cookie is cached. However, after the cookie expires, RSS readers cannot refresh their authentication, and so they stop working.

**Migration path:** After migrating a site to SharePoint 2013, you can create search-based alerts to be notified of changes to search results. You can also create a custom RSS feed in SharePoint document libraries, by using the UX extensibility platform.

### **Custom word breaker dictionaries**

**Description:** The format of the custom word breaker dictionaries has changed in SharePoint 2013. In SharePoint 2013, you can only create one language-independent dictionary. In SharePoint Server 2010, you can create language-specific custom dictionaries (one dictionary for each language) to edit the word breaker behavior of enterprise search. The word breaker behavior for East Asian (CJK) languages has not changed in SharePoint 2013.

In SharePoint 2013, custom word breaker dictionaries from earlier versions of SharePoint Server are not supported.

**Reason for change:** The search processing framework for SharePoint 2013 is new, and the way the word breakers operate has changed.

**Migration path:** You must combine existing custom dictionaries into one language-independent dictionary.

### **Configuration of stemming in the registry**

**Description:** The configuration of stemming in the registry is no longer supported in SharePoint 2013. Modifying stemming entries in the registry has no effect during search. In SharePoint Server 2010, you can turn stemming on or off, or you can replace it with a third-party stemmer by changing the registry. In SharePoint 2013, you cannot use a third-party stemmer.

**Reason for change:** This feature has limited feature usage.

**Migration path:** There is no migration path available for custom stemmers. You can enable or disable stemming in the Search Result Web Part.

### **SharePoint Search SQL syntax**

**Description:** In SharePoint Server 2010, you could construct complex search queries by using SQL syntax.

Search in SharePoint 2013 supports FAST Query Language (FQL) syntax and Keyword Query Language (KQL) syntax for custom search solutions. You cannot use SQL syntax in custom search solutions.

Custom search solutions that use SQL syntax with the Query object model and the Query web service that were created in earlier versions of SharePoint Server do not work when you upgrade them to SharePoint 2013. If you submit queries by using these applications, you will receive an error.

**Reason for change:** The core search architecture has changed in SharePoint 2013, and the SQL syntax is no longer supported.

**Migration path:** Change current search solutions to use either the KQL syntax or FQL syntax for queries.

### **Shallow search refiners**

**Description:** SharePoint Server Search in Office 2010 supported shallow search refiners. FAST Search Server 2010 for SharePoint supports shallow refiners and deep refiners. In SharePoint 2013, only deep search refiners are supported.

We recommend that you use deep search refiners to refine searches. In SharePoint 2013, deep refiners are an improvement to the existing FAST Search Server 2010 for SharePoint functionality. For example, the resource usage for each refiner is improved in SharePoint 2013.

In SharePoint 2013, you can view refiners as you did in the earlier version of the product. However, the refiners are now computed differently. They are created based on index structures that are aggregated across the full result set.

**Reason for change:** The shallow search refiners are replaced with an improved implementation of deep search refiners.

**Migration path:** No specific migration steps are necessary.

## **FAST Search Server 2010 for SharePoint deprecated features**

The following section provides details about the deprecated features in FAST Search Server 2010 for SharePoint.

### **FAST Search database connector**

**Description:** The FAST Search database connector is not supported in SharePoint 2013.

**Reason for change:** The connector framework for SharePoint 2013 is combined with the BCS framework and the Business Data Catalog connectors.

**Migration path:** Replace the FAST Search database connector with the Business Data Catalog-based indexing connectors in the BCS framework.

### **FAST Search Lotus Notes connector**

**Description:** The FAST Search Lotus Notes connector is not supported in SharePoint 2013.

The Lotus Notes indexing connector (BCS framework) provides similar functionality as the FAST Search Lotus Notes connector. The FAST Search Lotus Notes connector supports the Lotus Notes security model. This includes Lotus Notes roles, and lets you crawl Lotus Notes databases as attachments.

**Reason for change:** The connector framework for SharePoint 2013 is combined with the BCS framework and the Business Data Catalog connectors.

**Migration path:** Replace the FAST Search Lotus Notes connector with the Lotus Notes indexing connector, or with a third-party connector.

### **FAST Search web crawler**

**Description:** The FAST Search web crawler is not supported in SharePoint 2013.

The SharePoint 2013 crawler provides similar functionality to the FAST Search web crawler.

**Reason for change:** The crawler capabilities are merged into one crawler implementation for consistency and ease of use.

**Migration path:** Use the standard SharePoint 2013 crawler. The following table explains the differences between the FAST Search web crawler and the SharePoint 2013 crawler.

### 

|  |  |  |
| --- | --- | --- |
| **Feature** | **FAST Search web crawler** | **SharePoint 2013 crawler** |
| Refeed documents | You can refeed documents that you have previously downloaded to the index without having to recrawl them. | You can perform a full recrawl with similar functionality, but with slightly decreased performance of feeds. |
| Extract dynamically generated links and content from Java | You can extract dynamically generated links and content from JavaScript. | No longer supported. There is no replacement for this feature in SharePoint 2013. |
| Language-focused crawls | You can extract dynamically generated links and content from JavaScript. You can perform crawls focused on language.  You can focus a crawl on a certain language, by only following links from and storing content for documents that match specific languages.  This feature is intended for large scale crawls that target specific languages but that do not limit the crawl to a top level domain. | No longer supported. There is no replacement for this feature in SharePoint 2013. |
| Modify URIs | You can modify the URIs before crawling them.  Such a modification of the URI enables you to remove certain features of the URI, such as dynamic components, and to rename host names. | You can apply prefix-type URI rewriting with the "Server name remapping" feature in Search Admin. This allows you to perform the most relevant modifications of the URI. |

### **Find similar results**

**Description:** The Find similar results feature is not available in SharePoint 2013. The Find similar results feature is supported in FAST Search Server 2010 for SharePoint to search for results that resemble results that you have already retrieved.

**Reason for change:** The Find similar results feature is available only within the query integration interfaces, and it does not consistently provide good results in many scenarios.

**Migration path:** There is no migration path available.

### **FAST Query Language (FQL) deprecated features**

**Description:** The FQL features are aligned with the features of the SharePoint Keyword Query Language (KQL) syntax

The following table describes the FAST Query Language (FQL) features that are deprecated in SharePoint 2013.

### 

|  |  |
| --- | --- |
| **FQL operator or feature** | **Changed behavior in SharePoint 2013** |
| ANY operator | This operator has the same effect as the OR operator. |
| RANK operator | This operator is accepted but does not affect result ranking. |
| XRANK operator | This operator has a new and more flexible syntax.  The old syntax is deprecated.  The boost parameter is mapped to the new cb parameter. The boostall parameter is ignored. |
| STRING operator | The N parameter is accepted but ignored.  The MINEXPANSION/MAXEXPANSION parameters are not supported.  The ANNOTATION\_CLASS parameter is not supported.  For the MODE parameter, the following arguments are deprecated, and have the following behavior:   * ANY: Equal to the OR mode. * NEAR/ONEAR: Equal to the AND mode. * SIMPLEALL/SIMPLEANY: The query string argument is evaluated according to the KQL query syntax. |
| Implicit typing of numeric data types | The FQL parser is not search schema-aware, and some implicit numeric data typing is no longer supported. |

**Reason for change:** To simplify the query syntax, some redundant syntax features were removed from SharePoint 2013.

**Migration path:** The following table describes what to replace the deprecated FQL operators or features with.

### 

|  |  |
| --- | --- |
| **Replace this FQL operator or feature** | **With** |
| ANY operator | WORDS operator |
| RANK operator | XRANK operator |
| XRANK operator | New syntax |
| STRING operator | For proximity operations, use the NEAR/ONEAR operators. For mapping of end-user query text, use the KQL mode. |
| Numeric data types | Type numeric data explicitly. Use either the int/float/decimal operators, or consistently use decimal/floatsyntax (with decimals always included) in the query. |

### **URL Query syntax**

**Description:** In FAST Search Server 2010 for SharePoint, the URL-related managed properties (such as site, or path) are tokenized as a text string, and you can query any subpart of the URL. This includes STARTS-WITH, ENDS-WITH, PHRASE and proximity queries on URL properties. Special characters such as “/”, “\_” and “-”are handled as word delimiters.

In SharePoint 2013, the entire URL is tokenized as one word. This includes special characters such as “/”, “\_” and “-”. You can query these managed properties by:

* Searching for the full string for the site or path.
* Searching for the leading part of the site or path.
* Omitting the protocol part (http, https), and omitting the leading part of the domain address in the query expression, for the site managed property.

**Reason for change:** The implementation in SharePoint 2013 is aligned with SharePoint Server 2010 search. The FAST Search Server 2010 for SharePoint implementation has a very high query performance cost, especially when you search for the full URL or a leading subset of the URL.

**Migration path:** The following table provides details on how to change FAST Search Server 2010 for SharePoint query expressions to match the SharePoint 2013 URL query syntax.

### 

|  |  |
| --- | --- |
| **To match** | **Then** |
| The complete URL string | Search for the exact string. Special characters in the URL must match. Do not use the PHRASE operator. |
| The leading part of the URL | Do not use the wildcard character. |
| Any part of the URL | * Map the relevant crawled property to an additional managed property of type text. * Use this managed property as a property filter in your query. |

### **Specific search scope filters**

**Description:** In SharePoint 2013, search scopes are automatically converted to result sources.

In FAST Search Server 2010 for SharePoint, you can specify additional filtering conditions for search scopes, as described in the following table:

### 

|  |  |
| --- | --- |
| **Filter(s)** | **Description** |
| FQL scope | These filters may contain FQL syntax. In SharePoint 2013, you can use migrated FAST Search scope filters, but you cannot change them. |
| Alternative full-text index for the query | This filter provides a non-default full-text index for the full-text part of the queries.  In SharePoint 2013, you can use migrated FAST Search scope filters that contain an alternative full-text index. However, you cannot change or convert these filters to result sources. |

**Reason for change:** The search scope functionality was replaced by a more powerful functionality for result sources. For more information, see [Configure result sources for search](https://technet.microsoft.com/library/jj683115.aspx).

**Migration path:** You must convert FQL scope filters to corresponding result sources. You can use an alternative full-text index in the query syntax.

### **Anti-phrasing**

**Description:** The search anti-phrasing feature in FAST Search Server 2010 for SharePoint is not supported in SharePoint 2013.

Anti-phrasing removes phrases that do not have to be indexed from queries, such as “who is”, “what is”, or “how do I”. These anti-phrases are listed in a static dictionary that the user cannot edit.

In SharePoint 2013, such phrases are not removed from the query. Instead, all query terms are evaluated when you search the index.

**Reason for change:** The FAST Search Server 2010 for SharePoint feature has limited usage due to the limited number of customization options.

**Migration path:** None.

### **Offensive content filtering**

**Description:** The filtering of offensive content in search is deprecated in SharePoint 2013.

In FAST Search Server 2010 for SharePoint, you can choose to filter offensive content. Offensive content filtering is not enabled by default.

In SharePoint 2013, you can no longer block documents that contain potentially offensive content from being indexed.

**Reason for change:** The feature has limited usage.

**Migration path:** None.

### **Substring search**

**Description:** The substring search feature was removed in SharePoint 2013.

In FAST Search Server 2010 for SharePoint, substring search (N-gram indexing) can be used in addition to the statistical tokenizer in East Asian languages. Substring search can be useful for cases in which the normal tokenization is ambiguous, such as for product names and other concepts that are not part of the statistical tokenizer.

**Reason for change:** The feature has limited usage, and has very extensive hard disk requirements for the index.

**Migration path:** None.

### **Person names and location extractions**

**Description:** In SharePoint 2013, you cannot extract person names and locations from documents by using predefined extractors.

In SharePoint 2013, you can create custom extractors to extract person names and locations. The difference between the predefined extractors in FAST Search Server 2010 for SharePoint, and custom extractors in SharePoint 2013, is that custom extractors are only based on dictionary entries, whereas the predefined extractors also use extraction rules.

**Reason for change:** This feature has limited usage and usually requires extensive customization. In most cases, we recommend that you use customer-specific dictionaries.

**Migration path:** Use custom extractors for person names and locations.

### **Number of custom entity extractors**

**Description:** In SharePoint 2013, the number of custom entity extractors that you can define is limited to 12.

In FAST Search Server 2010 for SharePoint Service Pack 1 (SP1), you can define an unlimited number of custom extractors. You can use custom entity extractors to populate refiners on the search result page.

There are 12 predefined custom entity extractors in SharePoint 2013:

* Five whole-word case-insensitive extractors
* Five word-part case-insensitive extractors
* One whole-word case-sensitive extractor
* One word-part case-sensitive extractor

**Reason for change:** By using a predefined set of custom entity extractors, the content processing architecture is more simple and easier to use.

**Migration path:** Use the predefined set of custom entity extractors.

### **Supported document formats**

**Description:** SharePoint 2013 no longer supports rarely used and older document formats that are supported in FAST Search Server 2010 for SharePoint by enabling the Advanced Filter Pack. Both the ULS logs and the crawl log indicate the items that were not crawled.

In SharePoint 2013, the set of supported formats that are enabled by default is extended, and the quality of document parsing for these formats has improved.

**Reason for change:** The file formats for indexing are older formats and are no longer supported.

**Migration path:** You can work with partners to create IFilter-based versions of the file formats that can no longer be indexed.

### **Content processing extensibility**

**Description:** The FAST Search Server 2010 for SharePoint content processing extensibility feature has changed in SharePoint 2013. Content processing prepares an item from a content source for indexing and searching. The FAST Search Server 2010 for SharePoint content processing extensibility feature uses a sandbox where your custom code runs. See [http://msdn.microsoft.com/library/ff795801.aspx](https://msdn.microsoft.com/library/ff795801.aspx) on MSDN, FAST Search, for more information.

SharePoint 2013 provides a new web service interface for content processing extensibility.

The new implementation of this feature has the following improvements:

* The web service callout provides more flexibility about where the custom code runs than it does with the sandbox callout.
* You can define triggers for the web service callout to optimize performance.
* Content processing is performed on managed properties instead of on crawled properties. This makes it simpler to manage the items that are changed.

**Reason for change:** The content processing architecture of search has changed to improve performance and flexibility.

**Migration path:** To integrate with the new SharePoint content processing component, you must change the code. The custom content processing code must be packaged as a web service.

### **Custom XML item processing**

**Description:** FAST Search Server 2010 for SharePoint includes a custom XML item processing feature as part of the content processing pipeline. Custom XML item processing is not supported in SharePoint 2013.

**Reason for change:** In SharePoint 2013, the content processing architecture has changed. Custom XML item processing was removed and we recommend that you implement a mapping functionality outside SharePoint.

**Migration path:** Custom XML item processing can be performed outside the content processing pipeline, for example by mapping XML content to a SharePoint list, or to a database table.

### **Adding a test item to the index**

**Description:** DocPush is a test and diagnostic command-line tool that submits test documents to the FAST Search Server 2010 for SharePoint index. A similar command-line tool is not available in SharePoint 2013.

**Reason for change:** The administration and diagnostics of feeding and crawling has changed in SharePoint 2013.

**Migration path:** None. You can create test documents or test lists in SharePoint to test crawling and feeding. To remove items from the search index or to verify that there are any errors on an item, you can use the crawl log. See [View search diagnostics in SharePoint Server 2013](https://technet.microsoft.com/en-us/library/jj219611.aspx) for more information.

**Working with SharePoint list data using OData, REST and JavaScript**

Recently I’ve been doing a fair amount of work with SharePoint list data using OData and the SharePoint 2013 REST APIs; so I thought I would share some of my experiences with you.

As you are aware, SharePoint lists are far from a new thing. However, they do offer a really flexible method to store a lot of data. Creating data is simple through the OOB user interface and you can find many different ways of filtering, grouping and ordering your data using views.

However, what if you have a completely customised user interface, and need to both create and surface list based data into it? Using the client object model, the new OData and REST functionality in SharePoint 2013 is definitely one way, and this is the focus of this blog post.

**Working with OData and REST in SharePoint 2013**

I'm going to assume that you have some understanding of OData and REST. If you are really new to it as a concept then there are plenty of informative articles out there, so we are purely focusing on SharePoint 2013's implementation here.

My first impressions about the way that you construct your RESTful HTTP requests very closely mimics the client object model; so it doesn't take long for you to familiarise yourself with; lets take a look at a few examples:

Accessing a list with the client object model:

List.GetByTitle(Sales)

And with the corresponding REST endpoint:

<http://contoso/sites/sales/_api/lists/getbytitle('Sales'>)

In the example above, you can see that we are requesting a list titled 'Sales', which is what will be returned to us. If we wanted to have the items within the list returned then we would create the following:

<http://contoso/sites/sales/_api/lists/getbytitle('Sales')/items>

We can go a lot further than this. I have included a few examples below:

|  |  |
| --- | --- |
| **Operation** | **REST Endpoint** |
| Retrieve a single list | /lists/getbytitle('listname') |
| Retrieve all items in a list | /lists/getbytitle('listname')/items |
| Retrieve all items in a list and select a specific property | /lists/getbytitle('listname')/items?$select=Title |
| Filter items in a list, returning only entries that start with 'A' | /lists/getbytitle('listname')/items?$filter=startswith(Title, A) eq true |

 url: "<http://contoso/_api/web/lists/GetByTitle('Supplier')/items>", this is the URL to where we want the send the request; note this is our OData/REST HTTP endpoint.

**Differences between SharePoint 2010 and 2013 Workflows**

|  |  |  |
| --- | --- | --- |
| **SharePoint 2013** |  | **SharePoint 2010** |
| **.Net Version** | Version 4.0 |  | Version 3.5 |
| **Hosting Engine** | Workflows are hosted outside of SharePoint.  Windows Azure Workflow (WAW) is an installable product and can be hosted on an on-premise SharePoint farm. Workflow is now run as a separate service. Communication of workflows with SharePoint will happen via REST / CSOM or OAuth. Workflow architecture is shown in Figure 1 below. |  | SharePoint 2010 hosted the workflow runtime. |
|  | SharePoint 2013 workflow architecture.png  **Figure 1: Workflow Architecture in SharePoint 2013** | | |
| **Workflow Data Storage** | Workflow definitions reside within SharePoint and the actual workflow is stored on Windows Azure. |  | Workflow Data is stored in Content database. |
| **Execution Context** | Workflows run in the context of Azure workflow. |  | Workflows executed in the context of super user. |
| **Deployment** | Can be deployed On-Premise or in a hosted environment. You can have a dedicated workflow farm for large deployments. |  | On-Premise deployment. To deploy in a sandbox environment we need to write a full trust proxy. |
| **Analytics** | Analysis is very much improved for information related to workflows. |  | Analysis on the number of instances of workflows running, last run by and last run date are hard to acquire. |
| **Performance** | Since workflows are decoupled from SharePoint runtime it provides better stability, scalability and transparency. |  | Issues with scale and large deployments |
| **Types of Workflows** | Two types of workflows, SharePoint 2010 (Workflows hosted by SharePoint) andSharePoint 2013 (Workflows hosted by Windows Azure) as shown in Figure 2 below. New 15 workflow actions with Windows Azure Workflow . |  | Workflow Foundation 3.5 hosted by SharePoint 2010 |
|  | Workflow Options in SharePoint 2013.png  **Figure 2: Workflow Platform Types** | | |
| **Changes for developers** | Workflow Manager Client 1.0 is a redesigned workflow infrastructure that is built on Windows Workflow Foundation 4 and brings new power and flexibility to workflow authoring in SharePoint 2013. Developers can perform declaratively all of the actions as mentioned in the right hand for SharePoint 2010 by using services. SharePoint designer is significantly improved and many features that makes a developer's life easier like copy paste, undo, redo.  A new data type called DynamicValue is being introduced to capture run time dynamic values. |  | Write code for:   * Calling SharePoint API * Connect and Query Database * Consume ATOM/REST Feeds * Call Web Services/REST Services |
| **Visio Designer** | Visio Designer is integrated into SharePoint designer, |  | Create in Visio and Export to SharePoint Designer. |
| **General Improvements** | * Introduction of "Stages":    + Mitigates SharePoint Designer's lack of loop support   + Provides functionality of "state machine" workflows in WF 3.5 * Declarative workflows have loops:    + Loop # times / with condition / with expression * Declarative workflows can call REST/SOAP services |  |  |
| **Workflows for SharePoint Apps** | Visual Studio can be used to build workflows for SharePoint Apps whereas SharePoint Designer workflows cannot be used to build Apps.  Since workflows execute in the cloud and not in SharePoint it provides enormous flexibility in designing workflow-based apps for SharePoint. |  | Workflows execute in SharePoint hence it is difficult to protect the intellectual rights that goes into building apps. |
| **Packaging** | \*.wsp or \*.app |  | \*.wsp |

## **Workflow actions in SharePoint Designer 2013**

The following is a reference for workflow actions available for the SharePoint Workflow platform. In addition to the SharePoint Workflow platform, SharePoint Designer 2013 also supports the SharePoint 2010 Workflow platform. To view workflow actions for the 2010 platform,

### **Core actions**

Core actions are those that are most commonly performed, and they are grouped together for easy access.

| **Action** | **Description** |
| --- | --- |
| Add a Comment | Enables you to leave informative comments in the workflow designer for reference purposes. This is especially helpful when there are other users collaborating on the workflow. |
| Add Time to Date | Adds a specific time in minutes, hours, days, or months to a date (Year is not supported), and stores the output value as a variable. The date can be a current data, specific date, or a lookup. The 'Current Date' value returns UTC midnight. |
| Build Dictionary | Builds a Dictionary variable of key/value pairs.  **Note:** The Dictionary uses JSON notation to store data. For more information on the Dictionary variable, see [Understanding Dictionary actions in SharePoint Designer 2013](https://docs.microsoft.com/en-us/sharepoint/dev/general-development/understanding-dictionary-actions-in-sharepoint-designer) |
| Call HTTP Web Service | Functions as a method call to an HTTP web service and returns data using the JSON format. Basic authentication is supported through the RequestHeader.  For more information on the Dictionary variable, see [Understanding Dictionary actions in SharePoint Designer 2013](https://docs.microsoft.com/en-us/sharepoint/dev/general-development/understanding-dictionary-actions-in-sharepoint-designer) |
| Count Items in a Dictionary | Returns a count of the number of items in a specified dictionary. |
| Do Calculation | Performs an arithmetic calculation and stores the output value in a variable.  **Note:** For SharePoint, this action supports only the **Double** numeric type. Integers are not supported. Use of the "+" operator (concatenation) for strings is not supported. |
| Get an Item from a Dictionary | Returns a particular item from a dictionary variable. |
| Log to History List | Writes a message from a list of predefined message items to the workflow history list. |
| Pause for Duration | Causes a workflow to pause executing for a specified time interval, in days, hours, and minutes. |
| Pause Until Date | Causes a workflow to pause executing until a specified date and time. |
| Send an Email | Automatically sends an email message that contains a predetermined message to a user or group when a specified workflow event occurs.  **Important:** If the site is not added to the Trusted Sites list then emails are routed to the Outlook Junk folder. |
| Set Time Portion of Date/Time Field | Creates a timestamp, and stores the output value in a variable. You can set the time in hours and minutes and add a current date, specific date, or lookup. |
| Set Workflow Status | Sets the status of the workflow. |
| Set Workflow Variable | Sets a workflow variable to a value. You can also use this action when you want the workflow to assign data to a variable. |
| Go to Stage | Specifies the next stage to which flow control should be handed. |
|  |  |

### **Coordination actions**

**Table2. Coordination actions reference**

| **Action** | **Description** |
| --- | --- |
| Start a List Workflow | Starts a List workflow based on the SharePoint 2010 Workflow platform.  **Note:** The Start a list workflow has the following issues:> The 'Assignments' type field cannot be used as a parameter when the 2010 workflow has a TaskProcess action in it.> When multiple calls are made to the same 2010 workflow the result will be multiple data sources in the 2013 workflow lookup functionality. These data sources are all the same.> Variable names in 2013 cannot contain special characters such as '?' and '#'. If a 2010 workflow contains special characters then they will be converted to hexadecimal code in the 2013 workflow. |
| Start a Site Workflow | Starts a Site workflow based on the SharePoint 2010 Workflow platform..  **Note:** The Start a list workflow has the following issues:> The 'Assignments' type field cannot be used as a parameter when the 2010 workflow has a TaskProcess action in it.> When multiple calls are made to the same 2010 workflow the result will be multiple data sources in the 2013 workflow lookup functionality. These data sources are all the same.> Variable names in 2013 cannot contain special characters such as '?' and '#'. If a 2010 workflow contains special characters then they will be converted to hexadecimal code in the 2013 workflow. |

Coordination actions are used to invoke a workflow based on the SharePoint 2010 Workflow platform. For more information on Coordination actions, see [Understanding Coordination actions in SharePoint Designer 2013](https://docs.microsoft.com/en-us/sharepoint/dev/general-development/understanding-coordination-actions-in-sharepoint-designer)

### **List actions**

List actions group together actions that are used to manipulate lists and list items.

**Table3. List actions reference**

| **Action** | **Description** |
| --- | --- |
| Check In Item | Checks in an item that is checked out. You can check in items only from a document library. **Caution:** The workflow crashes if you try to check in an item that is not checked out. |
| Check Out Item | Checks out an item. The workflow verifies whether the item is checked in before it checks out a document. You can check out items only from a library in your site.  **Caution:** The workflow crashes if you try to check out an item that is not checked in. |
| Copy Document | Copies a document from the current list to a different Document Library list. |
| Create List Item | Creates a new list item in the list that you specify. You can supply the fields and values in the new item. You can use this action whenever you want a new item to be created with specific information. |
| Delete Item | Deletes an item.  **Note:** This action is terminated on the computer running the Workflow Manager workflow engine and throws a **System.InvalidOperationException** exception. There is no workaround. |
| Discard Check Out Item | Discards the changes and checks the item back in if an item is checked out and changes have been made to it.  **Caution:** The workflow crashes if you try to check in an item that is not checked out. |
| Set Field in Current Item | Sets a specified field in the current item to a specified value.  **Note:** If you need the workflow to pause until the value of the field has changed, use the **Wait for Event in List Item** action instead of this action. |
| Translate Document | Translates a document into a particular language  **Note:** Requires a preconfigured Machine Translation Service Application. |
| Update List Item | Updates a list item. You can specify the fields and the new values in those fields. |
| Wait for Event in List Item | [Enhanced version of Office 2010 action.] Pauses the current instance of the workflow to await a specified list item event. This action listens for two events: **ItemUpdated** and **ItemAdded**. |
| Wait for Field Change in Current Item | Waits for a field on the current item to equal a particular value. |

### **Project actions**

Project actions support the integration of Microsoft Project. They are used to build Project-based workflows. All of the Project actions are new in SharePoint Designer 2013.

**Table4. Project actions reference**

| **Action** | **Description** |
| --- | --- |
| Create Project from Current Item | Takes the current item and creates a new project in the SharePoint farm PWA site. Using App Steps with this action is not supported in Project Online. |
| Set Project Field | Sets a value for a particular field on Project Server.  **Note:** This action requires the project to be checked in first. If the project is not checked in, the workflow will be terminated and users cannot open that project in Project Web App. |
| Set Project Stage Status | Sets the status of the Project Stage.  **Note:** An exception occurs when a current project is checked out. |
| Set status field in idea list | Updates the status on the original list item that is associated to the current project. |
| Wait for Project Event | Waits for a particular Project Event. |

### **Task actions**

Task actions provide the ability to invoke a workflow based on the SharePoint 2010 Workflow platform from within a workflow based on the SharePoint Workflow platform.

**Table 5. Task actions reference**

| **Action** | **Description** |
| --- | --- |
| Assign a Task | Assigns a workflow task to a user and establishes a due date for completion of the task. |
| Start a Task Process | Creates tasks on multiple users and enables the tasks to be taken through a customized process. |

### **Utility actions**

Utility actions are actions that manipulate strings or find the interval between dates.

**Table 6. Utility actions reference**

| **Action** | **Description** |
| --- | --- |
| Extract Substring from End of String | Copies a specified number of characters starting from the end of a string and stores the output in a variable. |
| Extract Substring from Index of String | Copies a substring starting at a specified index in the string and places the value in a variable.  **Note:** Be aware that although the index value in Microsoft SharePoint Designer 2013 is zero-based, values in SharePoint Designer 2010 were indexed starting at 1. |
| Extract Substring from Start of String | Copies a specified number of characters beginning at the start of a string and stores the output in a variable. |
| Extract Substring of String from Index with Length | Copies out a substring comprising a specified number of characters, starting at a specified index in the string, and places the value in a variable.  **Note:** Be aware that although the index value in Microsoft SharePoint Designer 2013 is zero-based, values in SharePoint Designer 2010 were indexed starting at 1. |
| Find Interval Between Dates | Calculates the time interval in minutes, hours, or days between two dates and stores the output in a variable. |
| Trim String | Removes white spaces from the beginning and end of a string. |
| Find Substring in String | Finds a particular substring inside of a string and returns the index of the substrings's starting position. |
| Replace Substring in String | Replaces a particular substring with another substring. |
| Trim String | Removes white spaces from the beginning and end of a string. |