Content Matrix Performance Optimization Guide

May 2017

www.metalogix.com info@metalogix.com 202.609.9100

Copyright

© 2017 Copyright Metalogix Corporation International GmbH.

All rights reserved. No part or section of the contents of this material may be reproduced or transmitted in any form or by any means without the written permission of Metalogix Software Corporation.

Metalogix Content Matrix Consoles™ are trademarks of Metalogix Software Corporation.

Windows SharePoint Services is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Technical Support

For information about Metalogix Technical Support, please visit http://metalogix.com/support.

Technical support specialists can be reached by email at support@metalogix.com or by phone at 1.202.609.9100.

We want to hear from you. Please send any comments regarding this document to support@metalogix.com.

The level of technical support provided depends upon the support package that you have purchased. Contact us to discuss your support requirements.

Contents

Introduction 4

Requirements 4

Scaling Hardware 4

Scaling Out 4

Dedicated Migration Servers 4

Installation on WFE Servers 6

Scaling Up 6

Distributed Migration 7

Configuration 8

PowerShell 9

Hardware Best Practices 9

Non-WFE server/agent 9

WFE server/agent 9

Connection Adapters 9

Database (DB) 9

Local Object Model (OM) 10

Metalogix Extensions Web Service (MEWS) 10

Native Web Services and CSOM (NWS) 11

Disable Unnecessary Features 11

Multithreading and Batch Jobs 12

Single Jobs 12

Configuring Single Job Multithreading 12

Optimizing Single Job Multithreading 13

Concurrent Jobs 13

Configuring Concurrent Jobs 13

Optimizing Concurrent Jobs 14

Batched Jobs 14
Distributed Jobs 15
PowerShell 16
Configuring PowerShell 16
Optimizing PowerShell 16
StoragePoint and Content Matrix 17
Configuring StoragePoint and Content Matrix 17
Next Steps 17

Introduction

Metalogix Content Matrix Console and its additional components, Metalogix Organizer and the Metalogix Extensions Web Service, allow users to migrate their data into SharePoint, and continuously improve usability and findability after that migration.

This document will cover a number of different topics to help you with optimizing the performance for migrating your content into SharePoint with Content Matrix Console. These topics include: hardware best practices and scaling, SharePoint connection adapters, PowerShell, etc.

These migration optimizations are only a general set of guideline to help you with your migration. For more information on any of the specific features and abilities of Content Matrix Console, please see the appropriate Edition's Help file (located within the client application, or they can be found online at: https://www.metalogix.com/documentation).

Requirements

The only requirement that is necessary to use this document is to be in the process of installing, or have already installed, Content Matrix Console into your environment, for migrating your content into SharePoint. Since appropriate installation of Content Matrix can have a major impact on performance, it is preferable to modify your installation if necessary based on the guidance in this document.

Please refer to the below documents for details on installing Content Matrix:

- Content Matrix Console Quick Start Guide.pdf
- Content Matrix Console Advanced Installation Guide.pdf

Scaling Hardware

Content Matrix can take advantage of scaling hardware within the migration environment.

Scaling Out

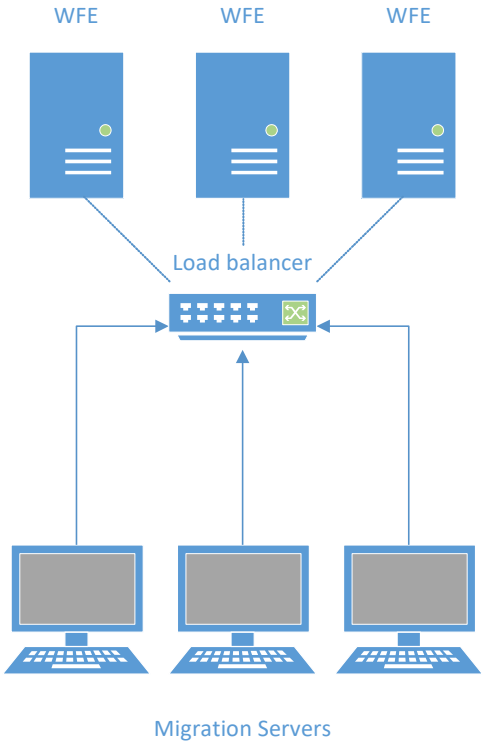
Scaling out is the process of adding servers with Content Matrix installed to perform migrations concurrently to different Web Front End (WFE) servers within the target SharePoint farm.

Dedicated Migration Servers

Content Matrix can be installed on dedicated migration servers to facilitate maximum resource utilization in migrations whenever possible. In this configuration, Content Matrix can then be remotely pointed at the target SharePoint farm.

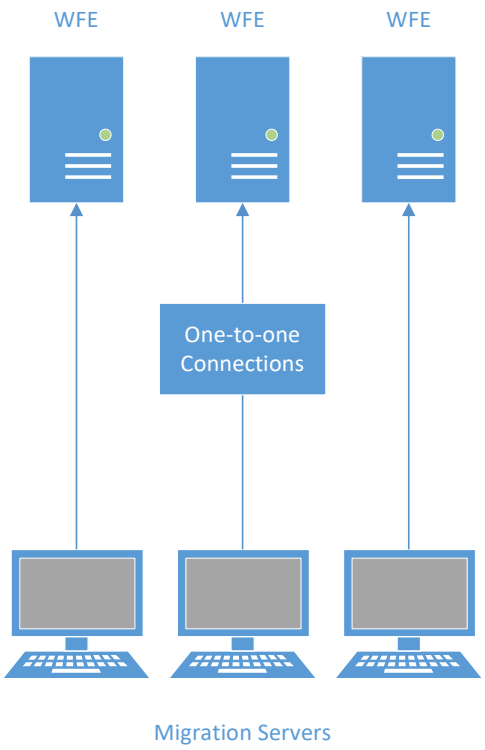
Note: This deployment scenario will require the installation of the Metalogix Extensions Web Service adapter on the target SharePoint server (as defined in the Connection Adapters section of this document). This will allow for remote connections to the SharePoint Object Model (OM).

Load balancing can be effectively used in this case to ensure that migration requests from Content Matrix will be properly balanced to WFE servers in the target farm based on utilization.



The number of migration servers that can be utilized should be managed to ensure that the target SharePoint server can effectively handle all of the migration requests. Generally a one to one correlation with WFE servers is recommended.

If load balancing is not being used on the target farm, migration servers can be directly pointed at specific WFE servers to ensure maximum concurrent throughput. See below image:



Installation on WFE Servers

When dedicated migration servers are unavailable, Content Matrix can be directly installed on WFE servers. In this configuration, Content Matrix can utilize the OM adapter for connections.

In this configuration, load balancing will be bypassed due to Content Matrix utilizing only the local WFE OM for migrations.

It is recommended that Content Matrix be installed on the maximum number of idle WFE servers in order to maximize throughput.

> **Note:** If WFE servers are already servicing user requests in a production environment, extreme care should be taken to ensure that the user experience is not impacted. Due to the potential resource utilization of Content Matrix on the host server, it is recommended that migrations via the application never be run on a production WFE **outside** of a maintenance window.

Scaling Up

Scaling up is the process of adding additional hardware to a single migration machine in order to facilitate faster copies.

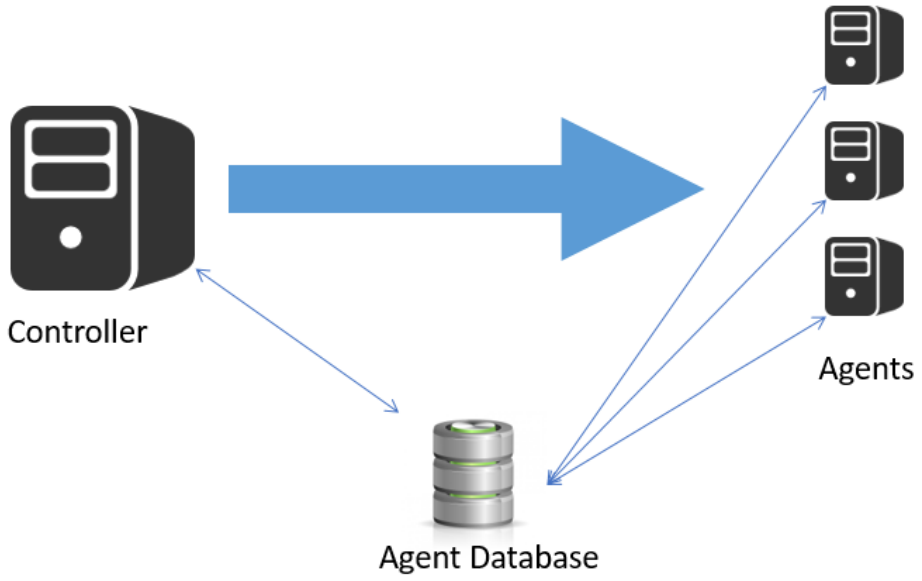
In general, scaling up is only recommended to the point specified in the hardware best practices section below. It is more efficient to scale out the migration infrastructure and parallelize migration instead of overly increasing resources on a single machine.

Distributed Migration

Content Matrix is able to leverage a unique feature called Distributed Migration in order to shift the load of multiple jobs to multiple machines using an Agent-Controller model for content distribution.

In the Agent-Controller model, one installation functions as the primary source for the distribution of migration jobs across multiple machines. The Controller does not (by default) participate in the migration process and only functions as a conduit for generating, configuring, and distributing jobs across the multitude of Agents. This does not prevent the Controller from executing jobs locally and independently.

On the Controller a Configuration Wizard will guide through the appropriate steps to generate an architecture similar to below:



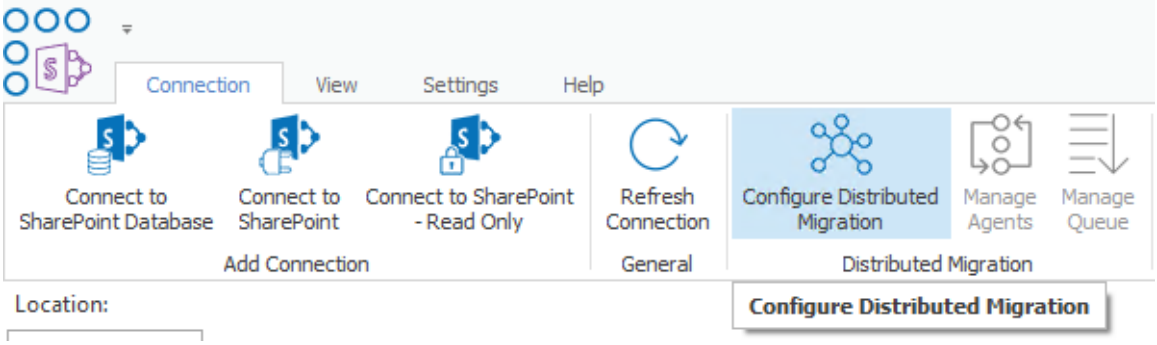
In this architecture, there is a shared database across all the listed machines. This database is known as the "Agent Database" and contains a shared list of all jobs configured for migration in this model, as well as a listing of all Agents and (if configured) a shared grouping of settings for all consoles.

In the previously picture scenario, the Controller generates all jobs and writes these into the Agent Database. The Controller then queues up the listed jobs, identifies any Agents that are not currently performing a migration, and pushes that job to that Agent. Should there be no free Agents, jobs will remain queued until an Agent becomes available. As an Agent migrates content it will write job information back to the Agent Database. This information can then be viewed from the context of the Controller.

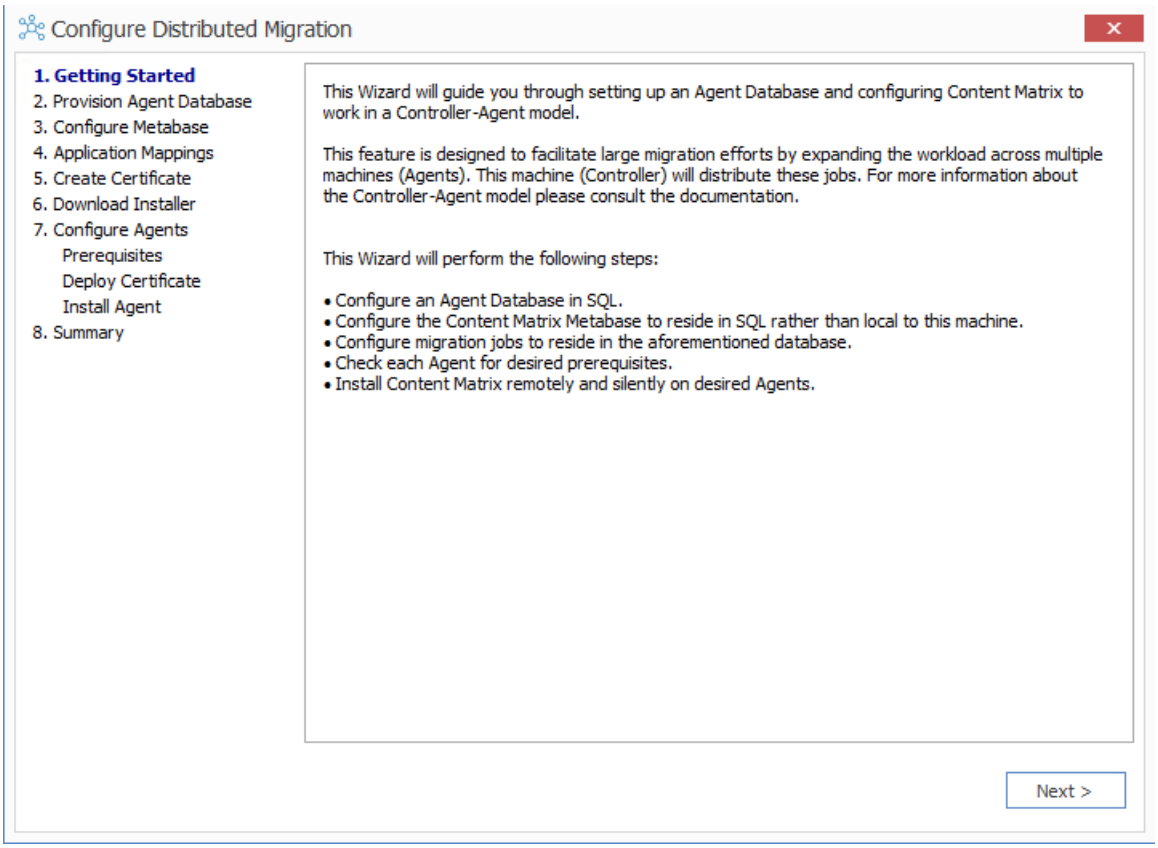
Configuration

The following steps should be followed for leveraging Distributed Migration appropriately:

- Launch Content Matrix SharePoint Edition.
- Within the top ribbon of the console is a button labeled "Configure Distributed Migration":



- Upon clicking that button, a Wizard will pop-up walking the user through the configuration. The Wizard will provide information for all actions being performed. Please consult the documentation should there be concerns with any particular screen or configuration.



PowerShell

As PowerShell extensibility is one of the major tenants of Content Matrix, that extensibility has been brought to the concept of Distributed Migration. Please reference the document "PowerShell Distributed Migration" for further details of how to leverage PowerShell for an Agent-Controller model.

Hardware Best Practices

Please consult the following document for minimum requirements when installing Metalogix Content Matrix:

Content Matrix Console Advanced Installation Guide.pdf

In addition to the minimum requirements laid out in the prior document, migration servers where Content Matrix will be installed will perform optimally with the following hardware configurations:

Non-WFE server/agent

- 4 core CPU or higher
- 8gb of RAM or higher
- 20gb of free disk space

WFE server/agent

- 8 core CPU
- 16gb of RAM or higher
- 50gb of free disk space

Note: Please avoid installation of Content Matrix on a disk that also hosts the SQL server for the target SharePoint installation. This can result in significant resource utilization, and will eliminate many of the benefits of parallelization.

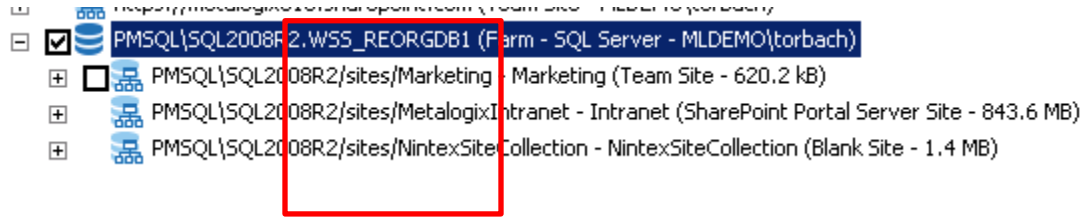
Connection Adapters

There are four primary adapter types available in Metalogix Content Matrix. They are defined below:

Database (DB)

The DB adapter is a **read-only** adapter designed for source side connections. If possible, **always** use the database adapter as the source for migrations, as it is the fastest adapter available in the application. It is also particularly useful for WAN based migrations or migrations to Office 365. This is because you can take database backups of your Content DBs, and move the resulting .bak files to a SQL Server on the target LAN (or Azure VMs in the case of O365 migrations), and then do a local migration.

DB connections look like this in the Content Matrix UI once connected:

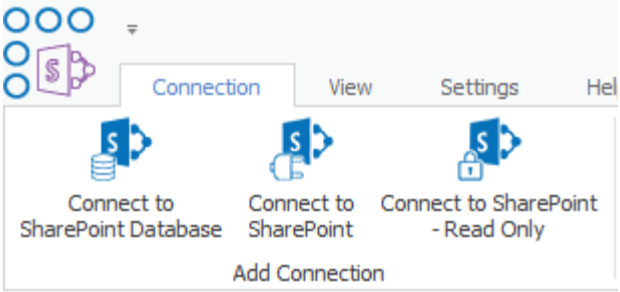


While the Database adapter does have some limitations, such as the inability to migrate Managed Metadata Services from one farm to another, users can use MEWS for that part of the migration, and the DB Adapter for the rest of the migration for best performance.

Local Object Model (OM)

The OM adapter is the second fastest adapter available in the application, and the fastest adapter for writing content. It can only be accessed when Content Matrix is directly installed on a WFE server in the SharePoint farm. This adapter is preferred if Content Matrix is installed on a WFE server in the **target** SharePoint farm (as of Version 8.3.0.1 of Content Matrix, this is only available in SharePoint 2013 and SharePoint 2016).

OM farm connections look like this in the Content Matrix UI once connected:



Metalogix Extensions Web Service (MEWS)

The MEWS adapter is a very fast adapter, but requires installation of a solution on the source or target SharePoint farm which it is being used in. The solution is included in the Content Matrix installer, and can be installed by running the installer on any WFE in the target SharePoint farm and selecting the "Install Content Matrix SharePoint WSPs" button, then checking the box to install the "SharePoint Extensions Web Service." If the target SharePoint farm will be connected to remotely, this adapter is preferred.

MEWS farm connections look like this in the Content Matrix UI once connected:

Note: The distinguishing factor between an OM connection and a MEWS connection is the "Local Connection" or "Remote Connection" string to the right of the connection.

Native Web Services and CSOM (NWS)

The NWS adapter should be avoided at all costs if optimizing for performance or functionality. Please utilize the DB adapter for source connections, and the OM or MEWS adapter for target connections wherever possible as defined above.

NWS connections look like this in the Content Matrix UI once connected (Note the globe behind the site icon. If this icon is showing on any of the currently connected nodes in Content Matrix, immediately attempt to remedy the problem by connecting to a different adapter type):

Disable Unnecessary Features

The following options can significantly increase the time taken to migrate content, and should be turned off whenever possible:

- Verbose Logging
  - **Location:** In the configuration dialog, under the "General Options" tab. A checkbox labeled "Verbose".
  - **Function:** Turns on verbose logging. Is unnecessary unless full XML of all objects is required in the job listing.
- Item level permissions
  - **Location:** In the configuration dialog, under the "Permission Options" tab. A checkbox labeled "Copy Item Permissions"
  - **Function:** Copies item level permissions. As a best practice these objects should never be copied unless absolutely necessary.

- Copy form web parts
  - **Location:** Under the "Web Parts Options" tab, a checkbox labeled "Copy web parts on form pages".
  - **Function:** Copies web parts on form pages. In most cases these forms are not customized and this option will simply add checking overhead.

Multithreading and Batch Jobs

Multithreading in Content Matrix can be configured in two places.

- Multithreading within a single job.
- Multithreading in the form of multiple jobs started concurrently.

Jobs can additionally be batched together to more efficiently migrate larger sets of data.

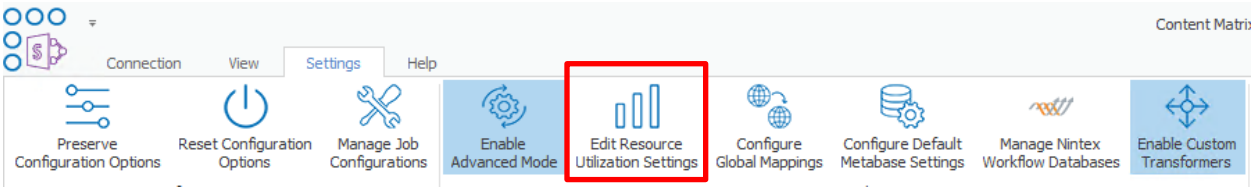
Note: This is more efficient when using Distributed Migration and multithreading.

Single Jobs

Within a single job, Content Matrix can utilize multithreading to copy site data concurrently.

Configuring Single Job Multithreading

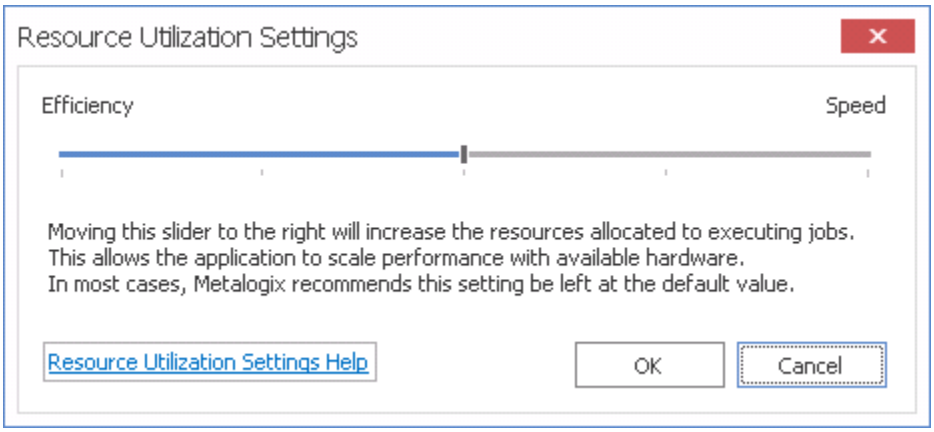
How many threads can be used simultaneously per job in Content Matrix is controlled by the "Edit Resource Utilization Settings" button within the "Settings" ribbon at the top of the Console UI



If clicked, a dialog containing a slider will appear that will allow control of the number of threads available within a single action. If the slider is moved all the way to the left, this will turn off multithreading and only allow a single thread to be used during the action.

Note: Turning off multithreading in this way can be a valuable way of troubleshooting whether multithreading is causing issues.

If the slider is left in the middle, it will allow twice the number of processors in threads to be created per action. For example, if the machine running Content Matrix has a two core CPU, four threads will be able to be used if the slider is in the below state:



Moving the slider farther to the right will allow more threads to be used, but can potentially overwhelm system resources.

Optimizing Single Job Multithreading

Content Matrix will only spawn new threads when there are multiple SharePoint site nodes to be migrated in the current location of the job. This means that in order to optimize single job multithreading, multiple sites should be migrated within a single job.

Concurrent Jobs

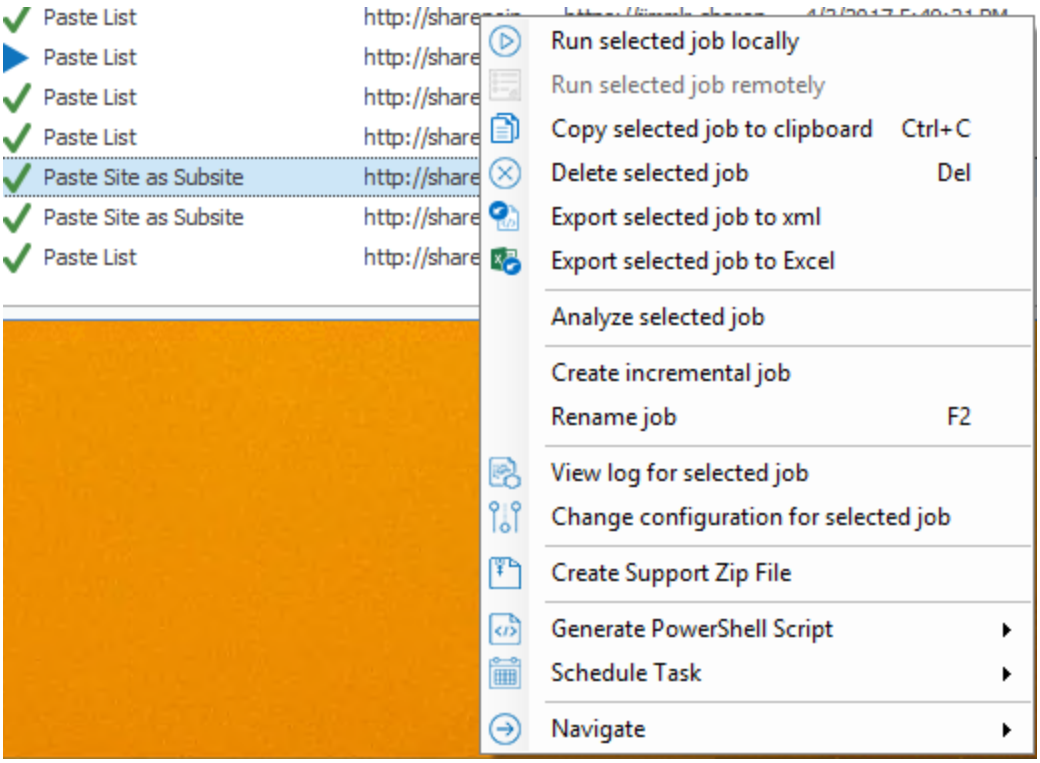
Content Matrix has the ability to concurrently run jobs.

Configuring Concurrent Jobs

To achieve this, pre-configure two different migrations. In the job listing, there should be two separate entries, as displayed below:

Paste Site as Subsite	http://sharepoin...	https://jimmlr.sharep...	4/7/2017 6:10:26 PM	Done	Lists Copied: 10, Items Copie...	Bytes Copied: 44.3 MB
Paste Site as Subsite	http://sharepoin...	https://jimmlr.sharep...	4/8/2017 12:08:34 AM	Done	Lists Copied: 10, Items Copie...	Bytes Copied: 44.2 MB

To run these jobs concurrently, for each job, right click and choose “Run selected job locally”



After both jobs have been run separately, there will be two status dialogs tracking the status of their respective jobs. You also have the ability to schedule these jobs to run at the same time later in the day by multi-selecting the jobs, right clicking and selecting 'Schedule Task'.

Note: An alternate method of running two jobs concurrently would be to generate a PowerShell script for each job and run the scripts simultaneously, or to leverage Distributed Migration.

Optimizing Concurrent Jobs

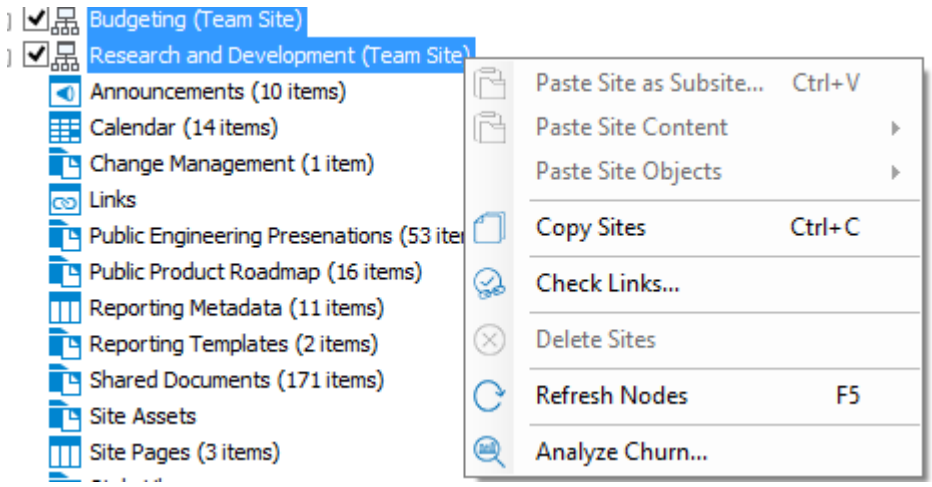
To ensure data integrity, it is important to enforce the following rules when running concurrent jobs:

- Multiple jobs should never migrate data into the same list simultaneously
- Multiple jobs should take care to avoid migrating from the same source data simultaneously

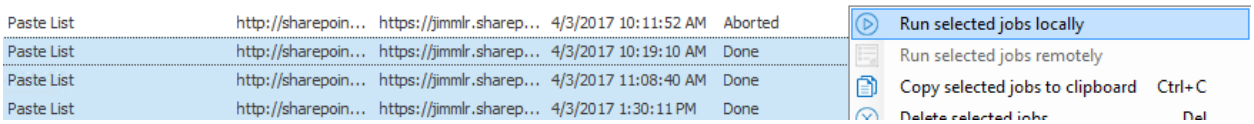
Batched Jobs

Jobs can be batched into groups, and then run. There are two primary ways of batching data into runnable jobs.

- Multi-selecting same-type nodes (sites can only be multi selected with other sites, lists with other lists, etc.)



- Multi-selecting previously configured (or run) jobs



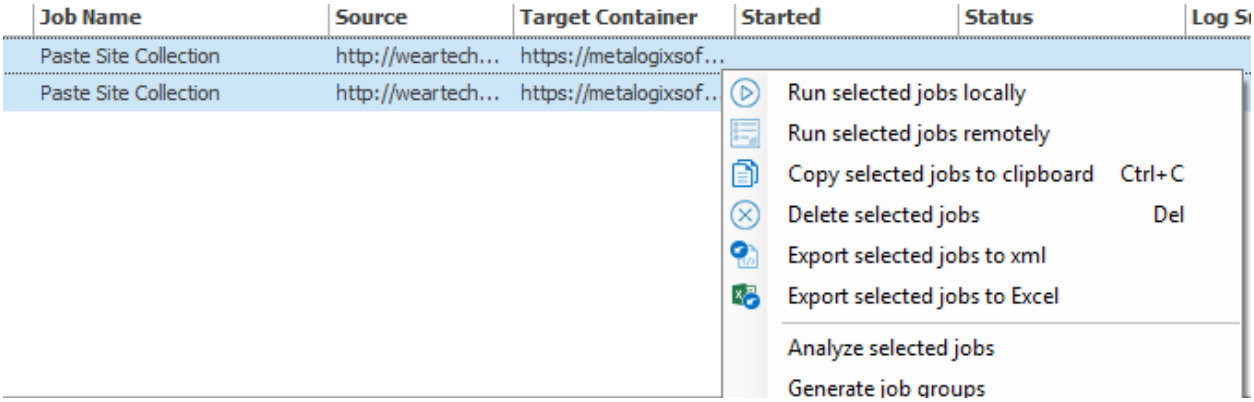
When batching data in either fashion, migration of nodes within the batch will be run sequentially. This means that if two sites are selected, they will be migrated in order. Similarly, selecting and running multiple jobs will run those jobs from top to bottom.

Batched jobs can be run concurrently with other batched jobs. Previously defined settings on multithreading and concurrent jobs also apply to all jobs run in a batch.

Distributed Jobs

As part of Distributed Migration, jobs can be run remotely across all previously configured Agents. Setup Distributed Migration as explained in the earlier section.

Select as many jobs as desired, and choose 'Run selected job remotely'. This will allocate one job per Agent VM.



PowerShell

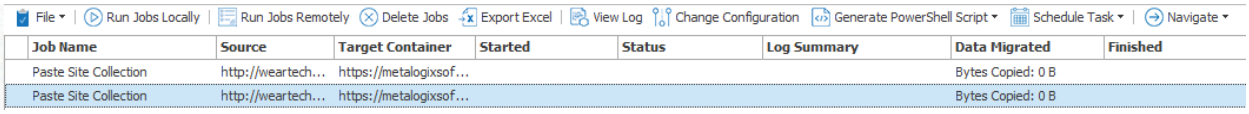
Metalogix Content Matrix has the ability to run jobs from a PowerShell context. Running jobs in this way provides some notable advantages:

- Minimizes resource consumption (no requirements to render or cache the UI)
- Separate job lists can be used for each distinct job (job lists are the storage medium for jobs, usually a SQL CE database)

Configuring PowerShell

To take advantage of PowerShell functionality in Content Matrix, simply configure a job via the UI in the normal fashion, and click the "Save" button at the bottom of the configuration dialog.

This will create a configuration entry for the job in the job listing that can be used to generate a PowerShell script. To generate a script, highlight the desired job and click the "Generate PowerShell Script" button at the top of the job listing, or right click and select Generate PowerShell:



This will create a script of the configured job that can be copied and pasted into an existing script, or saved or run as desired.

> **Note:** It is more efficient to pre-configure jobs in the UI instead of trying to write them initially in PowerShell. This will allow configuration of desired job functionality without having to learn all of the PowerShell parameters for the Metalogix actions.

Optimizing PowerShell

Optimization of Metalogix PowerShell cmdlets:

- Use the –quiet parameter
  - This parameter will disable all visual logging to the PowerShell console (all logs are still tracked in the job listing), and will increase the speed of the migration significantly.

- Use the –joblist parameter to specify multiple job listings
  - This parameter will specify which job list to write log entries to when the PowerShell job is running. Providing multiple job lists for logging purposes will help to spread load and decrease bottlenecks during the migration.

Note: You can configure PowerShell to leverage Distributed Migration. Please see "PowerShell Distributed Migration" for appropriate usage.

StoragePoint and Content Matrix

Content Matrix, when migrating to an environment with StoragePoint installed, can provide significant performance increases during migration.

Configuring StoragePoint and Content Matrix

How to install and configure StoragePoint is outside the scope of this document. All subsequent instructions assume that StoragePoint has been installed and configured.

Note: If the StoragePoint options are disabled, please ensure that the target location has StoragePoint correctly configured.

StoragePoint and Content Matrix have one primary method of interaction

- Sideloading
  - **Location:** In the configuration dialog, under the "StoragePoint Options" tab, the radio button labeled "Add documents directly to StoragePoint if an endpoint is configured".
  - **Function:** Sideloading will automatically externalize all BLOB data during the migration without it ever reaching SQL. In this case BLOB data will still transfer over the network.
  - **Benefit:** With sideloading configured, migration speed can be improved by a factor of up to 50 percent.

Next Steps

Now that you have reviewed the various migration optimization options you are ready to start your migrations with Content Matrix Console.

For more information on the available features of Content Matrix please see the Help: https://www.metalogix.com/documentation.