



HOW TO TACKLE INTEGRATIONS

SDL Tridion 2011 integration and extension points

by Bart Koopman

Executive Summary

SDL Tridion comes with a broad extension framework. This document explains how to utilize the standardized interfaces to integrate SDL Tridion with your IT ecosystem.

Whilst SDL Tridion offers you standard features to enable your content and channel strategy to drive the business end of your interaction, there will always be a wish to leverage your existing software investments by means of back-end integrations. SDL Tridion caters for this need through its extension framework. On both the Content Management, as well as the Content Delivery side there are a wide range of extension points, which allow you to extend functionality without the need for customizations.

LEARNING MORE

For this article you will need knowledge of the SDL Tridion Content Management System in general and we recommend you read up on this before reading this article. To learn more, check the links below for more information:

Technical Brief SDL Tridion Architecture – <http://www.sdl.com/wcm> (Multimedia & Download)

Open Data Protocol - <http://www.odata.org>

SDL Tridion Community – <http://www.sdltridionworld.com>

OVERVIEW OF INTEGRATION AND EXTENSION POINTS

The total number of extension points in SDL Tridion is plentiful with a lot of places to hook extensions into. The main ones can be described with what I like to call the “Big Six”. These consist of three extension points on the Content Manager and three on the Content Delivery side as shown below.



Figure 1: The “Big Six”

These six extension points are all based on open and industry-adopted standards allowing for easy integration. Figure 2 shows a simple overview of when to use which of these extension points.

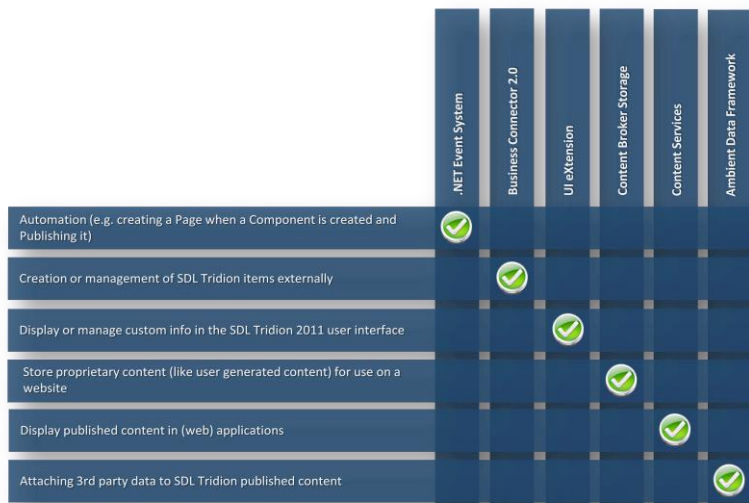


Figure 2: Extension points decision matrix.

CONTENT MANAGER

The Content Manager extension points allow you to leverage your business eco system. Each extension point integrates directly into the existing functionality and is designed for easy implementation.

User Interface Extensions

The Content Manager Explorer is the harmonization engine, which is built up out of views. These views can be extended to contain additional information or complete new views can be added to the user interface. This allows for management of 3rd party data from inside the Content Manager Explorer as if it were an integral part of the SDL Tridion interface.

The buttons and controls of the main view are displayed in a ribbon as shown in Figure 3. The ribbon is a tabbed toolbar, which contains controls for all the operations you can perform. The extension points of the ribbon and context menus include adding additional tabs and buttons as well as overriding the functionality of existing buttons. The controls of the ribbon and context menu are enabled or disabled based on selections made in the list view below and permissions of the user.



Figure 3: Ribbon toolbar.

The list view itself can be extended to contain additional rows or columns which can be populated with either existing SDL Tridion item information or information from a 3rd party system related to the selected items. The equally extensible tree view allows for display of additional items linking to 3rd party systems or data. To keep a concise view on data, the Content Manager Explorer will automatically add filters to each displayed list column.

Custom Pages

Custom Pages provide users with access to specific web pages. A custom page is a hyperlink to any URL which can be added to the Content Manager that all users will be able to access. Inside a Custom Page you can load 3rd party information as well as information directly from the Content manager itself. A Custom Page can be a useful way to integrate a dashboard for your content editors. More complex or integrated solutions are better implemented using the UI extensions.

Business Connector 2.0

The Business Connector 2.0 is a Windows Communication Foundation (WCF) service. The choice for a WCF service is based on its enterprise features. The use of the TOM.NET API is not supported outside the Event System and template code, because it is just a library which does not manage its own configuration or run in its own virtual machine. The Business Connector however has one configuration and all its resources managed in IIS.

With the WCF service you can integrate virtually anything with the Content Manager platform, either as part of a UI extension or as a standalone application. The performance of the Business Connector is designed to support synchronization tools as well as building a complete custom UI.

Event System

The SDL Tridion Event System allows you to subscribe to the delegate methods called by the SDL Tridion core. The events are split in different phases and you can subscribe to an event based on a subject type, event type and phase. The subjects are the SDL Tridion items (i.e. Components and Pages), the event type is the action taking place (i.e. save or delete) and the phase is the timing on/around the action (i.e. before or after). This setup will allow you to subscribe (synchronously or asynchronously) to a specific subject, action and phase or a combination of those (i.e. making it possible to add functionality which will be triggered when a Component and a Page is saved from a single code line).

Next to using the event system for automation (i.e. creating a Page and automatically publishing it after creating a Component) it can also be used for creating audit trail logs. The modular nature of the event system implementation will allow you to add multiple event systems to a single SDL Tridion instance - each having their own specific function and order in the process.

Workflow Automation

The workflow in its simplest form is defined as the movement of tasks through a work process. It ensures that specific individuals perform activities in a specific sequence. SDL Tridion Workflow is comprised of the following:

- A *Workflow Process Definition* contains the design of a Workflow. In a Process Definition you define a series of activities and assignees to ensure that specific tasks follow distinct steps to completion.
- When a Process Definition is applied to an item, each time the item is edited it enters into a *Workflow Process*. Each activity in a series of activities is started and finished by an assignee according to the Process Definition defined for the item.

Process Definitions are created in Microsoft Visio using the Visio Workflow Designer, an add-in to Microsoft Visio that provides a SDL Tridion specific toolbar and buttons. The interface allows you to create a flow of manual (performed by a User) and automatic (performed by the system) activities that are saved to the Content Manager. You can then associate the Process Definition to Components (via Schemas), Pages (via Structure Groups), or Page or Component Templates (via a Publication).

When an item enters Workflow, as each activity in the process is completed, the next activity is assigned to the next assignee until the Workflow Process is complete.

Application Data

Part of the extension framework is the option to store application-specific data on any SDL Tridion item, or storing it on a system-wide (global) level. This application data will be persisted in the SDL Tridion Content Manager database and can be set or retrieved through the APIs, allowing for seamless integration with 3rd party applications.

Single Sign-on

Authentication of the SDL Tridion Content Manager can be deferred to any modern 3rd party security system such as CS SiteMinder[®] or IBM Tivoli[®] Access Manager. This allows for a complete single sign-on integration of your entire back office. Important to keep in mind is that authentication of your extensions also needs to support the same security systems as SDL Tridion does. The UI extension framework, event system and workflow will automatically reuse the authentication of the SDL Tridion Content Manager, for the Business Connector 2.0 it might be required to cater for support of this single sign-on.

CONTENT DISTRIBUTION

Content Distribution is the part of SDL Tridion which connects the Content Management with the Content Delivery side. This includes the Rendering and Publishing engine. Figure 4 shows the extension points of Content Distribution.

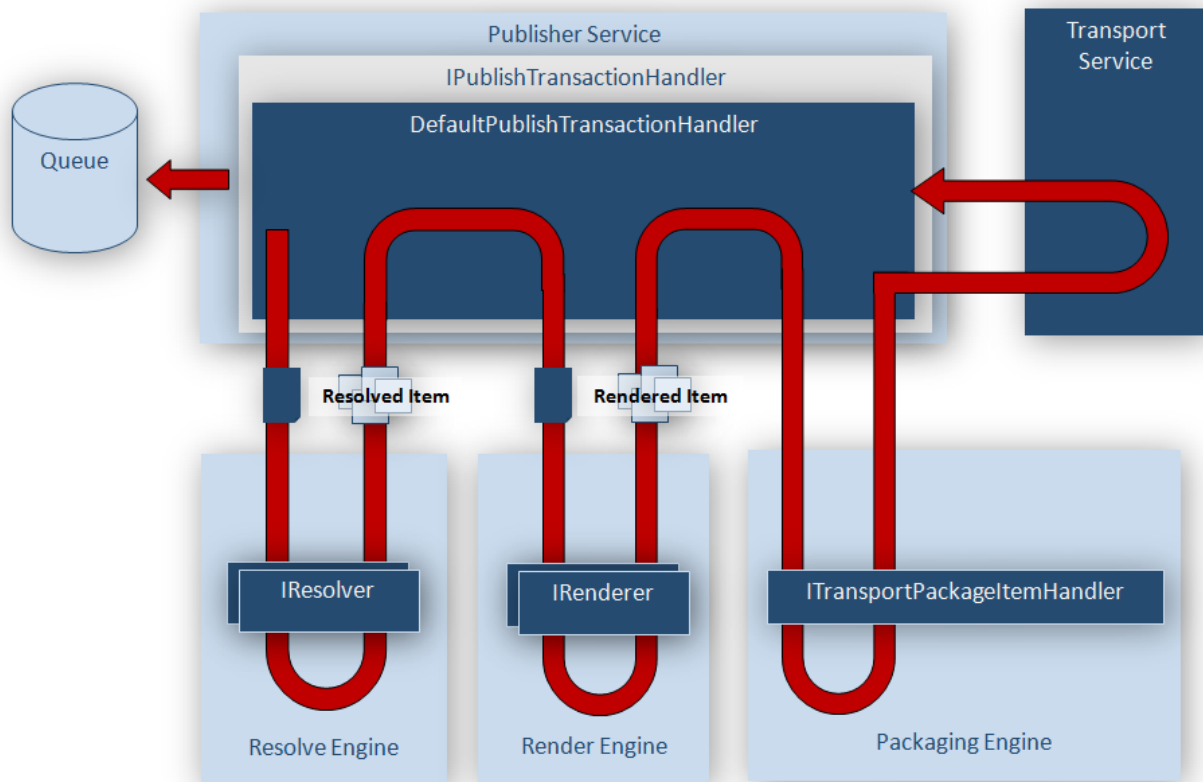


Figure 4. Content Distribution extension points.

The Publish process and the Transport process can both be extended using the IPublishTransactionHandler and ITransportPackageItemHandler interfaces respectively.

Most commonly however, the resolve and render processes are extended:

Custom Resolving

Extending the resolving process will allow you to add or remove items from the resolve result. This process decides which items should be published alongside the selected item. For instance when publishing a Page containing Dynamic Components the default resolver handles adding the separately published Components it to the Publish Transaction. This default behavior can be influenced by adding a custom resolver in the process. The process can be changed by using a custom resolver before or after the default resolver, or by completely removing the default resolver and replacing it with a custom one.

Custom Rendering

Extending the custom rendering process is similar to the resolving process. The Render engine handles rendering each resolved item based on the publish instruction and selected publication target. A custom renderer can change the rendered item, i.e. adding additional metadata to the item so this becomes available

on the Content Delivery side. Like resolvers, the renderers can also be chained in the render process, so you can keep the default renderer in place and add your customization as an extension.

For implementing specific enhancements on the render engine you might also consider the modular templating renderer. The modular templating framework handles templates in any technology by the use of mediators. These load, control and execute both the template and the underlying technology, giving it access to the underlying content package.

Where a custom renderer will be used to customize the output of the rendered item, a custom mediator will be used to implement a customization on the handing of a Template Building Block. A good example of this is the XSLT Mediator which allows for the use of XSLT Template Building Blocks as a replacement of Dreamweaver Template Building Blocks.

Functionality which is template specific should however be implemented directly inside a template (by the use of a Template Building Block for example) rather than in a custom renderer.

CONTENT DELIVERY

The Content Delivery extension points allow you to integrate your online ecosystem through the use of open standards such as JSR 220, OData, RESTful web services and OAuth.

Custom Storage

The storage layer is based upon the Java Persistence API (JPA) and its concrete implementation, Hibernate. It allows you to route storage by content type and/or publication so you can integrate both the SDL Tridion Content Broker as well as your proprietary storage in a single layer, making it available through a single API.

You can utilize this layer to store for example user-generated comments on published content, but also to store published content/assets directly to a self-owned or 3rd party Content Delivery Network.

Content Services

There is a growing trend on the web to make distinct pieces of content from a corporate website available on other websites, such as social networking sites or partner websites. For example, if you manage a hotel website, you may want to make a hotel room description available on, say, an airline's booking site, so that customers booking a flight can book your hotel room as part of their flight booking.

The SDL Tridion Content Delivery Web service is a thin layer on top of our existing Content Delivery APIs allowing you to easily integrate published content in any 3rd party solution. This OData-compliant Web service delivers up-to-date content in XML or JSON format which can be styled to create a custom look and feel.

Ambient Data Framework

The Ambient Data Framework (ADF) is a framework for gathering and transforming website data, for example information about the user visiting the website and information about the current user session (referral, location, language etc). As such, the ADF can pick up a visitor's geo-location and merge it with additional information from backend or 3rd party systems, like weather information, buying power, language etc. Organizations can then use this accumulated intelligence to deliver a more relevant experience to their audience across both desktop and mobile sites.

The framework allows you to program your own Claim Processors using the Ambient Framework Java API and organize them into cartridges that plug into the framework to implement specific functionality.

Extending Website Functionality

If you want to implement dynamic functionality for your website, you can do so in a number of different ways.

You can write code in ASP.NET or JSP directly on your Presentation Server, and have your SDL Tridion templates output script that calls this code, but this is unmanaged. You can also have SDL Tridion templates output the code directly in ASP.NET or JSP. This is a bit more managed, but the code is mixed in with templating code and is therefore difficult to debug. In addition, if third parties access your website through the Content Delivery Web service, the Web service returns the ASP.NET or JSP code you published, rather than the result of running that code, potentially making the outcome useless.

SDL Tridion offers ways to integrate your website functionality closely with the Content Manager. You can implement custom website functionality, with the code managed by SDL Tridion and correctly executed even when called from a Web service. For this you can create so called TCDL tags and implement their business logic in Java on the Content Delivery platform. TCDL is a language-neutral format for SDL Tridion dynamic functionality. The TCDL tags can be transformed at deployment time for ASP.NET or JSP websites/ applications (resulting in ASP.NET Server Controls or JSP Custom Tags) or at request time through the Content Delivery Web service, ensuring that a Web service call never gets code back in its response, but always content.