In-Store Publisher

Advanced Design and Setup Guide

Version: 9.5.x

Date: February 2016

Lexmark

Table of Contents

Overview

This document outlines the overall hardware and server requirements for the In-Store Publisher application environment. You can find the minimum system requirements for In-Store Publisher server and client workstations, as well as the database environments that In-Store Publisher supports, in *Publishing Platform for Retail Technical Specifications.* The ideal hardware and software configuration for your organization depends on the size and type of your business. This information is intended for general guideline. Contact your services consulting representative for additional advice on your specific needs.

System load and capacity factors

Choose hardware that supports your system when operating at peak system load. The following factors can contribute to, or help relieve, your system load.

- **Caching**. Specifically how application and web servers manage caching.
- **Database Access Speed**
- **Memory**
- **Network Speed**
- **Processing Power**

Estimate hardware requirements

There are many factors you should consider when estimating the total hardware requirements for your In-Store Publisher environment. The following factors can affect your hardware requirements.

Peak load factors

When you estimate your hardware requirements, a key factor is the peak system load you anticipate supporting. To calculate your peak load, you must pinpoint which hours of the day and which days of the week, month, or year generate the most signage. Also, determine the total number of store users who are accessing the system at any one time. Usually, you encounter peak loads when you hold a large store promotion and make a series of batches available to stores for printing.

Store operations processes and use patterns

When estimating your hardware requirements, consider operation processes, policies, and user habits. For example, do your store employees print the night before a sale, on the morning of a sale, or both? You should consider the following operational factors that affect the load.

Application use patterns

- How often you hold promotions or change prices and need to print signs.
- The time of day when your stores print signs.
- When you distribute to stores or users across time zones.
- The window of lead time you need for production.
- Other major publishing events, such as store openings or resets.

- Type of application use.

- The volume of batch printing activity.

- The volume of ad hoc sign creation.

- The volume of hand held request.

- The volume of hand held print requests.

Shared system resources factors

You should also consider the following environment factors.

- The availability of shared resources that other applications use, such as a server and a database.

- The availability of data sources.

- The bandwidth availability and capacity.

Application configuration factors

Your hardware needs can change based on the application configuration options you choose. Factors that affect how jobs print include partial batch reprinting, print job page count, complex collation, and page range. Factors that affect data integration include the type of pricing model used, side trips to point-of-sale systems for price look-ups, handheld application integration, and complex data import and export.

Template portfolio configurations can also affect your hardware needs. These factors include the total number of templates, the complexity of printed templates, template graphics, business rules (expressions), and advanced template design features, such as layers. Other application options that can affect your hardware needs include the number and nature of user interface (UI) pages, template edit screens and list grids, cache settings, and the number and type of print devices you use.

Preferred UI response times

When estimating your hardware requirements, consider web UI response times and printer speeds.

Application server types

You perform the following two primary web application server functions in the In-Store Publisher environment.

- **Web server**   A web server hosts the In-Store Publisher user interface by processing users requests for browsing the site, selecting batches and items, and creating ad hoc signs to print.

- **Publishing Engine server (or Print Engine server)**   The publishing server processes print requests by merging data, graphics, and formatting rules to generate and render print or display output.

   Note   Both the web and publishing servers require access to the Database Server.

Recommended hardware configurations

You can install the web server and publishing server on the same server or separate servers.

For high volume production environments, We recommend installing the web server and publishing (or print) server on separate dedicated servers or server clusters.

For low volume environments, a single server can handle both web and publishing functions or print functions.

Ensure that your system load balances properly if you plan for In-Store Publisher components to share server space with each other or with other unrelated applications.

The following table displays our recommendations for server hosting based on the volume of In-Store Publisher system usage. Each company has different use cases. If you expect to use your system at a volume close to the maximum, you should either follow the recommendations for the next-highest volume or invest in high performance hardware.

System Volume	Recommended Configuration
High. More than 400 stores.	Two or more web servers and two or more publishing servers behind load balancer Four-processor or better database server Maximum tiering Excellent failover protection
Medium. Between 30 to 400 stores. **Note** You can use either recommended configuration.	One or two web servers and one or two publishing servers behind load balancer or in a cluster
	Two to three combined web or publishing servers behind load balancer or in a cluster Two-processor to four-processor or better database server Minimal tiering Good to excellent failover protection
Low. Fewer than 30 stores.	One combined web or publishing server with no load balancer. Two-processor to four-processor or better database server No tiering No failover protection

Contact your services consulting team for recommendations based on your specific business needs.

Server tiers and load balancing

If your company has high volumes of activity, you can increase your throughput, performance, and uptime by placing multiple web and publishing servers within a load balancer or cluster server environment. The In-Store Publisher application architecture supports multiple load balancing methods.

You can configure load balancers to distribute requests to servers within each cluster, depending on which server has the lightest load at the moment. This enables you to add or remove servers without affecting users.

Web servers should have session redirection, also called "sticky sessions". Publishing servers can use round-robin redirection. If servers are down or not responding, the Publishing Engine and the Internet Information Services (IIS) environment return the appropriate HTTP headers to allow for redirection.

Hardware failover and redundancy

You can cluster the web and publishing servers in a web server farm to provide failover capabilities for each of the separate machines.

For large infrastructure environments, you should use a load balancer with lower cost, single or two-processor machines for added reliability and flexibility rather than a higher cost, higher capacity, four-processor machine for the web server.

Contact your system administrator for recommendations on hardware configuration options for server specific failover capabilities, such as disk mirroring and redundant network connectivity.

IIS worker processes and processor use

Web servers can only use one IIS worker process on a single machine so you can only use one processor for serving and processing the In-Store Publisher user interface.

Publishing and print servers can use multiple IIS worker processes on a single server; therefore, you can use more than one processor for print processing.

We recommend that you use multi-processor servers for both web and print functionality to optimize throughput.

Refer to the In-Store Publisher Load Balancer Configuration diagram.

Production, staging, and development server environments

Stage server environment overview

We strongly recommend that you set up a staging server environment as a clone of your In-Store Publisher production site. You can use the staging environment for testing and validating functionality, configuration changes, and other system modifications before making them available in the production environment.

Stage server environment options

The typical production system uses multiple servers in a server farm behind a load balancer. The staging server environment normally consists of one or two machines, depending on how closely you wish to mirror your production environment configuration. If your production configuration uses maximum tiering with separate web and publishing servers, then your staging environment should have two machines, one for each tier, otherwise a single machine is adequate. If your production configuration uses maximum tiering with separate web and publishing servers, the staging environment can host both web and publishing functions. However, you may not have the ability to fully test the performance and scalability of your system.

Development server environment options

You can clone additional copies of the environment into a development environment using minimal hardware for isolated development efforts.

Stage database options

Each environment requires a separate database but it is not necessary for a staging environment to be on a dedicated database server. It is common to use a shared database server for the staging or test databases. The staging database server's environment is essentially the same as the production system but its performance requirements can be scaled down.

We strongly recommend that you name all database and web servers consistently so their names indicate their function. For example, server names might include "DEV", "STG", and "PRD" for development, staging, and production.

Incremental environment scaling

You can easily scale up your environment with the In-Store Publisher architecture. You can install the application on minimal hardware infrastructure for initial setup and testing, then increase to accommodate full production levels closer to rollout.

We recommend you set up your first In-Store Publisher installation as a test system on minimal hardware requirements. After you configure and functionally test the application, you can replicate the environment and begin to build your production environment. After you roll out your production system, you can continue to use this server as a functional staging environment.

When you are ready to move to a production system for daily use, acquire separate machines to host the database, web, and publishing servers. If you run a scalability test and anticipate a system load too heavy for one set of servers to handle, you may decide to purchase additional servers.

If your system load increases, you can add more web or publishing servers, or both, depending on which part of the system you are using more heavily during daily use.

You can also convert a web server to a publishing server and vice versa to allocate your hardware to the function you need most.

Virtualized hardware

You may elect to use a virtual hardware platform. Virtual server platforms used by customers include VMWare, Xenserver, and Hyper-V.

There are many advantages to using virtual server platforms in the environment. Virtual servers are easier to maintain, support, modify, and replicate, making them an attractive option for web and publishing server clusters.

Refer to the following limitations for using virtual servers for In-Store Publisher.

- **Limited Throughput**   In some cases, virtual environments can provide significantly less throughput than physical servers with the same specifications. You can overestimate the hardware requirements by about 30 percent to accommodate this reduced throughput.

- **Limited Graphic Display Driver Resolution**   Some configurations of In-Store Publisher require you to use a high-resolution display driver to render output. However, some virtual server platforms have limited display-rendering capabilities. You should ensure your virtual server environment has high-resolution display driver capabilities to mitigate this display issue.

You might find virtualized hardware helpful when hosting development and staging environments. You should note, however, that it is not ideal to use different hardware to configure staging and production environments because it is difficult to reproduce and resolve issues you might see in your production configuration.

64-bit processing support

Publishing Platform for Retail supports 64-bit operating systems but some products are compiled in 32-bit and do not fully use 64-bit processing architecture.

You must use the 32-bit control panel ODBC manager to set up Data Source Name (DSN) connections.

Database server environment

Disk storage - Storage Area Network and standalone systems

If you run a smaller organization, you might want to use standalone systems for your web, publishing, and database servers. For example, if you use a dual-CPU, dual-core server with six drives in a RAID 5 configuration, a standalone system may adequately handle your expected database needs, both in traffic volume and future data growth.

You could store frequently accessed data, such as database files, on a Storage Area Network (SAN) system. SAN systems are suited for high transactional environments where there is a large demand for fast data reading and writing, data redundancy, and a relatively easy path for disk space growth. However, SAN systems may be more costly than comparably equipped Network-Attached Storage (NAS) systems or standalone servers.

When installing and using your software solutions, you can use either approach. Both approaches are adequate but have different failover and management capabilities.

Server redundancy - grid cluster and failover cluster

We strongly recommend data and database redundancy. Refer to the following two approaches to database server redundancy for information on the different options available.

Grid cluster

A grid cluster is a cluster of several database servers attached to a shared disk array backend, such as a SAN, where the data resides. Usually these servers are connected to the SAN using a fiber card and not an Ethernet card. When a client or application needs data, it calls the cluster.

In the grid clustering approach, any of the servers in the cluster can respond by retrieving data from the shared disk array and returning it to the client or application. As data requests increase, you can add additional servers to the cluster to handle the load.

Failover cluster

The second approach is a single, redundancy-equipped server or primary node that handles your database needs. You configure a similarly equipped server as a failover cluster node that connects to the same shared disk array as the primary node. For the shared disk array, you can use SAN, NAS, or mirroring to ensure that you can access database storage from all nodes. The primary and failover nodes are typically connected to one another with standard Ethernet for the "heartbeat" signal to pass between them. If the primary node fails, the failover node takes over, keeping perceived downtime close to zero. The primary node can be repaired or replaced and then brought back online. Multiple nodes can be used in multiple failover scenarios. Secondary nodes can be either active or passive. Active nodes allow use of the secondary machine for other unrelated tasks.

Microsoft SQL Server uses failover clustering when database server redundancy is required. You do not need a separate license for the Microsoft SQL Server software running on the failover server.

Note Adding more servers to a failover cluster configuration does not boost system performance because only one server at one time handles all database requests. However, if you add cluster nodes, you increase performance because concurrent nodes have access to the database at the same time.

Data models - Master Data Model and standalone

Critical and highly used data is held in a central repository accessed by many systems in the Master Data Model (MDM). For example, you may store your retail item data, such as pricing, description, and SKU, in a common database shared by your point of sale (POS) systems, e-commerce website, and signage software. When item data changes, it appears immediately across the other systems that share the data. However, if errors occur in item data, the errors may reflect across all systems that use the data. Also, you must test any changes to your data structure across all systems.

In a standalone data model, individual systems control their own data for their own use. For example, your POS and e-commerce systems might each have their own distinct database systems that house their own item and pricing data. Because the systems are independent of one another, if the POS system's item data changes, the change does not reflect in the e-commerce system and vice-versa. Note that it is difficult to synchronize data in this model. However, if data and the data structure change in one system, it does not negatively affect the other systems.

Publishing Platform for Retail products can perform well with either data model. Choose the model that suits your needs. Depending on the degree of MDM integration, your performance will vary from a stand-alone model.

Estimate hardware requirements

You can estimate hardware infrastructure requirements by forecasting application use, operational patterns, and configuration options.

Consult your services consulting representative for hardware infrastructure recommendations.

Determine capacity limits

To determine precise hardware capacity limits, We recommend that you perform scalability and performance load testing on a fully configured test environment in the client-specific environment considering client specific factors.

Appendix A: Load Balancer Configuration

In-Store Publisher Load Balancer Configuration diagram

We recommend that you use multi-processor servers for both web and print functionality to optimize throughput.
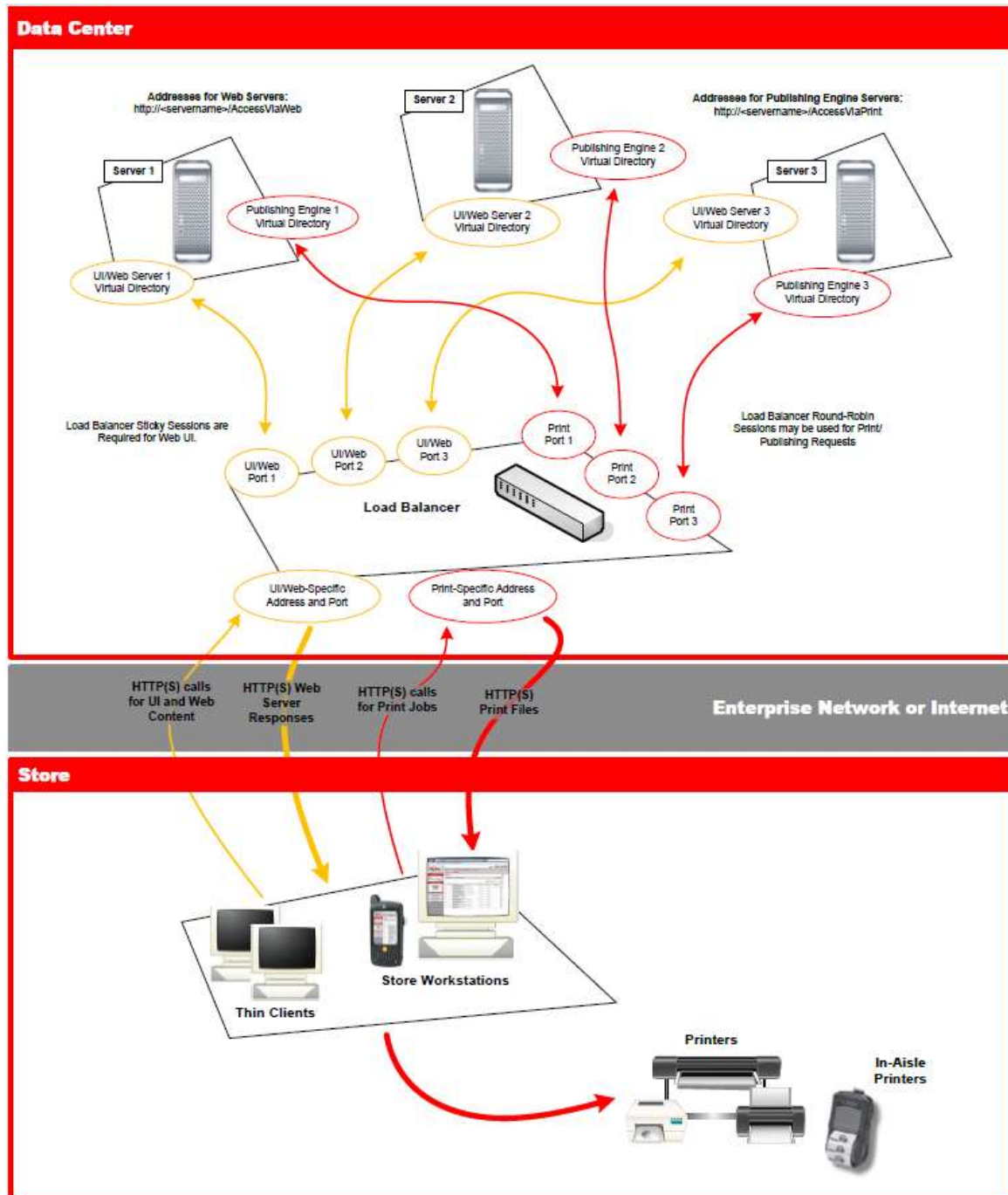
Appendix B: Staging and production environment

Staging and Production Environment diagram

Each environment requires a separate database but it is not necessary for a staging environment to be on a dedicated database server. It is common to use a shared database server for the staging or test databases. The staging database server's environment is essentially the same as the production system but its performance requirements can be scaled down.