



Magento®
An Adobe Company

Magento 2 Migration Technical Overview

Overview of data migration from Magento 1.x to Magento 2

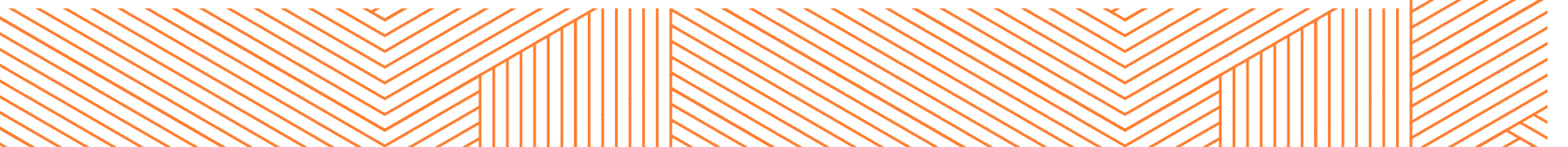


Table of Contents

Introduction	3
Best Practices and Benchmarking	5
Creating a Migration Plan	6
How Migration Works	10
Database Configuration and Mapping Files	12
Summary	15



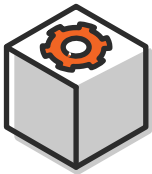
Introduction

This paper shares details on migrating from Magento 1.x to Magento 2. **A complete migration to Magento 2 involves four components: data, extensions, themes, and customizations.**



Data

The Magento 2 Data Migration Tool is available to help you efficiently port key data to Magento 2, such as product, customer and order data, store configurations, and promotions. This paper provides information on the tool and best practices for using it to migrate your data.



Extensions

The most widely used extensions for Magento 1 have been updated and vetted for Magento 2 by the Magento development community, and they are available in Magento Marketplace. More information on developing extensions for Magento 2 can be found in the [Magento 2 Extension Developer Guide](#) in [devdocs.magento.com](#).



Themes and customizations

Magento 2 uses several new approaches and technologies that give merchants an unmatched ability to create innovative shopping experiences and deliver performance at scale. To take advantage of these innovations, developers will need to make changes to their themes and customizations. Documentation for creating Magento 2 themes, layouts, and customizations is available in [devdocs.magento.com](#).

Just like an upgrade between 1.x versions (for example, from Magento Commerce 1.12 to Magento Commerce 1.14), the level of effort to migrate from Magento 1.x to Magento 2 depends upon how you have built your site and its level of customization.

High-Level Data Migration Roadmap

As mentioned previously, the primary focus of this paper is on the data migration process between Magento 1.x and Magento 2. To assist you with your migration, Magento has created the Data Migration Tool, a data migration command-line interface (CLI) tool with auxiliary verification, progress tracking, logging, and testing functions that allows direct data copying from a Magento 1.x database to a Magento 2 database. The tool efficiently and reliably migrates all core Magento data entities and any custom data entities that do not change between versions. The Migration Tool supports the following Magento versions:

- Magento Open Source (formerly Community Edition) 1.6, 1.7, 1.8, and 1.9
- Magento Commerce (formerly Enterprise Edition) 1.11, 1.12, 1.13, and 1.14

Architectural Overview

After considering several alternatives, we found the most efficient way to migrate is to use extract, transform, and load (ETL) because it migrates almost everything as quickly as possible while incurring the least amount of downtime of all options considered. To help migrate customized data, we use declarative language in our database mapping files. We also use migration steps to make it easier to write complex code for your custom needs. Mapping files and migration steps are discussed later in this paper [here](#).



Best Practices and Benchmarking

Below are some basic best practice approaches to speed up and simplify your migration, and guidance on how much time you can expect migration to require.

Best Practices and Recommendations

- **Migrate data** from a replicated Magento 1.x database instance
- **Remove outdated and redundant data** from your Magento 1.x database (for example, you could remove logs, order quotes, recently viewed or compared products, visitors, event-specific categories, promotional rules, and so on)
- **Stop all administrative activity** on both Magento 1.x and Magento 2 during your migration test runs and during the actual data migration
- **For Magento Commerce, archive all orders** including those currently unarchived

Benchmarking Estimates

Migration Tested on Systems:

- **Environment:** Virtual Box VM, CentOS 6, 2.5Gb RAM, CPU 1 core 2.6GHz
- **Database** had 177k products, 355k orders, 214k customers

Performance Results:

- **Settings migration time:** ~10 mins
- **Data migration time:** ~9 hrs (all data except URL Rewrites, ~85% of total data)
- **Site downtime estimate:** A few minutes to reindex and change DNS settings. Additional time required to “warm up” the page cache.



Creating a Migration Plan

Successful migrations are the result of thorough planning and testing. Use the following guidelines to get started on your migration plan.

Preconditions: Before You Run the Migration Tool

- Check that the data migration tool has a network connection to the Magento 1.x and Magento 2 databases
- Open ports in your firewall so the migration tool can communicate with the databases and so the databases can communicate with each other
- Disable Magento 2 cron jobs
- Set up the configuration and mapping files as discussed later in this paper.

Step 1: Review Your Current Site

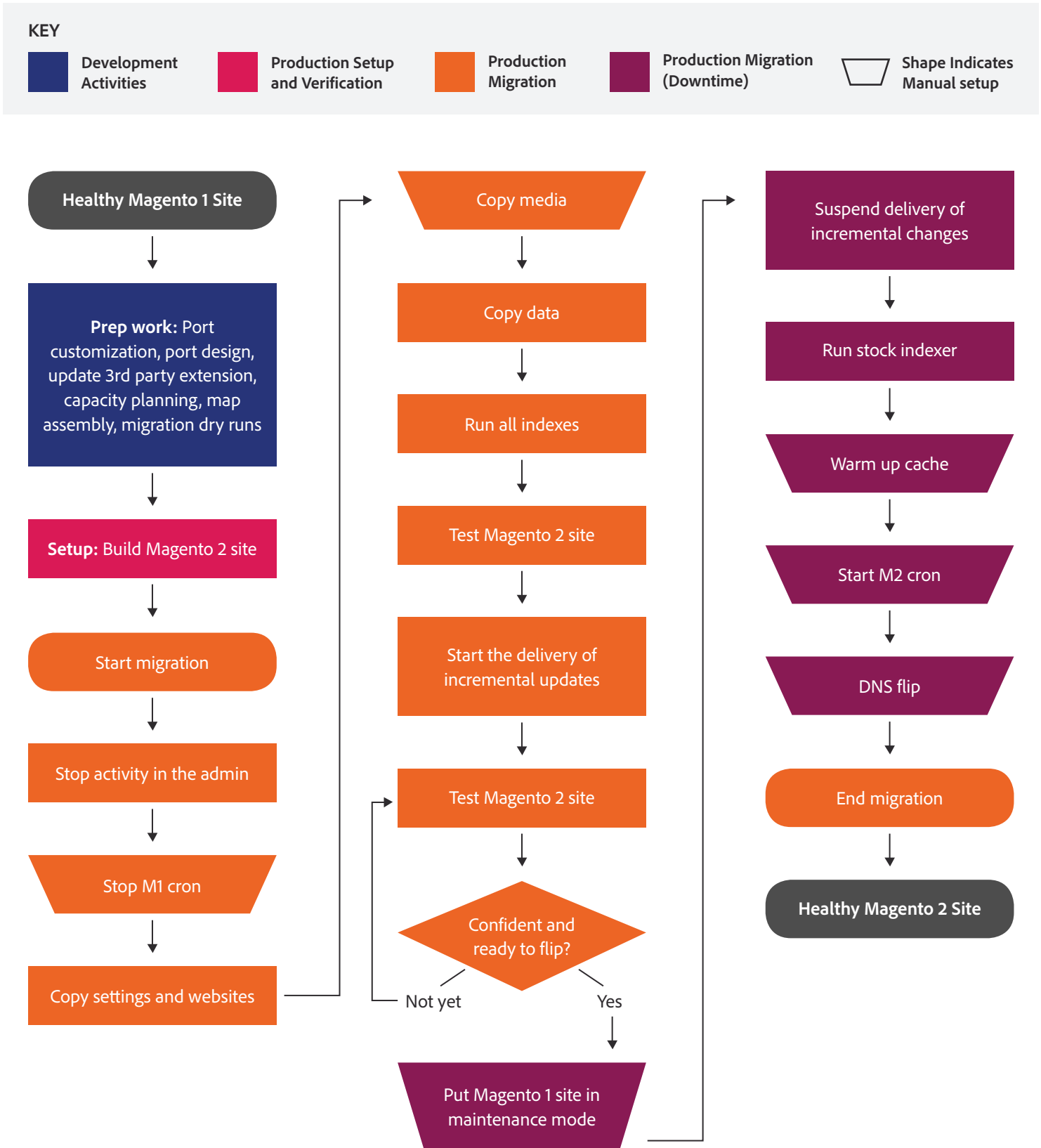
- What extensions have you installed?
- Have you identified whether you require all of these extensions on your new site? (Native Magento 2 capabilities may allow you to eliminate many extensions and/or customizations)
- For necessary extensions, identify the Magento 2 version in the Magento Marketplace (Also check with your extension providers)
- What database assets from your extensions do you want to migrate?

Step 2: Capacity Planning

- Consider whether your new site needs to be designed with more hardware or a more advanced topology. This is a great time to make more significant changes to prepare for accelerated growth.



The following figure shows the migration flow that is discussed in the remainder of this section.



Step 3: Build and Test Magento 2

- **Set up a Magento 2 system** that meets our [system requirements](#) (found on devdocs.magento.com) and that uses a topology and design that at minimum matches your existing Magento 1.x system
- **Reduce network latency** between your Magento 1.x and Magento 2 systems as much as possible to improve migration performance
- **To provide redundancy in the event of unexpected issues**, we advise you replicate your Magento 1.x database. Use this replicated Magento 1.x database as the source data for your migration.

Step 4: Start Your Migration

1. **Stop all activity** in the Magento 1.x Admin Panel (except for order management and shipping). Activity cannot resume until your migration is complete.
2. **Stop all Magento 1.x** cron jobs
3. **Use the migration tool** to migrate settings and websites
4. **Copy your Magento 1.x media assets** to Magento 2 (You must copy these manually from `<your Magento 1 install dir>/media` directory to `<your Magento 2 install dir>/pub/media` directory.)
5. **Use the migration tool** to bulk copy your Magento 1.x database to your Magento 2 database. If some of your extensions have data you want to migrate, you may need to use the mapping files provided with the migration tool.
6. **Reindex all Magento 2 indexers**. For details, see the [Configuration Guide](#) in devdoc.magento.com
7. **Thoroughly test your Magento 2 site**. Make sure you can place orders using all configured payment processors.

Step 5: Incremental Updates

Now that you've migrated the bulk of your data, you must incrementally capture data updates (such as new or updated customers, inventory changes, and orders) from Magento 1.x to Magento 2.

- **Start the incremental migration**; updates run continually. You can stop the updates at any time by pressing Control+C.
- **Thoroughly test your Magento 2 site** during this time so you can catch any issues as soon as possible. If you find issues, press Control+C to stop incremental migration and start it again after the issues are resolved.



Step 6: Go Live

1. Put your Magento 1 system in maintenance mode (**downtime starts**).
2. Press Control+C in the migration tool command window to stop incremental updates.
3. Start your Magento 2 cron jobs.
4. In your Magento 2 system, reindex all indexers. For more information, see the Configuration Guide.
5. Using a tool of your choice, hit pages in your Magento 2 system to cache pages in advance of customers using your storefront.
6. Perform any final verification of your Magento 2 site.
7. Change DNS, load balancers, and so on to point to new production hardware (**downtime ends**).
8. Resume activity in the Magento Admin.



How Migration Works

This section previews your migration experience by providing a high-level migration tool overview.

Terms Used

This document uses the following terminology to discuss the migration tool:

- **Step:** A unique migration task that must be executed in a prescribed order
- **Map:** A set of rules that describes connections between Magento 1.x and Magento 2 data structures
- **Mode:** Represented by a separate migration tool command, defines the basic mode of operation as:
 - Settings: Migrates the system configuration and website-related settings
 - Data: Migrates database assets in bulk
 - Delta: Migrates incremental changes (for example, orders and inventory)

Conceptual Overview

Each mode has several steps; steps might include data migration and integrity checking. Mapping files enable you to specify how custom database data is moved (For example, if you have custom data you can move it as-is or you can rename tables and fields in the Magento 2 database). The declarative language used in the mapping files makes it as easy as possible for you to map your custom data.

All Magento 1.x core tables are automatically migrated to Magento 2 with no action required; you need to map only your custom database tables. In addition, any custom tables that do not change between Magento 1.x and Magento 2 are migrated automatically with no action required.

Mapping files are discussed in more detail later in this paper [here](#).



What Needs to be Manually Migrated

Extensions and custom code

To take advantage of the architectural advances in Magento 2, developers need to manually port custom code and extensions to Magento 2.

Layouts

Layout updates implemented in Magento 1.x cannot be used in Magento 2. You must migrate the following manually:

- XML in CMS category pages in the Magento Admin
- Layout updates specified in widget instances

Web services credentials

You must manually create credentials for SOAP, XML-RPC, and REST in Magento 2.

Media on the file system

Media assets include, for example, images for products, categories, WYSIWYG editor, and so on. You must copy these manually from the `<your Magento 1.x install dir>/media` directory to the `<your Magento 2 install dir>/pub/media` directory.

Storefront design

Cascading Stylesheets (CSS), JavaScript, templates, and XML layouts are implemented differently in Magento 2 and must be migrated manually.

Data that will not be migrated

For the reasons stated, the tool will not migrate the following data:

- **Indexed data:** You should perform a full reindex before you enable your Magento 2 instance on the production server
- **Google Shopping:** Google Shopping shipped in some older Magento 1.x versions but has since been removed
- **Data that is not supported in Magento 2.0:** Poll, tag, staging modules, and recurring profiles are not currently supported in Magento 2



Database Configuration and Mapping Files

The migration tool is controlled by a configuration file that defines data migration rules. Using a configuration file and mapping files, you can:

- Automatically migrate tables and data for all tables that did not change between Magento 1.x and Magento 2.
- Migrate Magento 1.x tables that have different names in Magento 2 (for example, the Magento 1.x `sales_flat_shipment_comment` table migrates to the Magento 2 `sales_shipment_comment` table)
- Modify column names (for example, the `cms_page->root_template` Magento 1.x column migrates to `cms_page->page_layout` in Magento 2)
- Modify field values (for example, the Magento 1.x `cms_page->root_template->two_columns_right` field migrates to `cms_page->page_layout->2columns-right` in Magento 2)
- Ignore tables or columns that you don't want to migrate

Note: The preceding list uses examples of tables, columns, and fields provided with Magento 1.x and Magento 2. You can perform the same tasks for all of your custom database tables as well.



Configuration File

The configuration file provides a list of steps, options, and connection information about your Magento 1.x and Magento 2 databases. **Sample:**

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="../../../config.xsd">
  <steps mode="settings">
    <step title="Settings Step">
      <integrity>Migration\Step\Settings\Integrity</integrity>
      <data>Migration\Step\Settings\Data</data>
    </step>
  </steps>
  <steps mode="data">
    <step title="Data Integrity Step">
      <integrity>Migration\Step\DataIntegrity\Integrity</integrity>
    </step>
    <step title="Map Step">
      <integrity>Migration\Step\Map\Integrity</integrity>
      <data>Migration\Step\Map\Data</data>
      <volume>Migration\Step\Map\Volume</volume>
    </step>
    ...
  </steps>
  <steps mode="delta">
    <step title="Customer Attributes Step">
      <delta>Migration\Step\Customer\Delta</delta>
      <volume>Migration\Step\Customer\Volume</volume>
    </step>
    ...
  </steps>
  <source>
    <database host="localhost" name="magento1" user="root" />
  </source>
  <destination>
    <database host="localhost" name="magento2" user="root" />
  </destination>
  <options>
    <map_file>etc/opensource-to-opensource/1.9.4.0/map.xml.dist</map_file>
    <eav_map_file>etc/opensource-to-opensource/map-eav.xml.dist</eav_map_file>
    ...
    <source_prefix />
    <crypt_key />
  </options>
</config>
```



Mapping Files

The following is a sample mapping file that:

- Transforms all Magento 1.x database tables and fields that have Magento 2 equivalents
- Renames fields from the Magento 1.x database in the Magento 2 database
- Ignores fields

```
<?xml version="1.0" encoding="UTF-8"?>
<map xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="../../map.xsd">
  <source>
    <document_rules>
      <ignore>
        <document>googleshopping_items</document>
      </ignore>
      <ignore>
        <document>googleshopping_types</document>
      </ignore>
      <rename>
        <document>sales_flat_invoice</document>
        <to>sales_invoice</to>
      </rename>
      <rename>
        <document>sales_flat_invoice_comment</document>
        <to>sales_invoice_comment</to>
      </rename>
      ...
    </document_rules>
    <field_rules>
      <move>
        <field>cms_page.root_template</field>
        <to>cms_page.page_layout</to>
      </move>
      <transform>
        <field>cms_block.content</field>
        <handler class="\Migration\Handler\Placeholder" />
      </transform>
      <ignore>
        <field>admin_rule.role_type</field>
      </ignore>
      ...
    </field_rules>
  </source>
  <destination>
    <document_rules>
      <ignore>
        <document>customer_entity</document>
      </ignore>
      ...
    </document_rules>
    <field_rules>
      <transform>
        <field>oauth_token.user_type</field>
        <handler class="\Migration\Handler\SetValue">
          <param name="value" value="1" />
        </handler>
      </transform>
      ...
      <ignore>
        <field>sales_invoice.send_email</field>
      </ignore>
    </field_rules>
  </destination>
</map>
```



Summary

With innovative technologies and a modern architecture, migrating to Magento 2 opens up dramatic growth potential for your business. Data migration is a key component of the overall Magento 2 migration process, and the Data Migration Tool should give you confidence that your data will be transferred efficiently and reliably, with minimal downtime, so that your new site will be ready for action.

Don't forget we have the following resources available for you to help with your migration:

Magento Support

[Get Support](#)

Online Magento Migration Guide and the Data Migration Tool

[Online Migration Guide](#)

Documentation for creating Magento 2 themes

[Read Theme Documentation](#)

Documentation for creating Magento 2 layouts

[Read Layout Documentation](#)

Magento 2 Extensions Developer Guide

[Online Developer Guide](#)





About Magento

Magento, an Adobe company, is a leading provider of commerce innovation to merchants and brands across B2C and B2B industries and was named a leader in the 2018 Gartner Magic Quadrant for Digital Commerce. In addition to its flagship digital commerce platform, Magento boasts a strong portfolio of omnichannel solutions that empower merchants to successfully integrate digital and physical shopping experiences. Magento is the #1 provider to the Internet Retailer Top 1000, the B2B 300, and the Top 500 Guides for Europe and Latin America. Magento is supported by a vast global network of solution and technology partners, a highly active global developer community, and the Magento Marketplace—the largest eCommerce marketplace for downloadable extensions.

More information can be found at www.magento.com