# Correspondence Management Solution Implementation Guide

## Contents

**Objective** ........................................................................................................................................................................... 2

**Background** ....................................................................................................................................................................... 2

**Scope** ................................................................................................................................................................................. 2

**CM Solution Implementation Considerations** ................................................................................................................... 2

  **Installation and Configuration considerations** .............................................................................................................. 2

  **Hardware and system considerations** ........................................................................................................................... 2

  **System Capacity and Sizing Planning considerations** ...................................................................................................... 3

  **Deployment Topology Considerations** .......................................................................................................................... 4

**CM Solution implementation persona** .............................................................................................................................. 5

**Splitting Logic** ................................................................................................................................................................ 6

**Customization** ............................................................................................................................................................... 6

**Asset Naming** ................................................................................................................................................................ 6

**Asset Modifications and Asset Movement** ....................................................................................................................... 7

**Backup and Restore** .......................................................................................................................................................... 7

**Data purging** ...................................................................................................................................................................... 8

**Patching** ............................................................................................................................................................................. 8

**Author and Publish Considerations** ............................................................................................................................... 8

**Form Designing Considerations** ....................................................................................................................................... 9

**Data Dictionary Design Considerations** ............................................................................................................................ 9

**Text Module Design Considerations** ............................................................................................................................... 9

**Condition Module Design Considerations** ........................................................................................................................ 10

**Letter Template Design Considerations** .......................................................................................................................... 10

**Placeholder Usage Considerations** .................................................................................................................................. 10

**Anatomy of a letter** .......................................................................................................................................................... 11
Objective

This document is intended to provide best practices and guidance to new Correspondence Management (CM) customers for successful implementation of CM Solution.

Background

It has been observed in various CM Solution implementations that some aspects overlooked while designing the system come back later to haunt us. For example, most of the early CM Solution implementations did not consider asset movement across various systems [namely development (DEV), testing/quality assurance (QA), staging/performance (PERF), production (PROD)]. They went ahead with the full Adobe Content Package (ACP) export and import process, which led to lot of issues being raised by customers and sometimes these were escalated. In some cases, customers started creating the assets without defining the naming convention and ended up creating unusable and unmanageable number of placeholder variables to be filled while creating the letter in Create Correspondence UI. There are many such examples and it is evident that a best practices guide, documenting important CM activities is very much required to guide the implementation.

Scope

This document will not delve into the feature level (how-to) details as they are part of CM documentation. Instead, it will focus on system environment, deployment, configuration, asset change management, asset movement across various systems and stability.

CM Solution Implementation Considerations

Installation and Configuration considerations

- Ensure that hardware meets the requirement
- Ensure sufficient disk space is available on the system
  - Keep an eye on growing GDS/temp files so that they do not fill up the disk space
- Keep the system up-to-date by applying latest updates (patches/feature packs)
- Consult the LiveCycle install and configuration guide for detailed steps
  - LiveCycle and CM Solution are installed with default configurations but those need to be tweaked to best suit the requirements/needs
- Maintain a configuration document and capture all the configuration changes that have been made on the system
- Keep all the systems namely DEV, QA, PERF, PROD etc. in sync w.r.t. LiveCycle and CM Solution configuration settings

Hardware and system considerations

It is important to keep DEV, QA and PROD separate so that activities on one environment do not impact the other environments. It is recommended to have separate systems for different purposes:

- DEV: For customization, LC process development and asset authoring [can be a non-clustered system]
- QA: For validation of acceptance tests [can be a system exactly similar to DEV i.e. non-clustered system]
- PERF: Should be exactly same as PROD [can also be used for performance testing with expected load that would be on PROD]
- PROD: It is recommended to use a clustered system for high availability and scalability

Once the required systems are in place, it is very important to have detailed documented procedure to ensure that they all have exact same configuration and settings. System administrators should perform random checks across the systems periodically to keep the configuration and settings in synch. It will be better to have a change control
process in place for making any kind of configuration and settings changes. System administrators must have detailed documentation of all the settings and configuration done on each setup. Chances are that the configurations and patches running on DEV/QA will vary that from PERF/PROM, they should remain in sync for important activities like letter movement.

It is important to note that at least 2 systems have exact same configuration and settings (QA and PROM or PERF and PROM) else different results might be seen (on PROM environment) which may be considered as issues on PROM. Doing this would ensure that the solution being run on PROM has been tested/run on at least one environment, which is exactly similar to PROM.

It is extremely important to test before pushing ANYTHING on to PROM. It is recommended that a defined set of tests are available (acting as gatekeeper for promotion to PROM) for important workflows/use cases and they should all pass.

**System Capacity and Sizing Planning considerations**

Depending upon the use cases for your business, expected user load on the system and data requirement, you need to get the required capacity for smooth functioning. Some of the questions that you need to answer in order to do system capacity and sizing planning:

- How many users will be authoring/creating assets on a daily basis?
- How many users will be modifying the existing assets on a daily basis?
- How many assets will get created on a daily basis?
- How many assets will get modified on the daily basis?
- Is the predominant use case is of non-interactive (system generated correspondence) or is it interactive (where agent is using Create Correspondence UI)
  - If it is interactive then:
    - How many users will be generating the correspondences on a daily basis?
    - How many correspondences will be generated daily?
- In case of both interactive and non-interactive, what is the percentage of interactive correspondence generation?
- What is the distribution (percentage) of the complexity of letters generated (simple, medium and complex)?

Based on the answers to the above questions and expected performance levels, decide on the deployment topology and need for clusters. It is important to run the workflows with similar or little higher than expected load on PERF which is exact replica of PROM. Ensure that the results meet the expectations.
Deployment Topology Considerations

The above diagram is a representation of the recommended deployment topology for the CM Solution. Such a deployment topology addresses the familiar need to isolate DEV, QA, and PROD systems. Of course, each of these environments, typically the PROD ones, may be clustered to several instances, depending on the scale of a project.

The solution always resides as two instances -

- Authoring Instance: This is where all assets are authored and managed, including preview of correspondence templates. This is meant to be used by content/template authors.
- Publish Instance: This is where the final correspondence is to be created. Assets authored (and once finalized) on the Authoring Instance are published to the Publish Instance via a simple click of a button (on the Manage Assets interface). This is meant to be used by end users or agents.

Authoring and Publish instances would share a common LiveCycle Server, that the solution uses for various services (Forms/Output/ProcessManagement, etc.) on that stack. The Author and Publish being separate instances, have their own JCR repository (CRX: Content Repository Extreme).

Between the different Authoring systems (like DEV, QA, PERF etc.), any content, design templates, or customized application functionality is always transferred by exporting/importing assets (using the Manage Assets interface).

Depending upon load on the system and availability considerations, following are main topology considerations

1. Number of users working in parallel on application on both Authoring and Publish side will determine the number of instances required.
2. Static content stored on CQ instances (Non CM assets) will determine need of Dispatcher caching.
3. Amount of letter rendition requests on LiveCycle server will determine number of LiveCycle Servers.
4. Security considerations. For example, Author instance can be in internal network, publish instance can be in DMZ.

**Author and Publish Scaling**

- Based Upon number of users and availability requirements, Author and Publish instances are repository clustered.
- The data of an Author Instance is synced with another Author Instance over the network. This is known as Repository Clustering.
- Publish instances which are accessed by end user for creating correspondences can be scaled and they need not talk to each other.
- It is Author instance's job to keep all connected publish instances in sync.
- Adding a new Author instance or new Publish instance can be done by just restoring backup of existing running instance.

**Author/Publish configuration**

Refer to the Author Publish configuration in the Installation Guide to see steps involved in setting up the two instances, both on a single physical server as well as different physical servers.

**CM Solution implementation persona**

For various different operations in CM Solution, administrator needs to assign appropriate roles.

CM sample includes the following sample users who are expected to participate in the activities leading to generation of interactive customer communication:

<table>
<thead>
<tr>
<th>Sample User Name</th>
<th>Groups Assigned</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Todd Goldman (tgoldman)</td>
<td>CM Administrator</td>
<td>This user is the general system administrator. This role enables the user to modify all assets. This role also lets define the categories.</td>
</tr>
<tr>
<td>Heather Douglas (hdouglas)</td>
<td>CM Subject Matter Expert</td>
<td>This persona has the role enabling them to Create, Retrieve, Update, and Delete texts and images</td>
</tr>
<tr>
<td>Caleb Lopez (clopez)</td>
<td>CM Application Specialist</td>
<td>This user defines the letter template by judicious usage of the text, picture, condition, list objects. With this role, the user can Create, Retrieve, Update, and Delete the letter templates, layouts, lists, conditions, texts, and images</td>
</tr>
<tr>
<td>Gloria Rios (grios)</td>
<td>CM User</td>
<td>A customer interfacing employee such as a claims adjuster or case worker who uses the letter template defined by the business user to produce the letter communication to deliver to the customer</td>
</tr>
<tr>
<td>Jocelyn Robinson (jrobinson)</td>
<td>CM Form Designer</td>
<td>This user has the skills to design form layouts using Adobe® LiveCycle® Designer. Having equipped with the necessary know-how to design form layouts for use in CM. This user uses Designer, and designs the XML Data Packet (XDP) templates, which would serve as the boilerplate for the letter.</td>
</tr>
<tr>
<td>Frank Kricfalusi (fkricfalusi)</td>
<td>CM Developer</td>
<td>This user has the knowledge about XML Schema Definition (XSD) and data modelling concepts and is responsible for creation and maintenance of Data</td>
</tr>
</tbody>
</table>
All the user and group information and their relations are captured in a configuration file (template-correspondencemanagement-pkg) which is read during bootstrap.

### Splitting Logic

All the content for CM Solution is stored in the embedded CRX repository. It is seen that as the repository grows, it becomes difficult to navigate the repository if all the content is stored under the root node. Also, having thousands of nodes under the single node impacts the overall performance of the system. CM Solution provide a detailed documentation on how to apply the content splitting logic so content gets split, making the navigation easier and also helps boost the performance. Detailed documentation on splitting logic is available under “Implementation Overview” in CM Developer Guide.

**Important note:** Splitting logic should be applied while setting up the systems before creating any assets on the system.

### Customization

Customization is enabled to a great extent in CM Solution. Depending upon the needs of the project, various user interfaces and functionality can be customized. Following customization stories are documented under “Customizing your Correspondence Management solution” in CM developer Guide in detail:

- Rebranding CM
- Customizing the Create Correspondence user interface
- Customizing the Manage Assets user interface
- Customizing Editors
- Managing data captured in Asset Composer
- Overriding Building Block Services
- Adding handlers for operations around CRUD
- Adding new letters and customers to the Create Correspondence interface
- Reader Extending PDF preview
- Adding SaveAsDraft functionality
- Splitting Data in CRX
- Creating a portfolio using the API
- Upload supporting document
- Previewing and Reloading a Correspondence
- Creating non-interactive (System Generated) Letters
- Enabling offline data capture for a system generated correspondence

### Asset Naming

It is important to name the assets in a meaningful way so that searching and editing of assets as well of using them while creating List, Conditions and Letter Templates become simpler. Some of the advantages of having a naming scheme are:

1. Helps in getting better results in searches
2. Makes it easier to find and use the assets
3. Makes it easier to find name conflicts for PH variables
4. It is better to follow some convention for naming assets based on asset type or any other convention and define a proper breakdown of correspondence before actually creating it and keep this information updated on any subsequent updates.
5. Do not use lengthy names for assets, or DDEs or PlaceHolder variables, though CM does not restrict on size of names, it’s better not to use lengthy names so that wherever the name of assets are displayed it does not get truncated
Asset Modifications and Asset Movement

It is expected that asset creation and modification happen in a controlled environment. Editing of assets on any environment except on DEV is not recommended. All asset creation/ modification should happen only on DEV and the required assets can be exported & imported periodically on to all other systems.

In order to export, all the assets need to be in published state/ready to publish state (which indicates that the asset development/ modification is complete) so that they can be exported. Required assets can be exported from DEV and imported on QA and once all the required workflows are validated, the same exported zip can be imported on PERF and then on to PROD. It is recommended to keep a history of all the assets that are imported on a specific system. It is also recommended to move assets across systems (DEV to QA, PERF and PROD) in small chunks and more frequently rather than moving large number of assets using export all functionality. While modifying any asset, one can view the dependencies as well as export them to an xml. How to move assets across systems is documented under “Moving assets viewing dependencies” in CM Solution Guide.

It is highly recommended to use Asset Integrity Checker tool before exporting the assets from the source system. If there are any asset integrity issues reported by the tool, follow the recommendation to fix the issues and rerun the asset integrity checker tool. Similarly, after importing the assets on to the destination system, it is highly recommended to run the asset integrity checker tool to find out the issues and fix them before further use.

NOTE: “Export All” functionality needs to be used only once if the assets are imported on to a clean system without any existing assets on it.

NOTE: Asset Integrity checker tool should be run periodically (daily or weekly) on DEV where authors are actively creating or modifying the assets.

One can transport the assets in 2 ways, all assets or selectively. It is highly recommended to use the selective export of the assets from the source system and importing them on the target system.

**Few points to consider while using the selective export and import:**

- Only Published or ReadyToPublish assets can be exported.
- On exporting an asset, all the referred assets are also exported along with it. This means, if you are exporting a letter, you need not worry about any of the other assets used in the letter for export and import as all the referred assets will automatically get exported except for Data Dictionary.
- When exporting assets, its related Data Dictionary is not exported automatically. Export the Data Dictionary separately.
- At the time of import if an asset exists in the system with the same name, the existing asset is updated. If there is no asset, a new asset is created.
- All the assets will be in ReadyToPublish state after import.

Backup and Restore

Taking regular backup of the system is critical to avoid data and productivity loss.

Deciding on the frequency of backup:

- On DEV where assets are actively created or modified, it is highly recommended to back up the repository every day, so that in case of any unforeseen situations the system can be brought to previous working state without much loss of productivity.
- On all other systems i.e. QA, PERF and PROD, It is recommended to take a backup before importing any assets and take a backup periodically (daily backup is recommended)

Detailed backup and restore procedure for CM Solution is well documented.
Data purging

Over a period of time multiple versions of the same asset would come in existence, business owner needs to determine the assets that need to be retained and assets which need to be purged. Depending on business decision all assets prior any specific date can be purged. This activity need to be done periodically.

If there are any long lived orchestrations in use, then jobs and processes should also be purged periodically such that growth of large amount of data in database can be avoided. Detailed process purge tool documentation is available under Purging Process Data in LC Admin Help and documentation for purge records from the Job Manager database is available under Purge records from the Job Manager database in LC Admin Help.

Patching

It’s recommended to apply any new available patch or feature pack at the earliest possible time, as the new versions will always have bug fixes and enhancements. It is recommended to apply patches quickly on DEV and keep a buffer of sometime (like two weeks or a month) for adequate testing before they are moved to other systems like QA, PERF and PROD. It is recommended to have a set of acceptance tests that need to be run on PERF before any assets are moved to PROD or before applying any patch on PROD.

Before applying any patch, one needs to look at the existing customizations and impact on them. To understand the impact, read the release notes carefully and also look at the change logs.

Author and Publish Considerations

A CM deployment usually consists of multiple environments, used for different purposes on different levels. PROD often consists of at least one author instance and one publish instance. Depending on the scale of a project, PROD may consist of several author and/or publish instances. Additionally, separate DEV and QA may also consist of both author and publish instances, mirroring the PROD to a varying extent.

While different instances in different environments are all installations of the same CM software installed in different places in the overall system infrastructure, they differ mainly in the way they are configured. For example, it is that configuration which determines whether a CM instance behaves as an author instance or a publish instance.

Author instances are usually located behind the internal firewall. This is the environment where all the authoring tasks are performed, like, text, layout, Data Dictionary (DD) and letter template creation.

Content that was set as 'ReadyToPublish' is packaged and placed in the author environment's replication queue. The replication process then transports that content to the publish environment. An asset in ReadyToPublish state can be published along with other dependent assets without explicitly publishing this asset.

A publish environment may or may not be behind the firewall. This instance is accessible only to the agents generating correspondences.

Few points to consider while working with CM author and Publish instances:

- Never mark a work in progress asset as ready to publish.
- Once asset development is complete, mark it as ready to publish, so that it does not prevent publishing of other dependent assets.
- On publishing an asset all the referred (child) assets are published along with it. Dependent (parent) assets are published only if they have already been published at least once.
- Once an asset is published, it can never be deleted. So publish an asset only when you are sure.
- Data Dictionaries are not published automatically along with dependent assets. These are required to be published separately.
- If an asset is never published and there is no dependency on this asset, revert operation deletes the asset.
- If an asset is published, revert operation rolls back the asset to its last published version. All the dependent assets will be updated.

For more details on how to configure and use author and publish instances, please have a look at Post Deployment Activities in the document Installing and Configuring CM Solution.
Form Designing Considerations

You can use your existing layouts (XDPs) or create new one based on the requirement. You need to use LiveCycle Designer for creating or updating the XDPs.

Few points to consider while creating / updating a XDP that is in use:

- Updating the XDP in LiveCycle Designer does not automatically update the layout in CM. You need to update the layout using CM layout editor.
- Analyse the layout and determine the content that is being repeated across all pages; usually page header and footer fit into this category. This content should be placed on master pages of layout. The remaining content should go to body pages of the layout.
- By default, all subforms that are empty of content are considered target areas. If your layout contains an empty subform that is not considered a target area, name the subforms with a "_int" (internal) suffix. A target area should have no binding and should not be on master page.
- By default, all fields are considered relatable to various other data sources. Mark any field that should be ignored from relationship authoring by giving it a name with an "int" (meaning "internal") suffix (e.g. "pageCount_int"). A _relatable field should have a name and its binding should NOT be set to none.
- At the time of updating a layout or fragment layout, try to avoid changes in field and target area Scripting Object Model (SOM) expressions. This will ensure that letters using this layout or fragment layout preserve bindings in target areas and fields. In case the changes to fields and target areas are unavoidable, you need to reopen all the letter templates using it and check / update the bindings and save the letter template.
- Use a subform (target area) if you want to capture multiple module content in a top-down vertical-flow layout (multiple paragraphs or images). Your layout must handle the fact that the subform grows in height to accommodate its contents.
- Use a field if you want to capture module data or data dictionary element data into your layout's schema (because fields are bound to data) or to display module content on a master page (Remember that content in a master page cannot flow with body page content).

Data Dictionary Design Considerations

The Data Dictionary represents the underlying data model in business vocabulary, such as name, age, eligibility, and so on. It allows non-technical users to easily create correspondence, content assets, and processes using the vocabulary that they understand, without having to worry about how the actual data is sourced.

Few important points to consider while creating or updating a Data Dictionary:

- Only one DD can be used in a letter template, so all the data that you need from the back-end system should in some way map to the various Data Dictionary elements.
- You should be careful while updating an existing DD as it may impact the functioning of various letters and other assets created earlier.
- Deletion of a Data Dictionary Element (DDE) is not allowed if it is in use. This is to ensure existing letters are not negatively impacted.
- The DDE is not tightly coupled with the structure of the DD, 'binding' of each DDE helps in this. This binding can be referring to a structure separate from that of the DD.
- When marking a DDE as computed, it is important to validate it.
- When using a DDE in another computed DDE, mark the first DDE as mandatory so that the expression is always evaluated with this value.
- DDEs of type string and number can be enumerated with a fixed set of values that can be pre-populated.

Text Module Design Considerations

- Data Dictionary update is not allowed in a text module if it is included in any other asset. Assigning a new data dictionary is also not allowed if it is referred by any other modules.
- To avoid human errors, insert DDE variable, place holder variable form the data element panel. Repeat tags, links and inline conditions should be inserted from toolbar. It is recommended not to edit or copy/paste these references in the text content area.
• Only one state (protected/ unprotected) of Data Dictionary element is maintained throughout the text module. A checkbox is provided across each DDE in the Data Element Panel to set its value to protected/ unprotected.
• Font size can be specified explicitly by providing a Integer or decimal value into the font size text box. The new value of font is retained in the current session and is reset to its original set of values when the session is refreshed.
• English and French spell check is supported in the text module. By default Spell check is enabled for the language that is set in the Browser. The spell check language can also be manually set to English or French from the edit drop down.
• The agent can be restricted to choose from a specific set of values for a placeholder in Create Correspondence UI (CCR). Edit the placeholder in the data element panel. Enable the checkbox ‘allow specific values only’ and specify the list of values in the text box.
• Text alignments, text margin and line height is applicable at paragraph level.
• Data Elements panel is not available in the text editor in CCR. It is suggested not to insert or alter any DDE of place holder from CCR.

Condition Module Design Considerations
• If multiple expressions in a conditional module return true, then the data module corresponding to the first expression among the list of expressions is rendered in CCR. If a condition is required to support multiple resultant modules click on the icon ‘Condition may evaluate to multiple results’ in the condition editor.
• Always validate an expression using the validate button in expression manager to check if the expression is syntactically correct.
• The expression should always evaluate to a Boolean value (true or false) analogous to an IF condition.
• User defined local and remote functions can be inserted in an expression. The functions defined in the system are populated in the function tab in Expression manager.

Letter Template Design Considerations
Letter template design is the most important activity in correspondence generation. There are certain rules that need to be followed in order to get it right in the first go and during the updates.

Few important points to consider while designing or modifying the letter template:
• Use a consistent naming convention to avoid duplication.
• Use appropriate Data Dictionary binding to enable mapping of assets from that DD.
• Note that once you select the DD and save the letter template, then it can't be changed.
• All the assets shown in the content library to add into a letter would be filtered based on the DD. That means that you will see only the assets which do not have a DD or have the same DD as that of letter template.
• Fields should be bound to user only if that letter is to be generated by Agent. For non-interactive (system generated) Letters fields should not be bound to User.
• For mandatory / fixed content mark them preselected and required.
• Content should be marked as editable only if it requires agent modifications.
• On removing an asset (like list, or text), all the placeholder variables are automatically removed and bindings are lost, so be careful while removing any of the assets.
• While adding a new assets to the letter (like list, or text), do not forget to map the variables.

Placeholder Usage Considerations
Placeholder variables give the flexibility to agent to fill in the data at the time of generating correspondence in Create Correspondence UI. One can assign the default value and also mark them optional or mandatory. By using the placeholder variables, you can give a lot of power to the agent generating correspondence.

Few points to consider while creating and using the placeholder variables in text modules:
• Placeholder variables from various content data modules are treated as one at the letter level. Hence, naming should be carefully done to avoid getting confused between variables from two data modules. Also,
same name can be given to a variable which will have the same value for a correspondence though it would be used in multiple data modules.

- Placeholder variable with same name but of different data type will be treated as same variable.
- Placeholder variables should be carefully linked to in the data editing view. Make sure if they are linked to another data module, the resultant data module fits in well in the text.
- While linking a variable to another variable, make sure they are of the same data type.
- If a variable is linked to a field then, the field should not be linked to 'ignore', instead linked to a compatible type with the variable.

Anatomy of a letter

First step in creating a correspondence using CM Solution is to understand the various constituents of letter. Assuming that a paper based version of the correspondence is available, one needs to identify:

- Which part would be fixed text (static content) and hence needs to be part of the layout (XD of XDP) itself?
- Which part of content will have variables and from where the value of these variables come in?
  - Would they come from a back-end system and hence they should be part of Data Dictionary?
  - Or, they will be filed by the user?
  - What would be the type of the variable?
  - Whether the variable will be optional or mandatory?
- Which part of the letter would be coming as form fields?
- Which part can be embedded in the layout itself?
- Variables/fields values would be pre-populated in case of non-interactive (system generated) or need to be filled in by the agent (interactive)
- What would be conditional content etc.?

The image below shown one such example, text in blue are DDE, text highlighted in Pink is editable test.

Welcome to

**<Product Name>** Plan of Finance Corporation

**Dear**, **<Policy Holder's Title>**, **<Policy Holder's Full Name>**

On behalf of Finance Corporation, I’d like to welcome you to our vast family of esteemed customers. We are committed to providing you and your family with the most comprehensive financial protection thereby ensuring that you enjoy peace of mind and are able to live your life happily without worrying about the impact of unforeseen events.

(A product line specific paragraph to come here which describes the current state of customer base and a brief history of the plan)

This welcome kit contains valuable information about your policy, our contact information, procedures to be followed in the event of emergency and various facilities provided by us to help you make the best use of our services.

Again, we are pleased to add you to our family of satisfied customers. You are requested to keep this kit safely, so that you are able to readily access it in times of need. If you have any questions about your policy/product or require another copy of this welcome kit, please contact our Customer Relations Executive at 1-888-XXX-XXXX.

Sincerely,

John Doe

Chief Executive Officer, Finance Corporation, Inc.

Follow how to deconstruct correspondence in CM Solution Guide to understand and create correspondence templates.