EMC Centera

Advanced Design and Setup Guide

Perceptive Content Version: 7.0.x

Written by: Product Knowledge
Date: October 2014

perceptivesoftware
from Lexmark

Table of Contents

Perceptive Content and EMC Centera systems

Perceptive Software manages fixed content, digital assets of information in its final form. Perceptive Content can quickly integrate with business applications and different data storage platforms while continuing to capture, process, index, and then take actions on your digital assets. This technical paper focuses on how Perceptive Content integrates with EMC Centera where Perceptive Content is the preferred fixed content application and EMC Centera is the primary storage system.

When customers choose EMC Centera with an Perceptive Content component, Perceptive Content is the interface, input mechanism, and single-click access to each asset, capable of retrieving and returning documents in a matter of seconds. Working with the EMC Centera storage device system, Perceptive Content can capture documents and other unstructured data, and manage all of the metadata surrounding each asset. Perceptive Content uses the Centera API to copy its document objects directly from the Perceptive Content Server to the EMC Centera server.

Organizations are seeking options that can store large amounts of data (up to a petabyte) for various types of fixed content including e-mail messages, imaged documents, and industry-specific information such as medical images and engineering drawings. EMC Centera is a storage option that provides a robust redundancy option that Perceptive Software supports. It is a networked storage system and the first magnetic hard-disk based WORM data device of its kind. WORM means to Write Once, Read Many and assists Centera in assuring that each object stored is unique and unable to be erased unless specified.

By using traditional magnetic hard disks to store data, EMC Centera offers greater performance over optical disk and tape, which take considerably longer to access the stored data. Centera storage systems can hold 3.8 TB, 7.7 TB, 11.5 TB, and 15.4 TB in a single 19 inch rack cabinet, and scale to support 246 TB of total storage using multiple cabinets in a single cluster. A cluster consists of multiple Centera cabinets interconnected to create a single storage area.

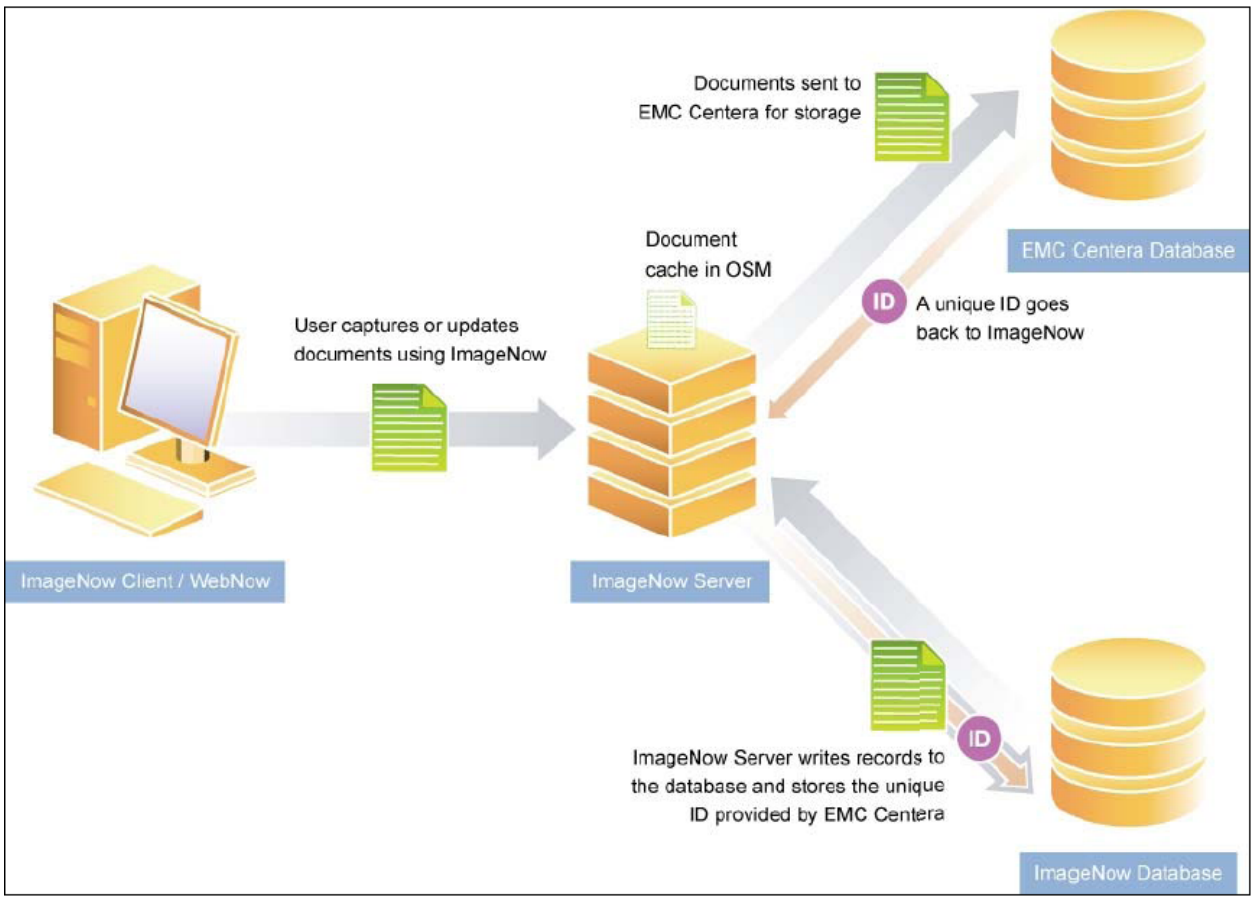
Discussion of the EMC Centera system is attributed in part to the EMC Centera website (www.emc.com), the EMC product guide, and the Centera API Reference Guide distributed by EMC Corporation. All other details are the intellectual property of Perceptive Software.

Perceptive Content communications with EMC Centera

Perceptive Content communicates directly with the EMC Centera server using the Centera API. While several gateway applications are available for use with EMC Centera, the Centera API provides the greatest performance and more options. Perceptive Content uses Centera API, version 3.1 SP1, which works with EMC Centera 3.0. Additionally, EMC Centera models that Perceptive Content can work with include EMC Centera Government Compliance Edition.

To integrate with EMC Centera, Perceptive Content takes all of the document objects that would otherwise be stored in the Perceptive Content OSM and writes them to the EMC Centera storage system. The EMC server is divided into nodes. When Perceptive Content sends a document to store in EMC Centera, that document is parsed into pieces and replicated. These document pieces are then stored in different nodes on the EMC Centera server. In the event of failure, a document can be stored in or retrieved from another set of nodes.

When the EMC System receives a packet containing a document, it returns a unique ID for that document. Thereafter, Perceptive Content uses that unique ID to request the target document as shown in the figure below. Using a unique ID to retrieve documents in storage is called Content Addressed Storage, where the unique ID or content address is used to fetch a document rather than a URL or filename. This content address is often referred to as a "claim check," discussed next in more detail.



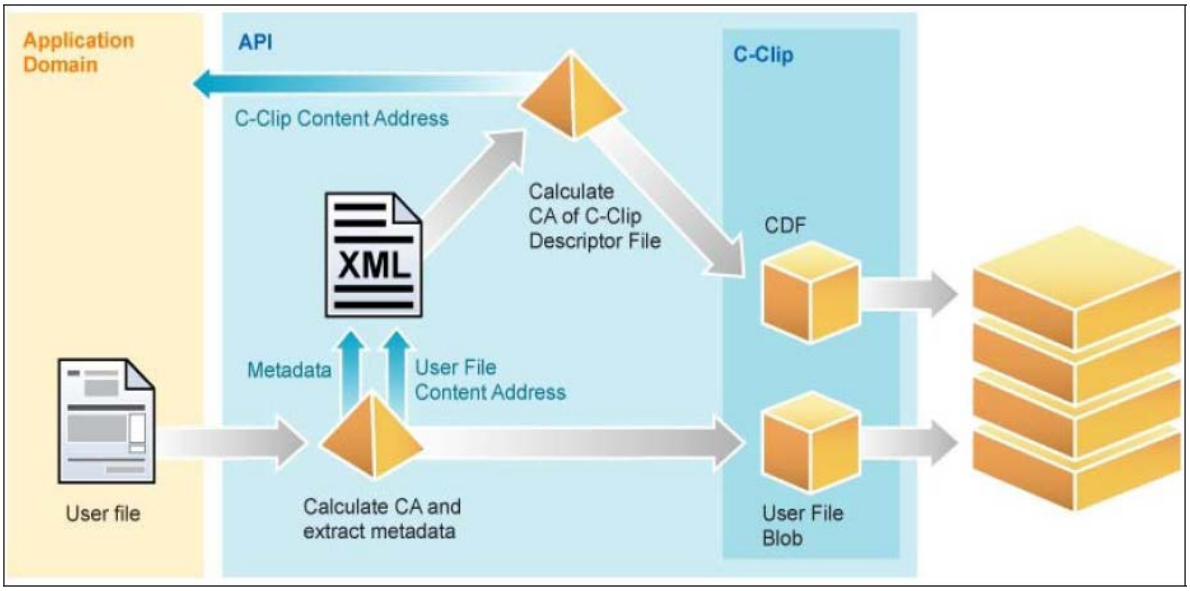
Storing and retrieving assets

Content Addressed Storage (CAS) is designed to offer authenticity, long life, scalability, and accessibility. It is built to manage fixed content. Unlike databases and file directories that can be updated and changed frequently, CAS systems are flexible but must also provide the highest security conditions available for content not intended to change. Perceptive Content always knows where an asset resides in EMC Centera: the content address (CA) is used to both store and retrieve assets from the Centera repository.

When Perceptive Content first stores (writes) a document object in EMC Centera, it is given a corresponding claim check in the form of a CA, as described earlier. Any subsequent requests for that document are made by providing Centera with that claim check. The following steps outline the way in which Perceptive Content stores and retrieves document objects in an EMC Centera system (and no installation kit is required to integrate Perceptive Content with EMC Centera):

1. Perceptive Content delivers a document object to the Centera API, which then calculates a corresponding claim check or CA and extracts the metadata such as the filename, creation date, and physical location.

2. Centera stores a BLOB (Binary Large Object) which contains the actual content that is accessible only by the CA. By attributing a CA to a BLOB, the content is verified as authentic and unique to the Centera system.
3. Both the BLOB and the metadata are contained in the C-Clip, an XML based description file called a CDF (Content Descriptor File) which has its own CA assigned to it. This C-Clip acts as the container for both the content and its metadata, and the CA assigned to the C-Clip as a handle for Centera to return the associated data.
4. A C-Clip CA is returned to Perceptive Content when Centera has made two copies of the CDF and two copies of the BLOB and has stored them as shown in the following figure.



A benefit of storing BLOBs of content is that it allows Centera to detect when duplicate content reaches the system. Only one instance is ever stored in addition to a mirror copy. Perceptive Content communicates with EMC Centera by referencing the C-Clip content address only. For customers running Perceptive Content prior to obtaining EMC Centera, Perceptive Content can copy pre-existing document objects.

To ensure maximum performance, Perceptive Content keeps a cache of each document in the OSM directory. When a user requests a document, Perceptive Content is able to return a document copy with minimal delay. Caching is relevant only for document retrieve sessions. See the “Centera API Calls used by Perceptive Content” section for more on caching as it relates to the Retrieve function.

Deleting documents

Even though EMC Centera is known for its ability to store documents for long retention periods a user can delete documents using Perceptive Content. There are two parts to deleting Perceptive Content data in EMC Centera. There is the deletion of clips and tags and the deletion of an actual BLOB, the latter of which must be deleted first.

Sample C-Clip structure

The following example shows the C-Clip structure that Perceptive Content writes to integrate with EMC Centera. The content from a document object is stored in a BLOB and each BLOB is assigned its own CA. The entire C-Clip as previously stated is also assigned a CA.

```
<c-clip>
  <top_tag>
    <Perceptive Content>
<!-Reference to the document file -->
<blob>JDFOWK983LD0934I</blob>
</Perceptive Content>
</top_tag>
</c-clip>
```

Configuration settings for EMC Centera

The following table outlines the Centera-specific setting available for use in the inow.ini file.

Group	Setting	Options	Description
OSM	centera.capacitycheck.enabled	TRUE FALSE	Specifies whether Perceptive Content Server checks the Centera device to determine if it has enough memory to store the OSM tree. TRUE = Perceptive Content Sever checks for memory. FALSE = Perceptive Content Server does not check for memory. The default is FALSE.

Centera API calls used by Perceptive Content

Perceptive Content uses the Centera API calls in this table to integrate with EMC Centera servers.

API Calls	Options	Description
	FPPool_Close	Closes the current C-Clip and frees memory on the Perceptive Content Server.
	FPPool_GetCapability	Returns the privileges, such as read, write, or delete, the Perceptive Content Server has on a Centera server.
	FPPool_GetLastError	Returns the last known error when one occurs.
	FPPool_GetLastErrorInfo	Provides additional information about the last error returned by the Centera server.
	FPPool_GetPoolInfo	Returns the capabilities of the pool, such as permissions, connection information, and other actions you can take on the Centera server.
	FPPool_Open	Opens a connection to the Centera server.
	FPPool_SetIntOption	Uses this function to set parameters to optimize the connection between the Perceptive Content Server and the Centera server.
Clip Functions	FPClip_Close	Closes the current C-Clip and frees up memory on Perceptive Content Server.
	FPClip_Create	Creates a new, empty C-Clip and stores it in memory. This function returns a reference to the new C-Clip.
	FPClip_Delete	Deletes the given CDF from the first writeable cluster of a given pool, but only when the retention period of the C-Clip is expired and the server capability “delete” is true.
	FPClip_GetName	Returns the name of the C-Clip.
	FPClip_GetTopTag	Returns the top tag in the .xml file (the “parent” tag).
	FPClip_Open	An accessible test server provided by Centera.
	FPClip_SetRetentionPeriod	Sets the C-Clip retention period. Note that Perceptive Content sets this period to zero.
	FPClip_Write	Writes the clip to the pool as a CDF and returns the C-Clip ID (Content Address).
Stream Functions	FPStream_Close	Closes the current stream.

API Calls	Options	Description
	FPStream_CreateFileForInput	Creates a stream to read from a file and returns a reference to the created stream.
	FPStream_CreateFileForOutput	Creates a stream to write to a file and returns a reference to the created stream.
	FPTag_BlobRead	Returns the BLOB data from the pool and writes it to the stream object.
	FPTag_BlobWrite	Writes BLOB data to the pool from a stream object that the application provides. This function opens a new BLOB, reads bytes from the stream object, writes the bytes to the pool, closes the BLOB, and associates the calculated CA with the given tag. By default, the CA is calculated by the client while data is being sent.
	FPTag_Close	Closes the given tag and frees all allocated resources.
	FPTag_Create	Creates a new xml tag within a C-Clip and returns a reference to it.
	FTTag_Delete	Deletes a tag (and all children of the tag) in the xml tag structure of a C-Clip. If the tag refers to data (a BLOB tag), it does not delete that data.
	FPTag_GetFirstChild	Returns the first child tag of the given tag.
	FPTag_GetSibling	Returns the sibling tag of the given tag.
	FPTag_GetTagName	Returns the name of the given tag in the open C-Clip.

Configuring a cache OSM

A cache OSM File System Storage (FSS) temporarily stores documents locally that you upload to your system. This allows the system to access the most recent documents that you have uploaded more quickly than retrieving them from the primary OSM (CAS).

Create a cache OSM set

Complete the steps to create your cache OSM set.

1. In a command-line interface, navigate to the **[*drive*:]\inserver\bin** directory.
2. Run the following intool command.`intool --cmd add-osm-set`.

3. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you create the cache OSM set.

Command window prompt	Example value
OSM Set ID (e.g., osm_01)	Enter a value, for example, **osm_99**.
OSM Set name	Enter a name, for example, **osm_99**.
OSM Set Location Type	To represent On_Line, enter **1**.
OSM Set Type	To represent mixed, enter **0**.
OSM Set Description	Enter a description, for example, **Cache for Primary OSM Set**.
Sub Object OSM Set ID	To leave the value blank, press ENTER.
Notes	Enter additional information, for example, **Cache for Primary OSM Set**.
OSM Integration Type	To represent FSS, enter **0**.
Cache OSM Set ID (leave blank to disable caching)	To leave the value blank, press ENTER.

The system displays the following message: OSM set was added successfully.

Create the cache OSM tree (FSS)

Complete the steps to create the OSM tree for the OSM set that you created.

1. In a command-line interface, navigate to the **[*drive*:]\inserver\bin** directory.
2. Run the following intool command: `intool --cmd add-osm-tree --type FSS`.

3. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you create the cache OSM tree.

Command window prompt	Example value
OSM Tree ID (e.g., osm_01.00001)	Enter a value, for example, **osm_99.0001**.
OSM Set ID (e.g., osm_01)	Enter a value that matches the OSM Set ID value for the cache OSM set that you created, for example, **osm_99**
OSM Tree Description	Enter a value, for example, **OSM Tree for Cache OSM Set**.
Enter Mirror On	To turn off mirroring, enter **0**.
OSM Tree Path (e.g., c:\inserver\osm_01.00001)	Enter a valid path for the tree you just created, for example, **c:\inserver\osm_99.00001**.
OSM Tree Next Slot (00000000/00000000/00000000)	Enter a value, for example, **00000000/00000000/00000000**.
Enter Media Type	To select magnetic, enter **0**.
Files Per Directory in OSM Tree	Enter a value, for example, **512**.
Retries	Enter a value, for example, **5**.
Delay	Enter a value, for example, **1**.

The system displays the following message: OSM tree was added successfully.

Connect the cache OSM set to the cache OSM tree (FSS)

After you create the cache OSM set and the OSM tree, you must complete the steps to point the set to the tree. To connect the previously created OSM set to the OSM tree, complete the following steps.

1. In a command-line interface, navigate to the **[*drive*:]\inserver\bin** directory.
2. un the following intool command: `intool --cmd update-osm-set`.

3. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you connect the cache OSM set to the cache OSM tree.

Command window prompt	Example value
OSM Set ID (e.g., osm_01),	Enter the value you used when creating the OSM set, for example, **osm_99**.
OSM Set Name	Enter the value you used when creating the OSM set, for example, **osm_99**.
OSM Set Location Type	Enter the value you used when creating the OSM set, to select On-Line, enter **1**.
OSM Set Type	To select mixed, enter **0**.
OSM Set Description	Enter the value you used when creating the OSM set, for example, **Cache for Primary OSM Set**.
Sub Object OSM Set ID	To leave the value blank, press ENTER.
Writable OSM Tree ID	Enter the tree ID you used when creating the tree, for example, **osm_99.00001**.
Notes	Enter additional information, for example**, Cache for Primary OSM Set**.
OSM Integration Type	To represent FSS, enter **0**.

The system displays the following message: OSM set was updated successfully.

Enable asynchronous write caching

You can optionally enable asynchronous write caching. To configure your cache OSM to be read and write accessible, complete the following steps.

1. In a command-line interface, navigate to the [*drive*:]\inserver\bin directory.
2. Run the following INTool command.

```
      intool --cmd update-osm-cache --permanent-osm-set osm_01 --cache-level read-
write
```

Step result The system delivers an `OSM cache updated successfully` message.

Configuring the Primary OSM (CAS)

The Centera OSM (CAS) provides a permanent storage space for your documents. Once documents leave your cache OSM that you created, they move to the permanent Centera storage.

Create the primary OSM set (CAS)

Before you can create the primary Centera CAS OSM set, you must first create the FSS cache OSM Set. Complete the following steps to create the CAS OSM set.

1. In a command-line interface, navigate to the **[*drive*:]\inserver\bin** directory.
2. Run the following intool command. `intool --cmd add-osm-set`.
3. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you create the Centera OSM set.

Command window prompt	Example value
OSM Set ID(e.g., osm_01)	Enter a value, for example, **osm_90**.
OSM Set Name	Enter a value, for example, **osm_90**.
OSM Set Location Type	To select Undefined, enter **0**.
OSM Set Type	To select Mixed, enter **0**.
OSM Set Description	Enter a value, for example, **Centera OSM Set**.
Sub Object OSM Set ID	To leave the value blank, press ENTER.
Notes	Enter a value, for example, **Centera OSM Set**.
OSM Integration Type	To select CAS, enter **1**.
Cache OSM Set ID	Enter the set ID that you entered for the FSS cache OSM set, for example, **osm_99**.

The system displays the following message: OSM set was added successfully.

Create Centera OSM tree (CAS)

Complete the steps to create the permanent Centera OSM tree.

1. In a command-line interface, navigate to the **[*drive*:]\inserver\bin** directory.
2. Run the following intool command: `[drive:]\inserver\bin>intool --cmd add-osm-tree --type CAS`.
3. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you create the Centera OSM tree.

Command window prompt	Example value
OSM Tree ID(e.g., osm_01.00001)	Enter a value, for example, **osm_90.00001**.
OSM Set ID(e.g., osm_01)	Enter the value that you used when creating the OSM set, for example, **osm_90**.
OSM Tree Description	Enter a value, for example**, Primary OSM Tree**.
Connection String	Enter a value, for example **128.221.200.184?c:\emea3\emea3profile1_rdqeDcw.pea**.

The system displays the following message: OSM tree was added successfully.

Connect the Centera OSM set to the OSM tree (CAS)

Complete the steps to point the CAS OSM set to the OSM tree.

1. Run the following intool command: `[drive:]\inserver\bin>intool --cmd update-osm-set`.
2. Follow the system prompts.

   Note The following table shows an example scenario of values you can enter when you connect the Centera OSM set to the OSM tree.

Command window prompt	Example value
OSM Set ID (e.g., osm_01)	Enter the value you used when creating the CAS OSM set, for example **osm_90**.
OSM Set Name	Enter the value you used when creating the CAS OSM set, for example, **Primary OSM Set**.
OSM Set Location Type	To select undefined, enter **0**.
OSM Set Type	To select mixed, enter **0**.
OSM Set Description	Enter the value you used when creating the CAS OSM set, for example**, Primary OSM Set**.
Sub Object OSM Set ID	To keep the value blank, press ENTER.
Writable OSM Tree ID	Enter the ID of the OSM tree you used when creating the OSM tree, for example, **osm_90.00001**.
Notes	Enter a value, for example, **Primary OSM Tree**.
OSM Integration Type	To select CAS, enter **1**.

The system displays the following message: OSM set was updated successfully.

(Optional) Enable asynchronous write caching

To enable read and write access to the CAS OSM, complete the steps.

- Run the following intool command: [drive:]\inserver\bin>intool --cmd update-osm-cache --permanent-osm-set osm_90 --cache-level read-write.

The system displays the following message: OSM cache updated successfully.

Point the primary storage set to the cache set

To point the Centera OSM set to the cached OSM set, complete the steps.

- Run the following intool command: `[drive:]\inserver\bin>intool --cmd add-osm-cache --permanent-osm-set osm_01 --cache-osm-set osm_99`.

The system displays the following message: OSM cache successfully added.