IBM Content 2017

New Tools for Digital Business

Sweep Framework

#IBMContent2017

Agenda

- The Sweep Framework
  - What is a sweep
  - Framework features
  - Types of sweep
- Building a Sweep
- Gotchas
- Customer use case
- Performance Tuning
- Questions

What problem needs this solution

- Sometimes, you need to look at or act on a lot of objects. Maybe all of them
- Doing that from a client
  - Is inefficient
  - Has a few logic traps
  - Requires a high-privilege userid in a script or batch job
- Means creating a framework for running the scripts/batches, recovering from problems, keeping track of results
- CPE server itself has the need to look at many objects for specialized activities
- CPE server internal solution was generalized and made available

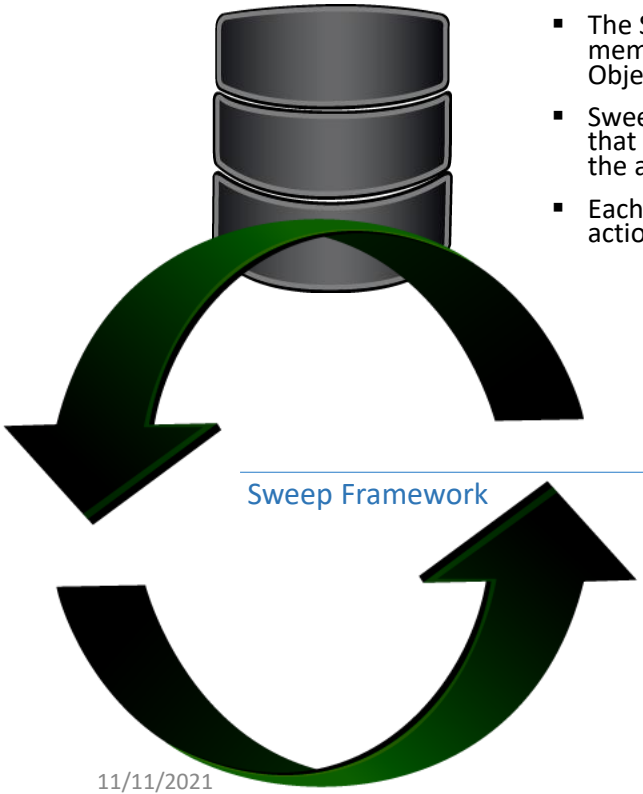
What is a sweep?

- A process which scans (sweeps) through a set of candidate objects, selecting those that satisfy a rule and applying an action to each matching object
- Much like issuing a query, iterating through the results and acting on each
- The action applied may be built-in or custom

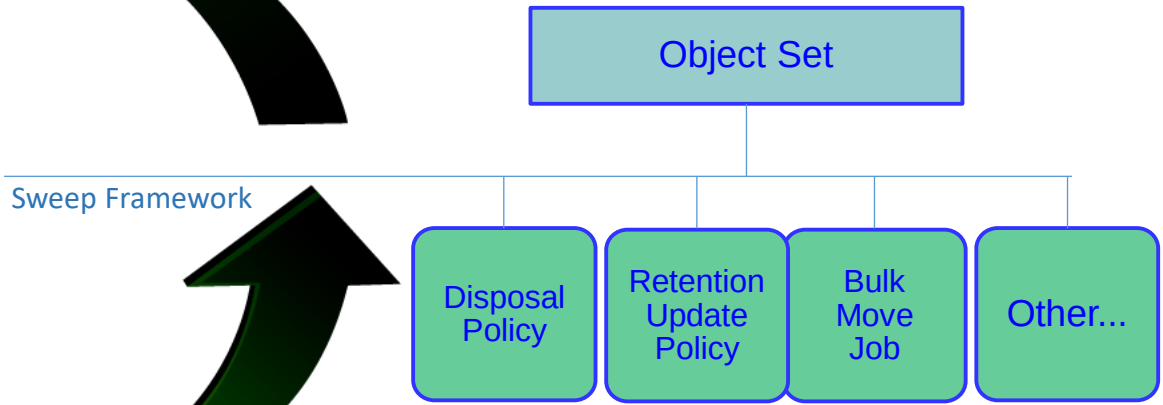
The sweep framwork

- The Sweep Framework was initially introduced in CPE 5.2 and has been expanded since then
- It is a Content Platform Engine subsystem that provides a generalized mechanism for efficiently examining large sets of objects and applying some operation to a user defined subset
- We refer to this process as 'Sweeping'

The sweep framework



- The Sweep Framework delivers batches of objects that are members of some target class to subscribing Sweep Objects
- Sweep Objects have an associated action and filter criteria that are applied to each object in the batch to determine if the action should be applied
- Each Sweep Object type implements a specific sweep action



Services offered by the framework

- Scalability
  - Horizontal scaling is achieved by increasing the number of server instances
  - Vertical scaling is achieved by increasing the number of worker threads dedicated to sweep processing
- Load management
  - Vary the allocation of server resources to sweeping
- Scheduling
  - Sweeps can be scheduled to run on specific days of the week and at specific times of the day
  - Allows resource intensive sweep operations to occur during off-peak hours
  - Provides the ability to halt and resume execution without losing context
- Failure recovery
- Error logging and auditing
- Performance monitoring

Sweep terminology

- Sweep Iteration
  - A single pass in which each object in the set of candidate objects to which a specific sweep instance has expressed an interest is visited exactly once
- Background search, job sweeps, queue sweeps, sweep policies
  - Specific types of sweep
- Sweep policies
  - For sweeps that need to be run on an on-going basis, provide the schedule and tuning settings
- Sweep actions
  - An action handler (sort of like an event handler) that allows you to create custom sweeps

Types of sweep

- Three basic types
  - Job: Runs a single scan
    - Example: bulk move
  - Policy-controlled: Scans repeatedly under the control of one or more policies
    - Example: archival process
  - Queue: Specialized repeating scan
    - Used internally for tasks such as thumbnail generation

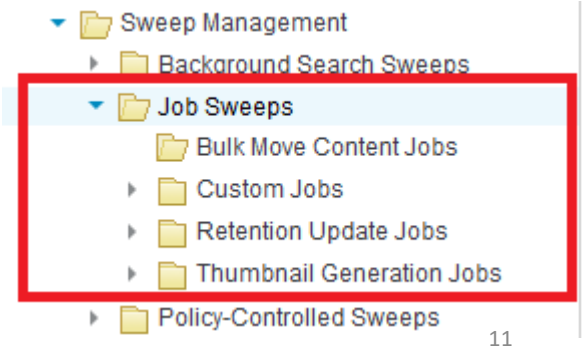
Common characteristics

- Target class – with include subclass option
- Filter expression
- Timeslots for scheduling
- Statistics recording – number of objects examined, processed, and failed
- Action handler

Sweep jobs

- Perform single iteration sweeps
- Have a definite start and completion
  - Starts when the first candidate object is visited
  - Ends when each candidate object has been visited exactly one time or when the end time assigned to the sweep is reached
- Once run, a Sweep Job cannot be restarted, instead the job can be cloned, modified as needed, and then re-run
- Sweep Jobs are useful for performing one-time-only bulk updates of many objects

  Examples:
  - Bulk Move Content Job
  - Retention Update Job
  - Thumbnail Generation Job



Policy-controlled sweep

- Scans repeatedly (iterations)
- Guarantees to process all matching objects eventually
- Looser guarantee (“eventually”) allows for greater concurrency

Bulk move versus archiving

- Both bulk move and archiving involve moving content from location A to location B
- The difference between the two is that bulk move is usually a process you want to complete just once, while archiving is a continuous process
- A bulk move is used to
  - Move federated content from storage external to the object store to a storage area that is part of the P8 environment (can be used with CFS-IS, CFS-CMOD, and CFS-ICI)
  - Move content from a storage facility that is being retired
- With a bulk move, you identify the complete set of documents that need to be moved at the start of the process and once that set has been moved, the bulk move job ends
- An archive move is used to move content that is no longer actively being used to lower cost storage
- An archive move is a process that has no defined end. As long as documents keep meeting the requirements for the archive, the archive process will handle them
- Bulk move is considered a “job sweep”
- Archiving is considered a “policy sweep”

Agenda

- The Sweep Framework
  - What is a sweep
  - Framework features
  - Types of sweep
- Building a Sweep
- Gotchas
- Customer use case
- Performance Tuning
- Questions

Building a sweep: Sweep Properties

- Sweep Target
  - OVP that refers to a Class definition that is used to specify the set of candidate objects for which the sweep job is declaring an interest
    - In some sweep definitions, you get to choose the class from a drop down list, for others you need to copy the object reference to the clipboard and then paste the value into the sweep definition
  - Scope can include sub-classes
  - Must be a searchable class
    - Searchable means it can be used in a 'FROM' clause of a query
    - Implication is that all instances must be stored in a single table

      This is important because it effects the way sweeps can be scheduled

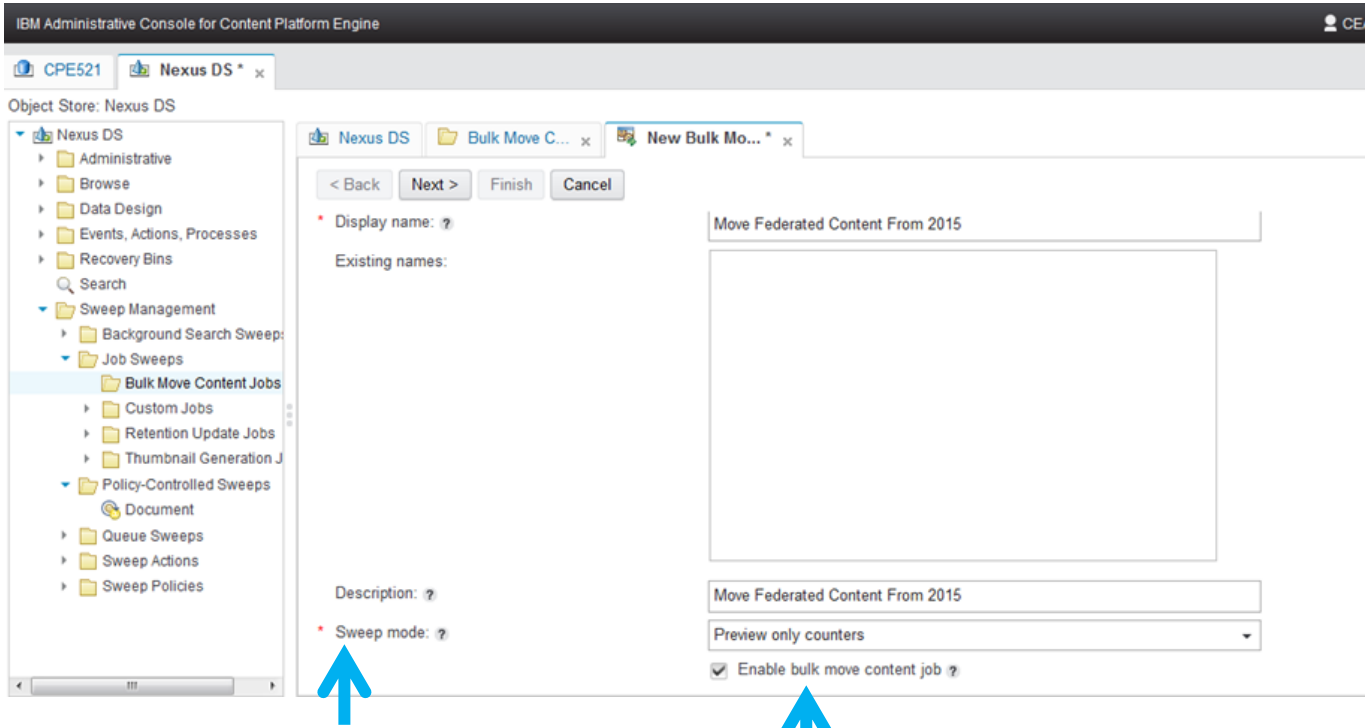
Building a sweep: Sweep Properties

- Filter Expression
  - Determines the subset of objects on which the action associated with the sweep will be performed
  - Logical expression consisting of property comparisons
  - Syntax is very similar to the WHERE clause of a query expression
  - Example
    - Color = “Blue”
  - See the following tech note for more examples: http://www.ibm.com/support/docview.wss?uid=swg22004491
- Timeslots
  - List of TimeSlot objects
  - Determines execution schedule
- Sweep Result - Enumeration of Sweep Result objects
  - Preview Results
  - Failure Results

Building a sweep: Sweep Properties

- Sweep Start Date
  - System date/time stamp that indicates when the sweep job started
- Sweep End Date
  - System date/time stamp that indicates when the sweep job completed
- Counters:
  - Examined Object Count
  - Processed Object Count
  - Failed Object Count
- IsEnabled
- SweepMode
  - Normal
  - Preview
  - Preview counters only

Building a sweep: Move content example



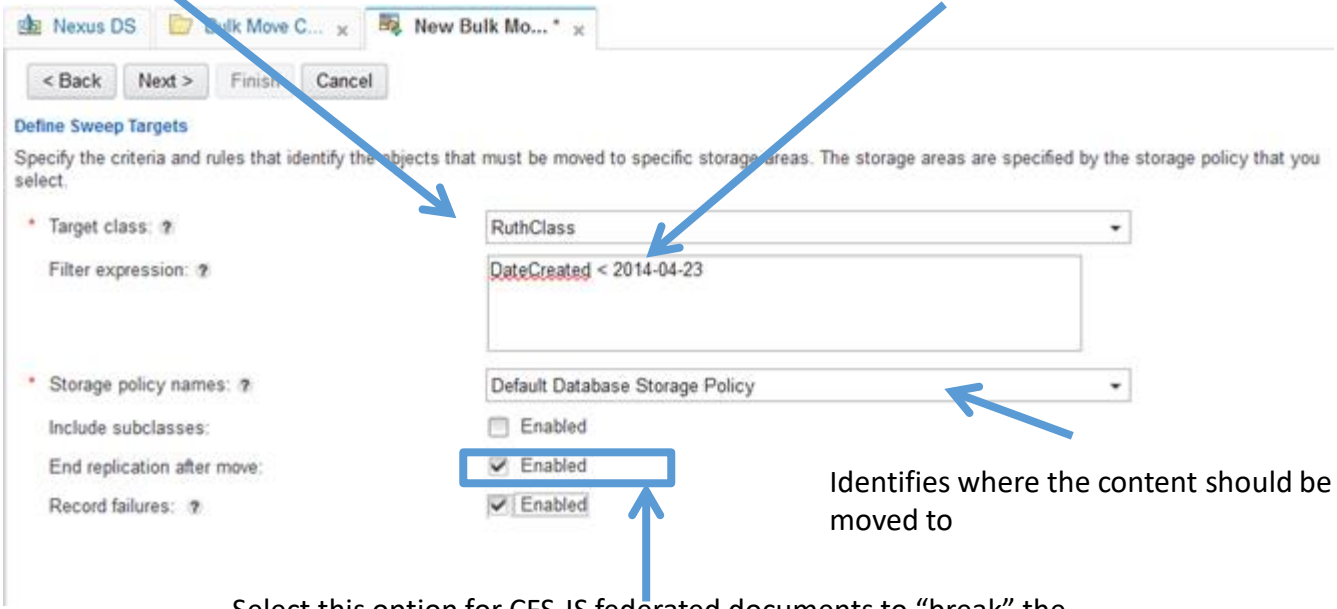
Default setting is preview only counters – will tell you how many objects will be affected

By default, the job is not enabled and therefore wont run

Building a sweep: Move content example

Select class from drop down list

Identify which documents to move from the specific class. Looks like the “Where” clause from a search.



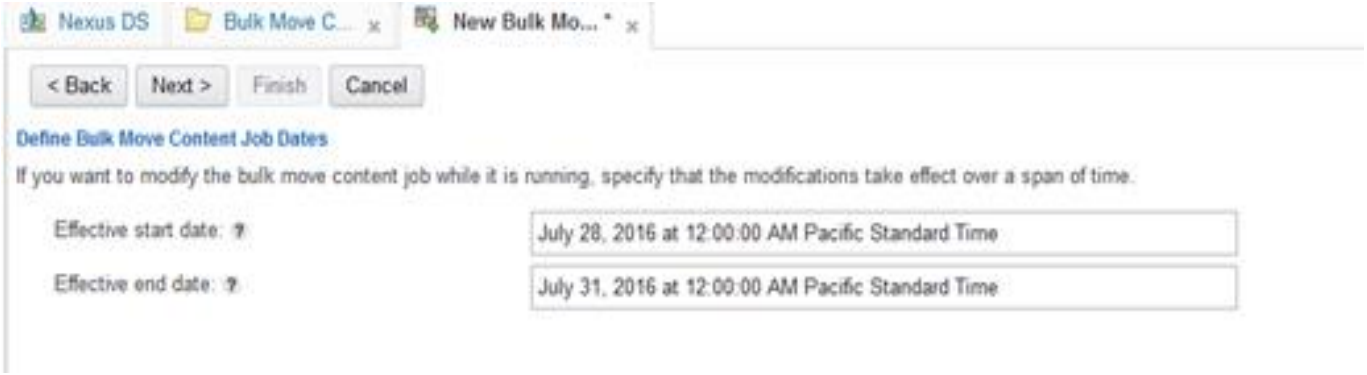
Identifies where the content should be moved to

Select this option for CFS-IS federated documents to “break” the federation. Not relevant for CFS-ICI federation or for non-federated documents.

Building a sweep: Move content example

Setting the schedule:

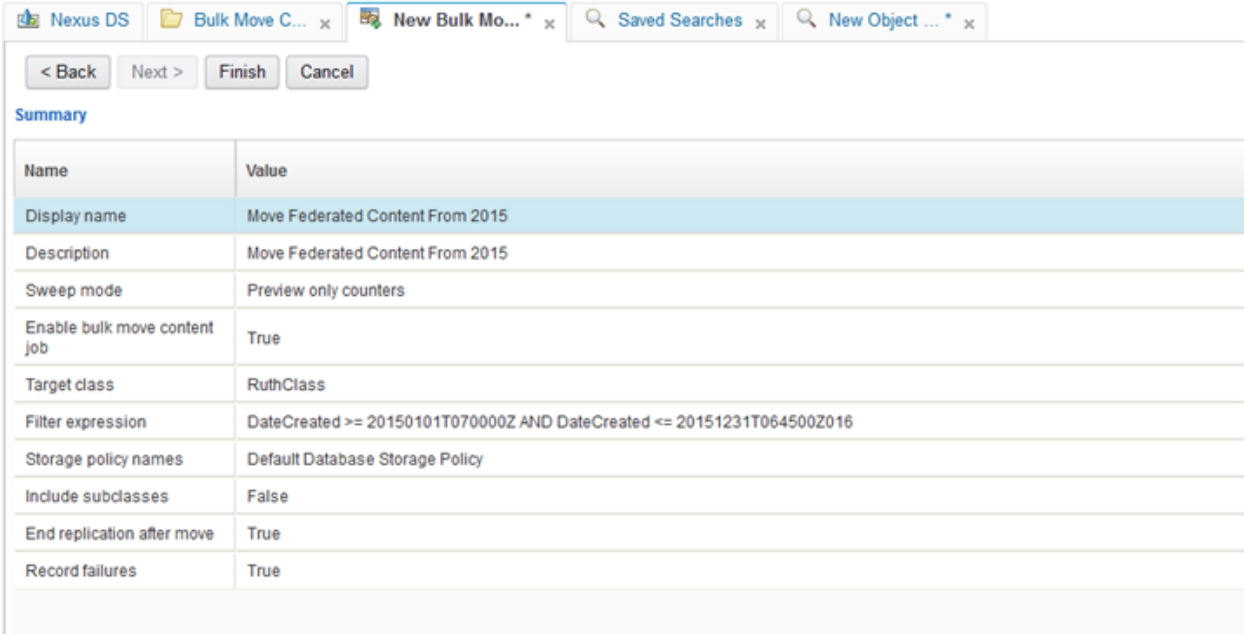
- By default jobs start as soon as the wizard is complete and run until all the selected objects have been processed
- Use the Effective start and end dates to limit when the job will run
- When the end date is reached, the job will stop even if there are candidate objects that have not been processed
- The date must be entered in the following format: mm/dd/yyyy. For example, enter 04/01/2011 for 1 April 2011
  - Caution: This is not the same date format that is used in the Target Expression!
- If you have entered valid date information….it will get translated into a long date format



Building a sweep: Move content example

Final screen of the wizard

- Review the information and make sure the who, what, when is correct. Once you click Finish, the job will start if it is “enabled”.

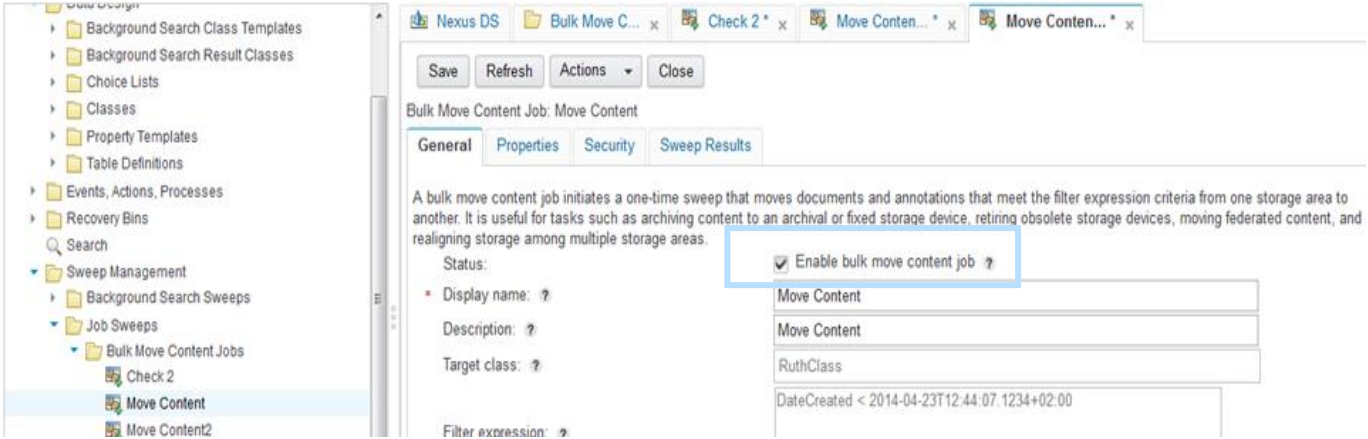


Name	Value
Display name	Move Federated Content From 2015
Description	Move Federated Content From 2015
Sweep mode	Preview only counters
Enable bulk move content job	True
Target class	RuthClass
Filter expression	DateCreated >= 20150101T070000Z AND DateCreated <= 20151231T064500Z016
Storage policy names	Default Database Storage Policy
Include subclasses	False
End replication after move	True
Record failures	True

Building a sweep: Move content example

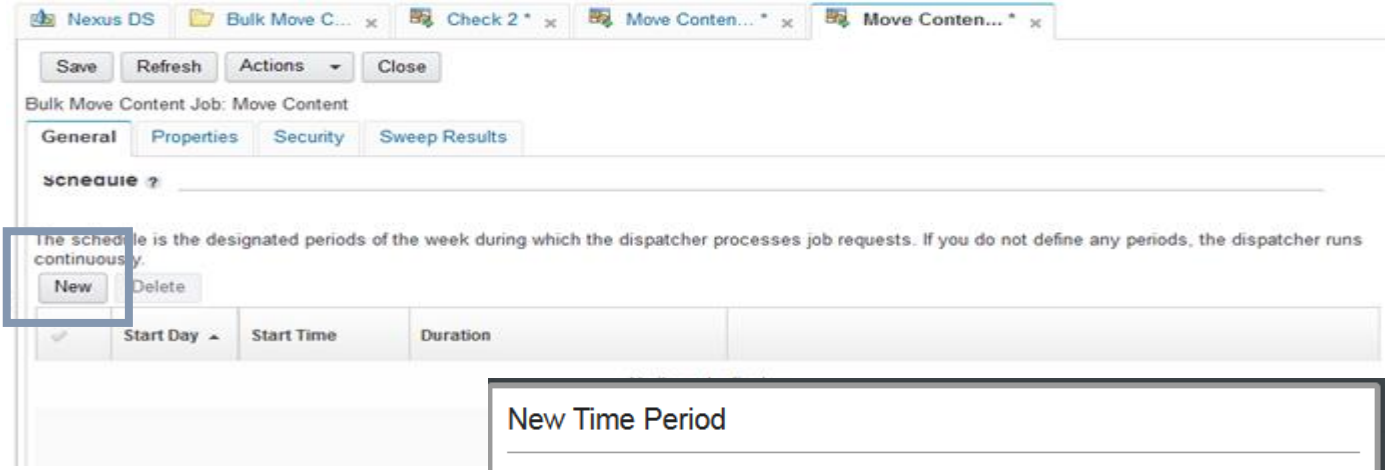
Adding a schedule

- Once the job has been saved, you can define a schedule for the job
- Open the job and on the General tab, check the “enable bulk move content job” option

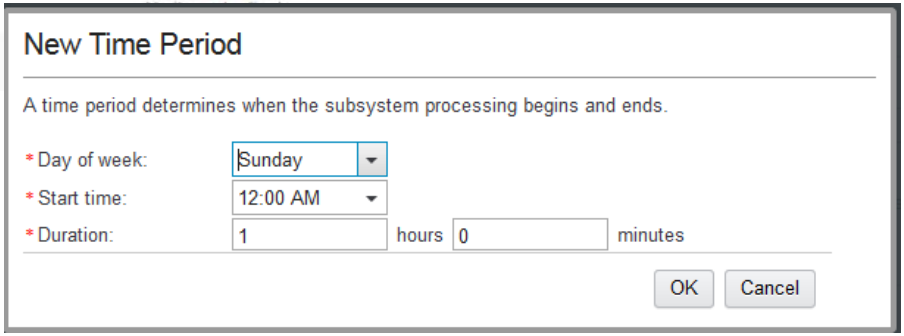


Building a sweep: Move content example

- Then scroll to the bottom of the screen to the scheduling section and click New

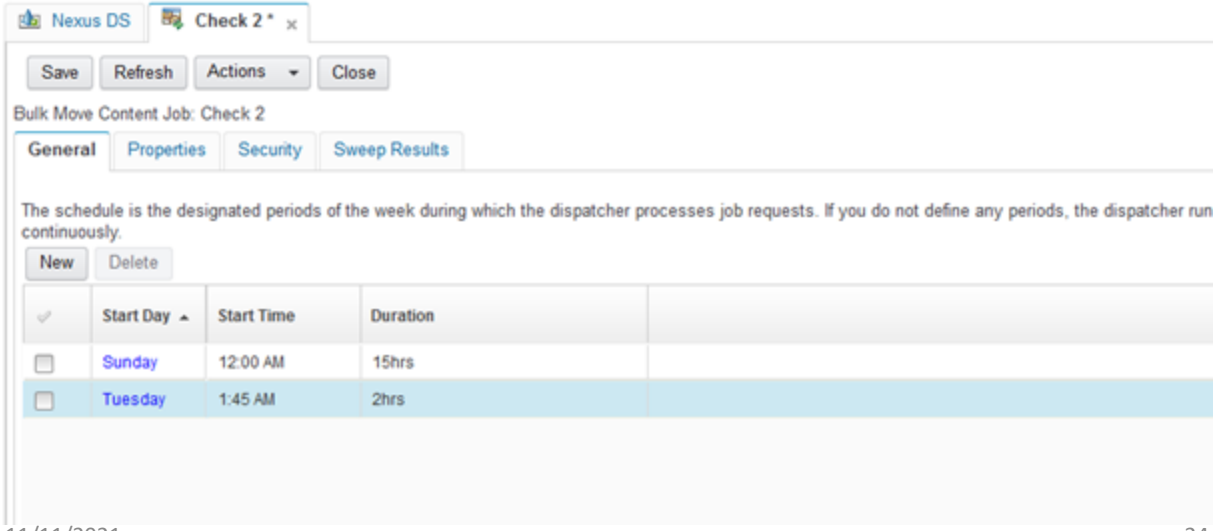


- This popup displays



Building a sweep: Move content example

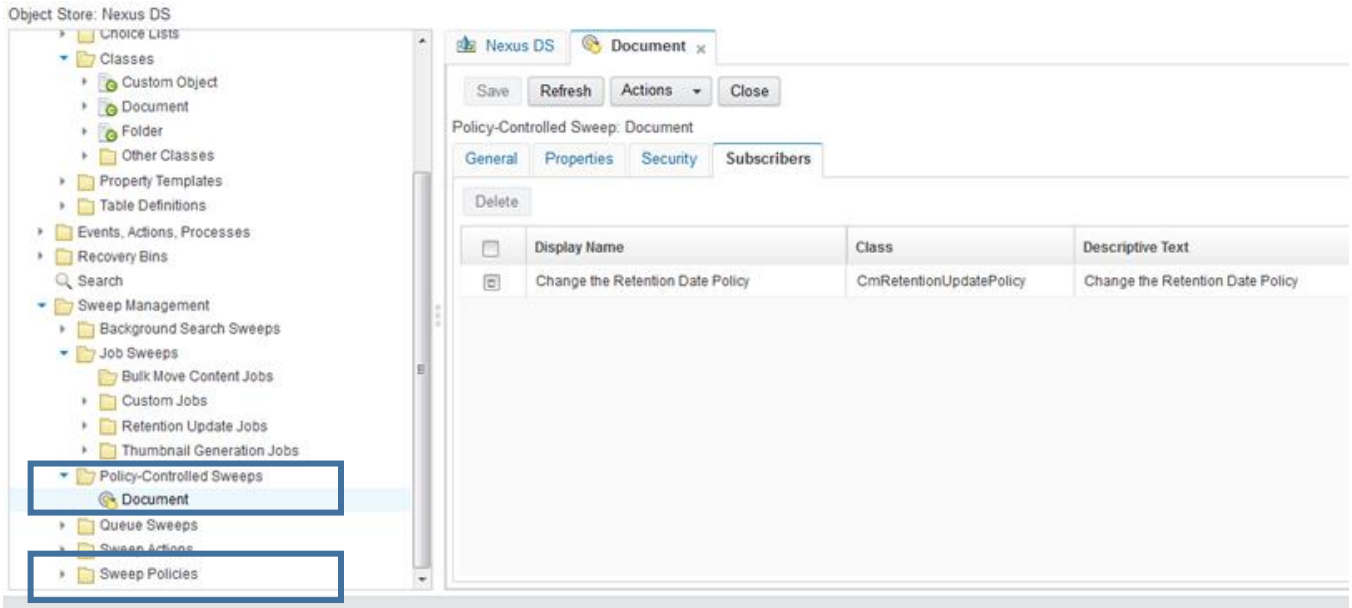
- Keep adding new schedule times as needed
- The job will continue to run in the selected times until all items are processed or the effective end date is reached



Policy Controlled Sweeps and Sweep Policies

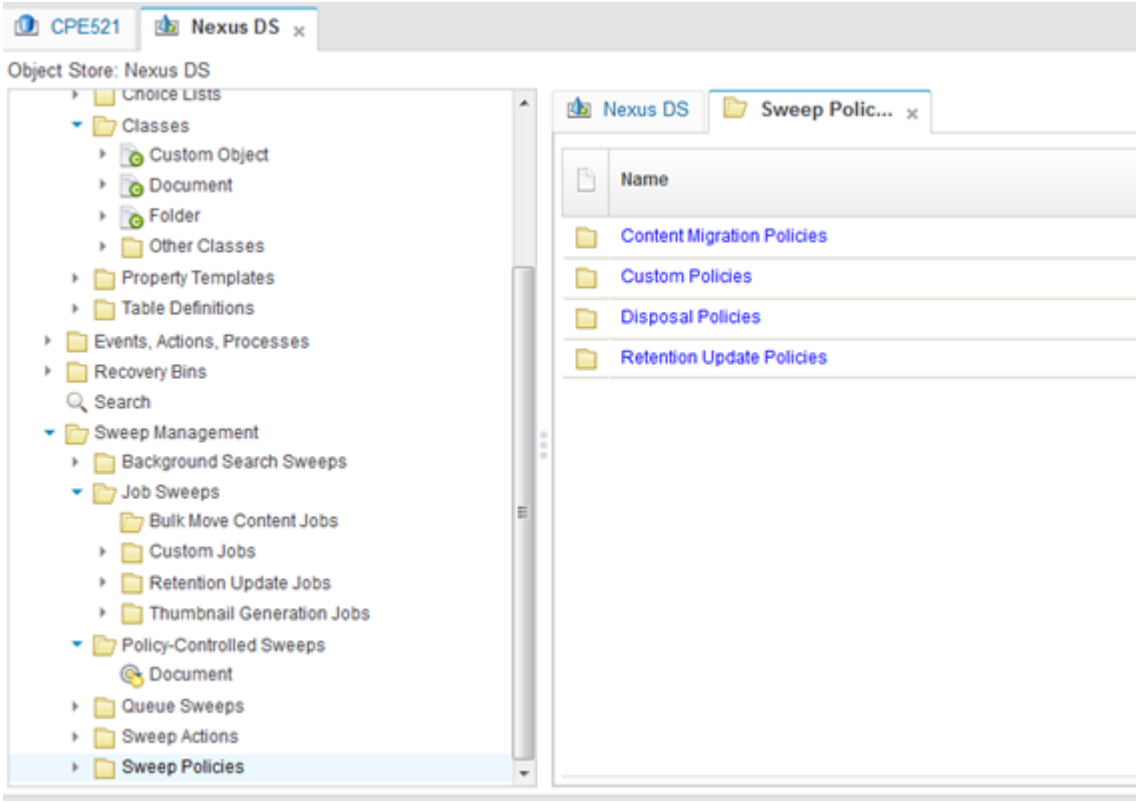
- The names are a little confusing!
- At a high level
  - Sweep policy says “do this”
  - Policy controlled says “when….and what have I done in the past”
- There can be lots of sweep actions and sweep policies, but only one policy controlled sweep per object type
- An object type is defined by a root class – that is, documents, folders, custom objects, or objects defined by a custom root class
- There can be at most one policy controlled sweep per root class type, that is one for documents, one for folders, one for custom objects and one for each custom root class
- You cannot create a policy controlled sweep directly, instead the policy controlled sweep for a specific root class gets created automatically the first time a sweep policy is defined that uses that root class (or subclass)

Policy Controlled Sweeps, Sweep Policies, and Sweep Actions

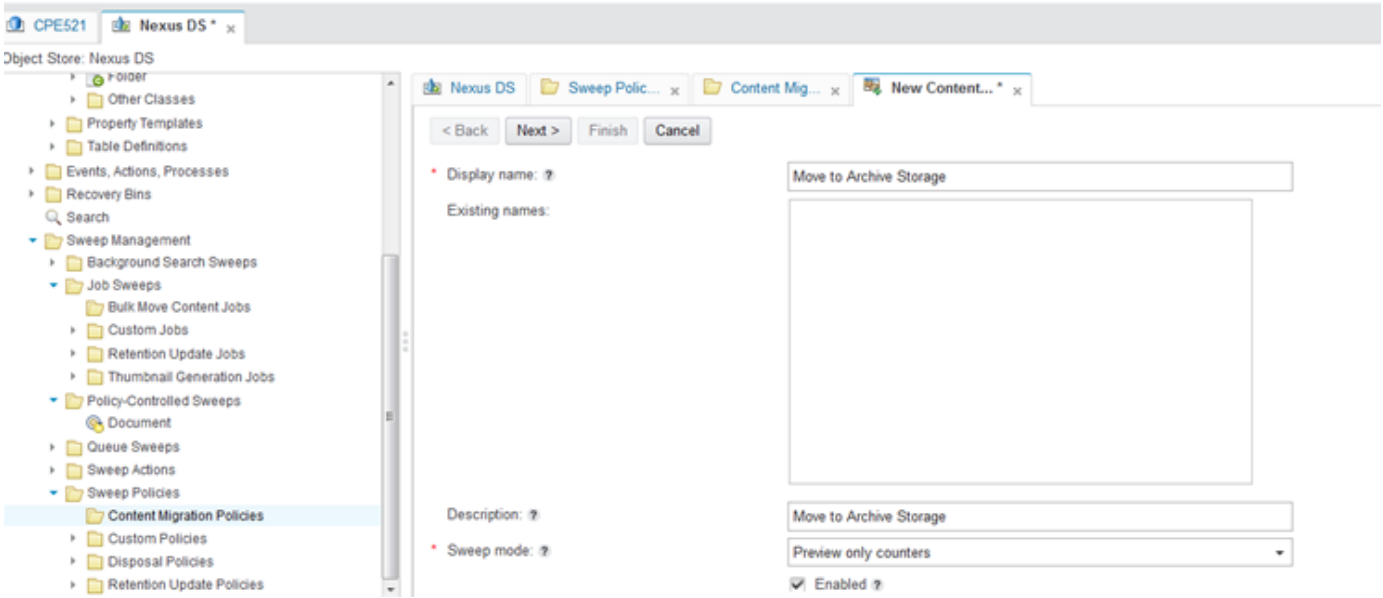


These items are related: policy controlled sweeps and sweep policies

Sweep policies

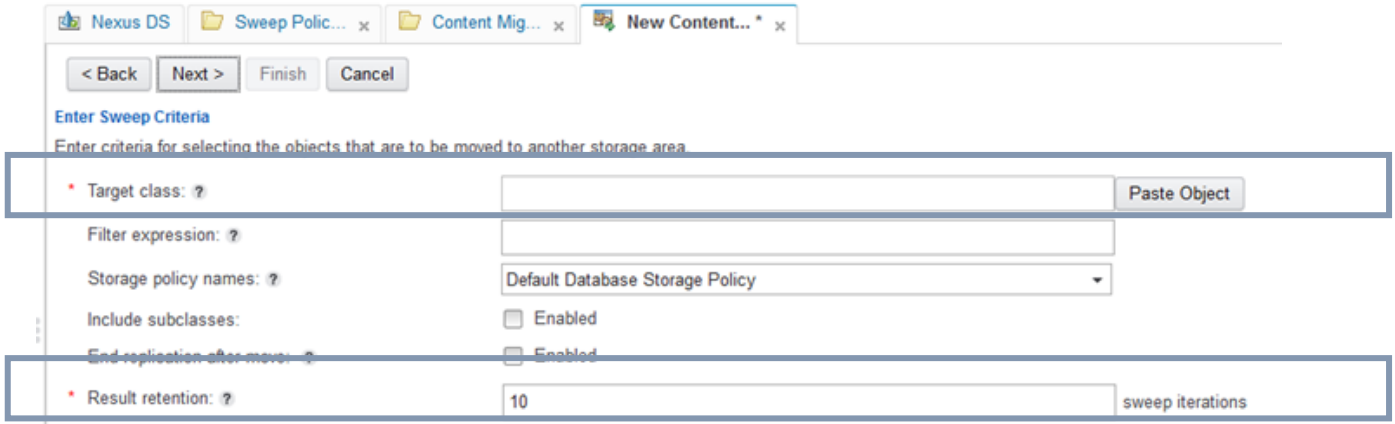


Content Migration Policy Wizard



First step looks just like the Job Sweep Wizard – provide a name, description, select a mode, and choose to enable the sweep

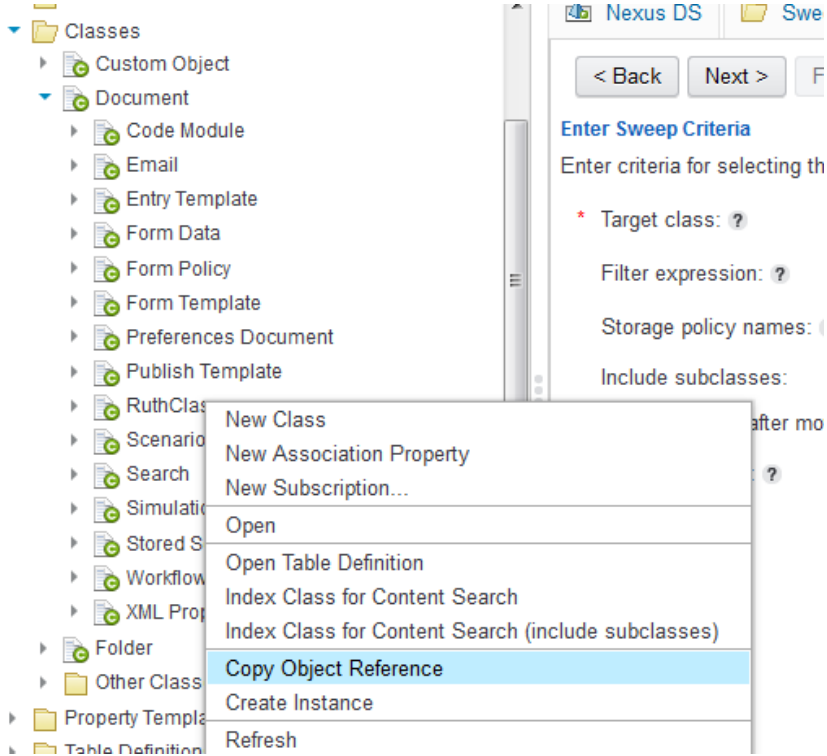
Content Migration Policy Wizard – Define the what



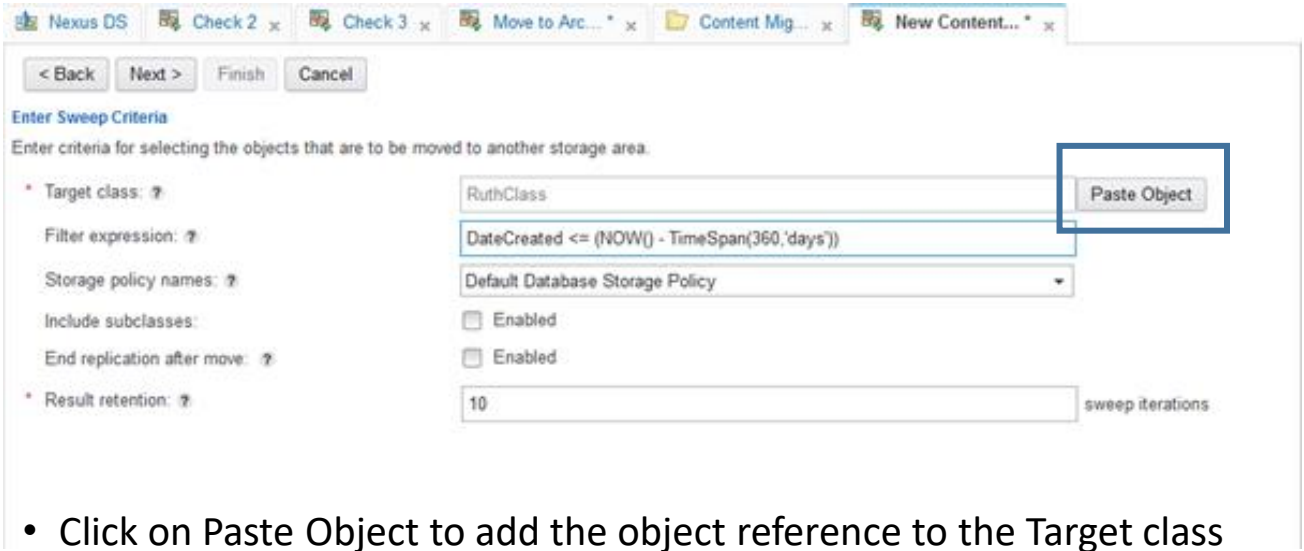
- Looks similar to job sweep wizard, except for
  - The Target class
  - Instead of a drop down list, you have to paste a reference to the class object
- The last field.
  - Instead of “record failures” there is the option “result retention”
  - Since the sweep is going to run multiple times, the result retention value indicates how many sets of results information should be maintained
  - Default value is 10

Identifying the target class

- In ACCE navigate to the target class definition
- Right-click and select object reference
- That places the object reference id in the clipboard
- Go back to the Sweep wizard and paste the reference

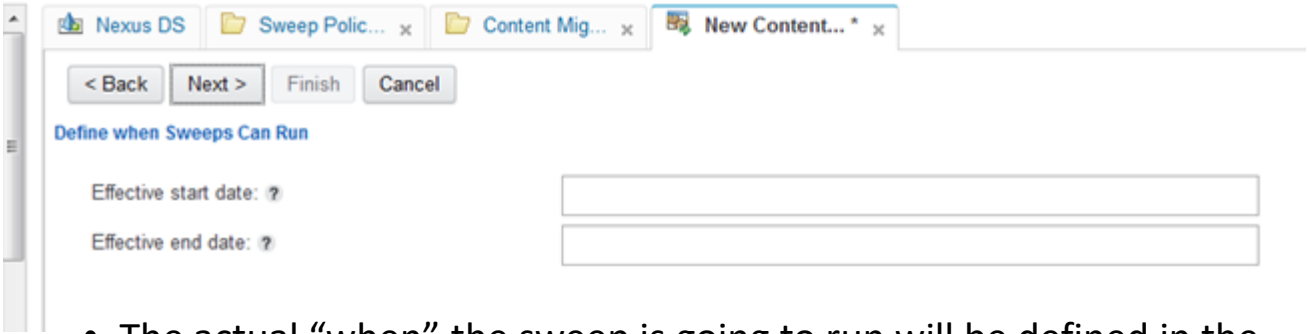


Identifying the candidate objects



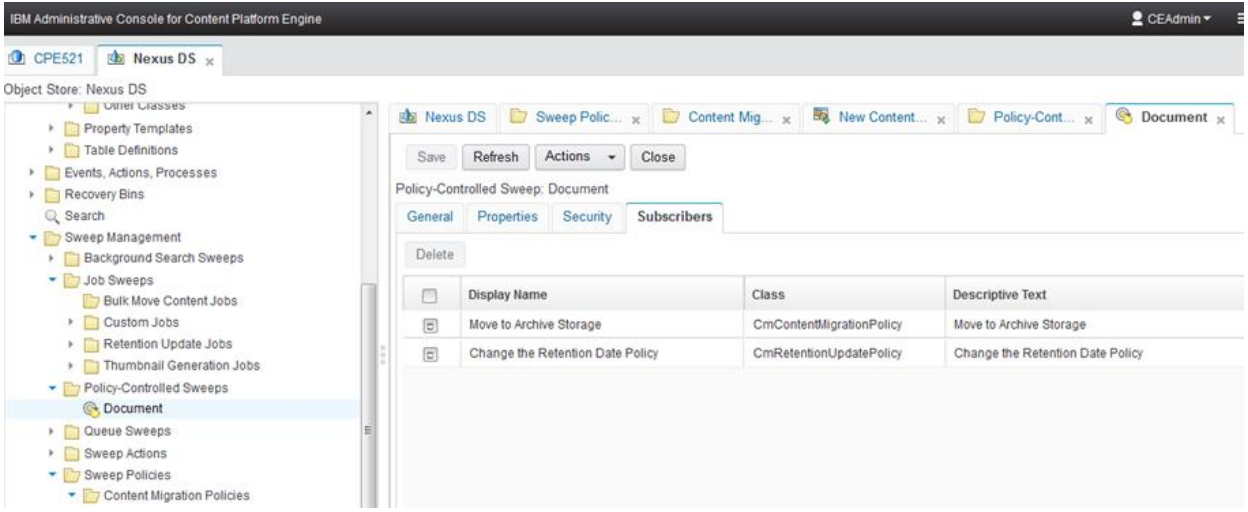
- Click on Paste Object to add the object reference to the Target class field – note the class name displays but slightly greyed out
- Enter a filter expression to select a subset of the members of the target class. The expression shown looks for all documents that are at least one year old
- The storage policy identifies where the objects should be moved to

Define how long the sweep is valid for



- The actual “when” the sweep is going to run will be defined in the sweep policy, use these fields to define the period of time that the sweep is valid for
- For example, you might want the sweep to be valid for the next two years only
- As with sweep jobs, enter the date in the format MM/DD/YYYY
- The next screen gives a summary of the choices made, and then when you click finish the sweep policy is created and if it didn’t already exist, a policy controlled sweep object is also created

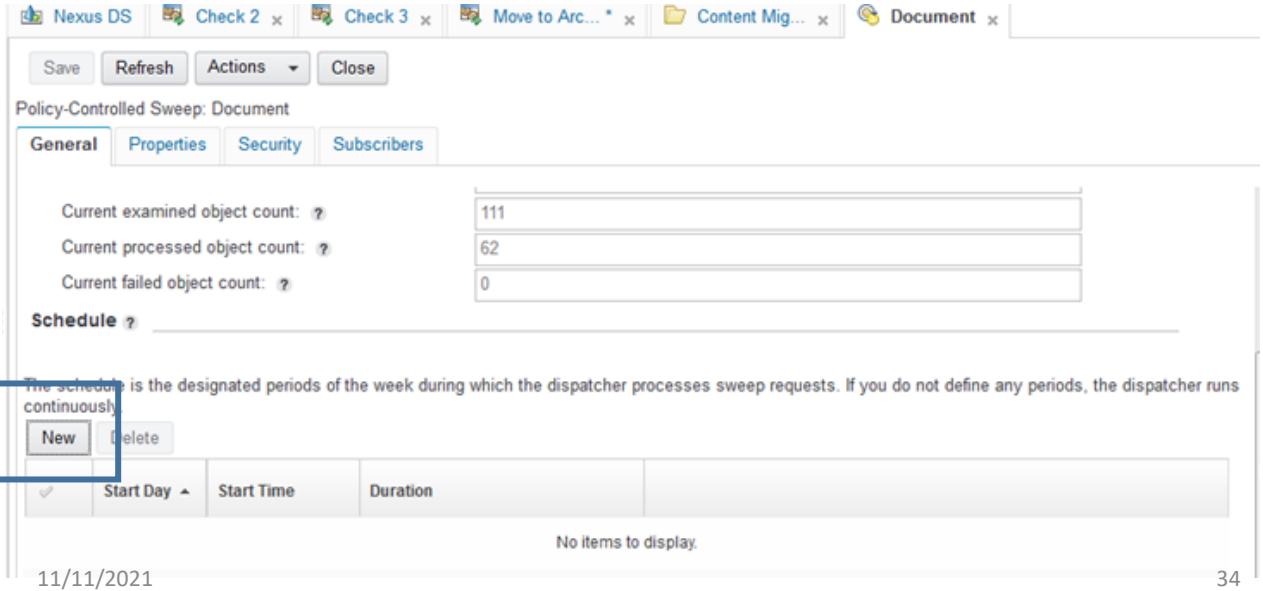
Sweep policy



- The Document policy controlled sweep is automatically created (if it doesn't already exist) when a content migration sweep policy is created
- The new sweep policy appears on the list of Subscribers in the Document Policy Controlled Sweep definition

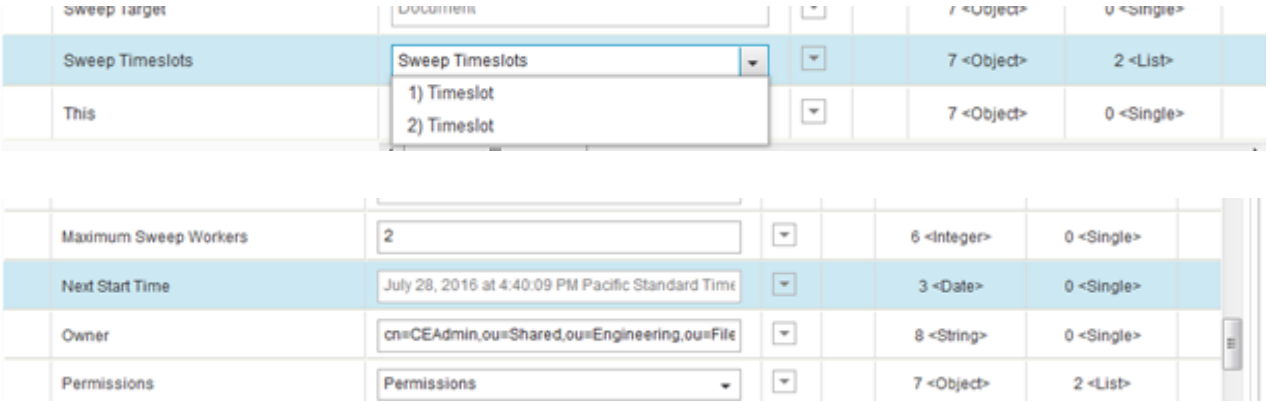
Sweep policy schedule

- On the General tab define the periods when the sweeps will run. If you don't add any time periods, all the subscribing sweeps will run continuously…probably not what you want!
- The number of time slots you define will be reflected on the Properties tab as Sweep Timeslots



Sweep schedules

- On the properties tab, in addition to seeing the number of timeslots, you will also find the next time the sweeps are scheduled to start
- Hint: To alphabetize the list of properties, click the Property Name column header



Sweep actions and other sweep policies

- Sweep actions are the equivalent of event handlers but for sweeps
- Use a sweep action with a custom sweep
- Out of the box, CPE also provides the framework for
  - Disposal policies – i.e. deleting documents that meet the designated criteria
  - Retention updates – changing the retention settings on designated documents. This sweep is the only way to reduce retention periods on documents that have already got a specific retention date set

Policy controlled sweeps: setting the schedule

- Format for date values 2014-04-23T12:44:07.1234+02:00
- YYYY-MM-DDTHH:MM:SS
- Don’t have to include time

Where clause examples

Take a look at this topic for example WHERE clause syntax

http://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.dev.ce.doc/query_sql_syntax_rel_queries.htm

- Based on title: DocumentTitle LIKE '%Acct%‘
- Based on creation date:
  - DateCreated < 2014-04-23T12:44:07.1234+02:00
  - DateCreated > 2014-04-23T12:44:07.1234+02:00 AND DateCreated < 2016-04-23T12:44:07
  - DateCreated <= (NOW() – TimeSpan(365,’days’))
- To delete superseded minor versions
  - Target class: Document (or Document subclass)
  - Filter expression: MinorVersionNumber > 0 AND IsCurrentVersion=False AND DateLastModified + TimeSpan(30,’days’) < Now()
- To delete temporary folders that no longer have any contents
  - Target class: TemporaryFolder
  - Filter expression: Containees IS NULL AND SubFolders IS NULL AND DateCreated + TimeSpan(24,’hours’) < Now()

Agenda

- The Sweep Framework
  - What is a sweep
  - Framework features
  - Types of sweep
- Building a Sweep
- Gotchas
- Customer use case
- Performance Tuning
- Questions

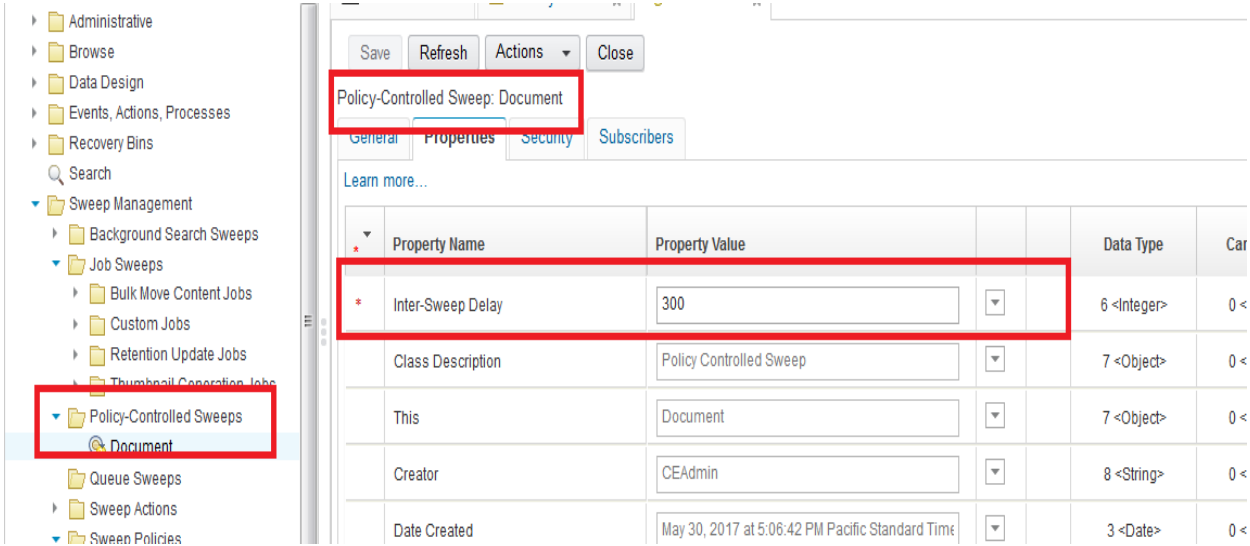
Gotchas

- Once you've created a sweep job (for example, a bulk move job), you can only
  - Enable or disable the job
  - Add schedule slots for when the job should run

  To make other changes, such as change the mode from Preview Counter to Normal or modify the filter expression, clone the job
- You cannot set separate schedules for sweep policies, the schedules are set using the Policy-controlled sweep for that object type (document, custom object, folder etc.)

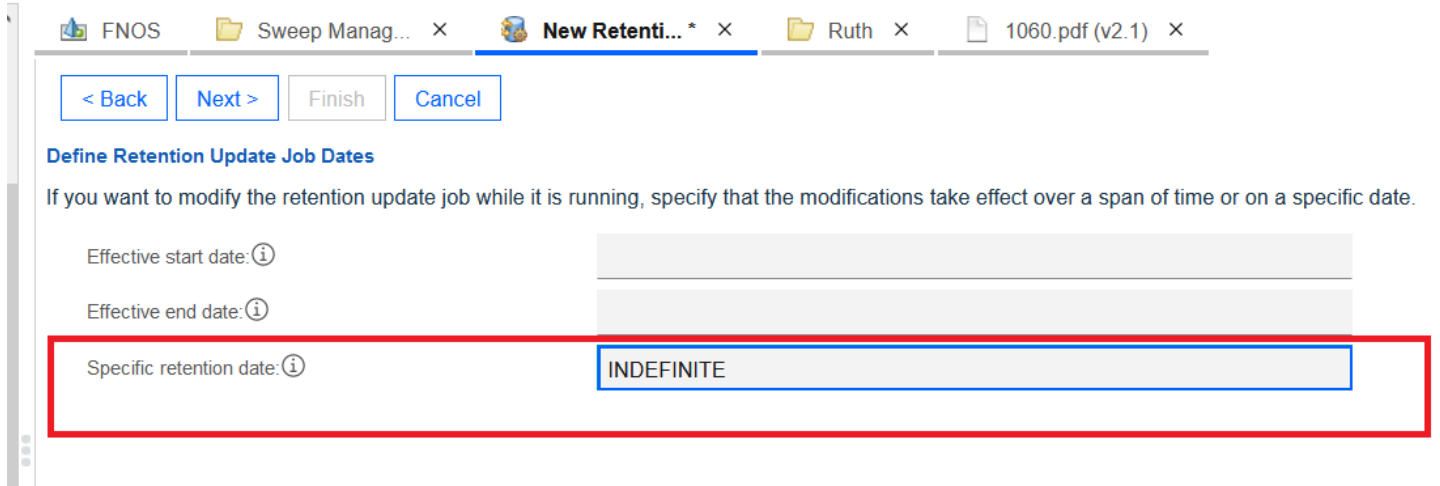
Gotchas

- Not really a “gotcha” – but if you want to limit the number of times the sweeps run in the allotted time slot, change the inter-sweep delay setting on the Policy Controlled object.
- Default is 300 seconds



Gotchas

- How do you set indefinite or permanent reduction via sweep
  - In the *Specific Retention Date* box of the retention sweep you can type in the literal PERMANENT or INDEFINITE (not intuitive, but it works).



Agenda

- The Sweep Framework
  - What is a sweep
  - Framework features
  - Types of sweep
- Building a Sweep
- Gotchas
- Customer use case
- Performance Tuning
- Questions

Actual customer use case

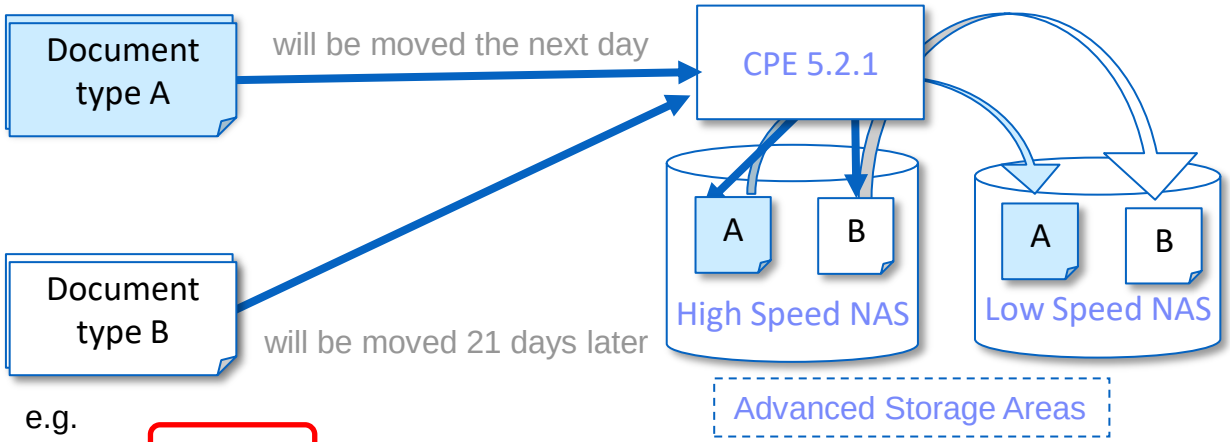
- Use Case
  - Documents with the same Document class have to be moved from High Speed NAS to Low Speed NAS
  - 5 and half hours (0:00 – 5:30) can be used for the Policy Controlled Sweep per day

Expected customer volumes

Document type	ratio	doc / peak day				
		2017	2018	2019	2020	2021
type A	80%	230,557	705,935	741,232	778,293	817,208
type B	20%	57,639	176,484	185,308	194,573	204,302
total	100 %	288,196	882,419	926,540	972,867	1,021,510

Document type	ratio	doc / year					
		2017	2018	2019	2020	2021	total
type A	80%	35,902,682	75,523,586	79,299,765	83,264,753	87,427,991	361,418,777
type B	20%	8,975,671	18,880,897	19,824,941	20,816,189	21,856,998	90,354,696
total	100%	44,878,353	94,404,482	99,124,707	104,080,942	109,284,989	451,773,473

Use case requirements



e.g.
In 2018, 882,419 documents should be moved per day.

Document type	ratio	doc / peak day				
		2017	2018	2019	2020	2021
type A	80%	230,557	705,935	741,232	778,293	817,208
type B	20%	57,639	176,484	185,308	194,573	204,302
total	100 %	288,196	882,419	926,540	972,867	1,021,510

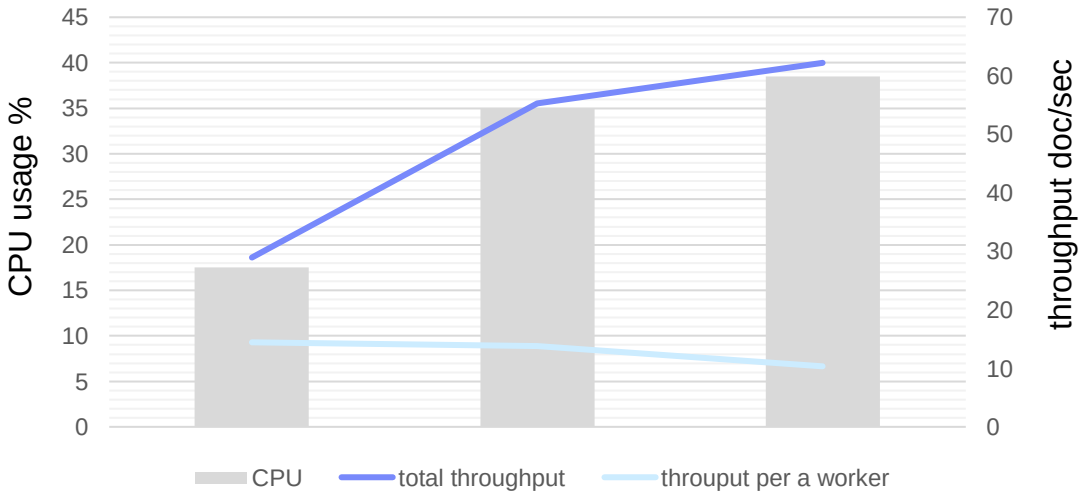
Validation results

parameter	default	pattern 1	pattern 2	pattern 3
The number of CPE servers assigned to sweep	(none)	1		
The number of total documents	0	60,000,000		
Properties used in Filter expression	(none)	1. testDate (this is a custom property) 2. storage_area_id 3. document title		
First column of the composite index	(none)	object_id		
The number of sweep candidates	0	288,196	12,000	
Maximum Sweep Workers	2	2	4	6
Policy controlled batch size	2,000	80,000	3,000	2,000
Results (with trace logging disabled)				
Search time [sec]		88	10	10
Move time [sec]		9,958	217	193
Throughput [doc/sec]		28.94	55.30	62.18
The ratio to 2 workers' throughput		1	1.91	2.15
CPU [%]		15 - 20	30 – 40	35 - 42

Results of testing

- Search time tends to be proportional to the total number of documents
  1.47 [min] / 60M [doc] * 500M [doc] = **12.2 [min]**

- Move time depends on the number of workers



Increasing the number of workers makes the total throughput increase, but the throughput per worker tends to decrease.

Lessons learned

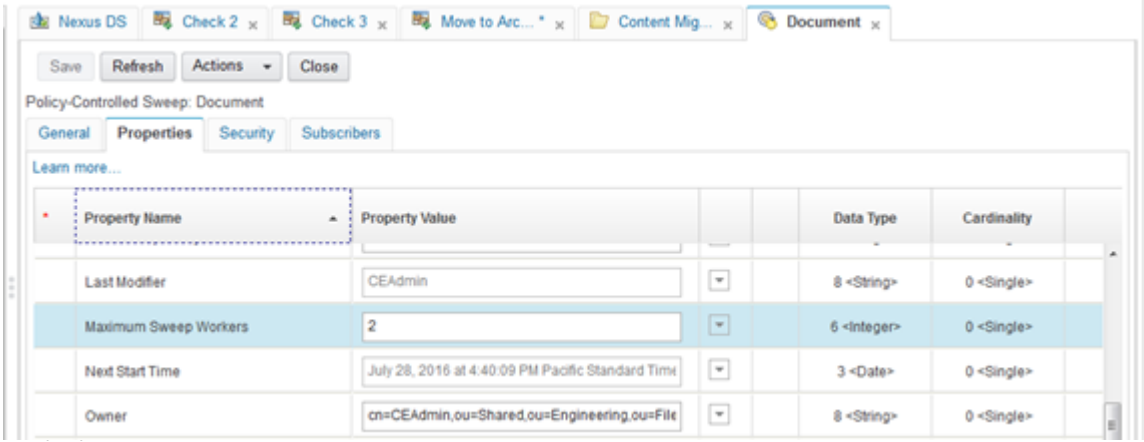
- To speed up the dispatcher's search time
  - Tune the filter expression and its composite index
  - Collect statistics and fix the execution plan
- The first column must be object_id or a property that efficiently narrows down the search results
- The composite index must have
  - All columns which are properties used in the Sweep SQL WHERE clause and SELECT clause
  - The order of the properties in the composite index must be same as the order of the properties in the filter expression
- Make sure the Oracle optimizer executes the SQL with your composite index
- Validating results
  - Reboot all middleware every time for each validation to clear cache
  - Take into account the size of the Oracle archive log
  - AWR and SQL reports are powerful analysis tools for search SQL

Agenda

- The Sweep Framework
  - What is a sweep
  - Framework features
  - Types of sweep
- Building a Sweep
- Gotchas
- Customer use case
- Performance Tuning
- Questions

Performance tuning

- Add a covering index – always do this
- Monitor the performance using the System Dashboard
- If appropriate adjust the number of workers assigned to a policy controlled sweep
  - There is always one dispatcher per policy controlled sweep
  - By default there are a maximum of two workers on each CPE server
  - To adjust the number of workers, edit the Maximum Number of Workers value on the Properties tab of the sweep policy



Performance tuning

- A covering index created on the table that contains the target objects for a sweep can significantly improve Sweep Framework throughput
- A covering index is a non-clustered index that includes all the columns referenced in either the SELECT clause or the WHERE clause of a particular query
- A covering index gains its advantage from the fact that all the information necessary to satisfy the query is contained in the index
- In this case the query is the one that is generated to execute a sweep for a particular sweep job or set of sweep policies that share the same base table

Performance tuning

- Columns to include when creating covering indexes for sweeping
  - The columns to include depend on the sweep type, target class and the filter expression
  - Always include:
    - object_id
    - home_id
    - security_id
    - epoch_id
    - recovery_item_id
  - Add to this the columns associated with any properties referenced in the filter expression
  - If Target Class is Document (or subclass), add:
    - security_folder_id
    - version_status
  - If the Target Class is Custom Object or subclass, add:
    - security_folder_id

Performance tuning

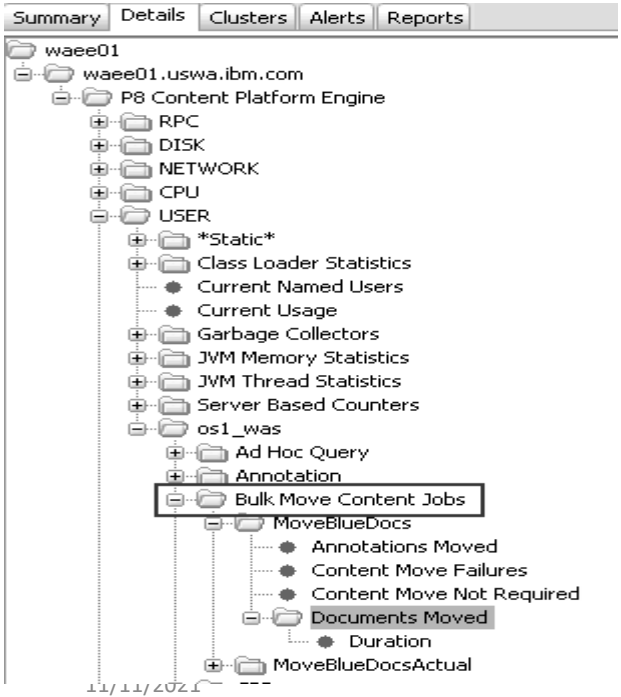
- Columns to include when creating covering indexes for sweeping (continued)
  - If sweep type is Bulk Move Content Job, add:
    - security_folder_id
    - version_status
  - If sweep type is Disposal Policy, add:
    - content_retention_date [sweep of Document class]
    - retention_date [sweep of any class except Document]
  - If sweep type is Retention Update policy or job, add:
    - content_retention_date [sweep of Document class]
    - retention_date [sweep of any class except Document]
    - If the Based Date Property Name is set on the policy, then the column related to the base date property is included. For example, if the Base Date Property Name is set to DateCreated, then the create_date column will be included in the selection list
  - If sweep type is Thumbnail Generation Job, add:
    - mime_type

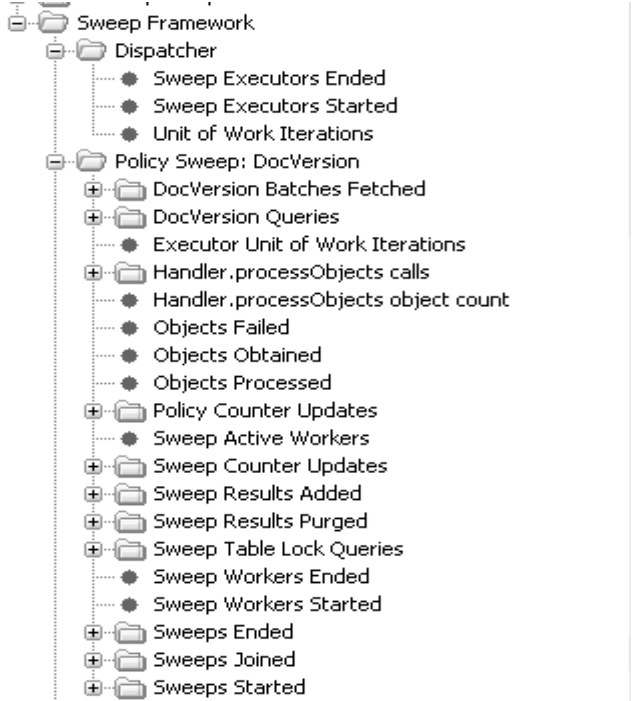
Performance tuning

- How to determine what columns are used in a query
  - Use CPE tracing to determine exactly which columns are included in the selection list
  - Enable Database detailed trace and Sweep Framework moderate trace, then run the sweep in preview counters only mode
  - Examine the trace, search for 'Sweep.SQL'
  - Locate trace statement like this:
    - …{EDCAF8D9-5DE3-413D-B35D-F7D5B10A59E4} Sweep.SQL: SELECT TOP 2000 bcn.Id, bcn.CmRetentionDate, bcn.DateCreated FROM Document bcn WITH INCLUDESUBCLASSES WHERE Id > {00000000-0000-0000-0000-000000000000} ORDER BY Id ASC FETCH FIRST 2000 ROWS ONLY OPTIMIZE FOR 2000 ROWS" In-bindings: ({00000000-0000-0000-0000-000000000000})
  - Follow the trace by thread id until you find the database query, the query will show all the columns included in the selection list.
    - …Executing SQL (0158C156): "SELECT epoch_id, object_id, content_retention_date, create_date, version_status, object_class_id, security_id, security_folder_id, recovery_item_id FROM DocVersion T0 WHERE (T0.home_id IS NULL AND (object_id > ?)) ORDER BY object_id ASC FETCH FIRST 2000 ROWS ONLY OPTIMIZE FOR 2000 ROWS" In-bindings: ({00000000-0000-0000-0000-000000000000

Monitoring performance – use the System Dashboard

The following sweep-related counters are available in the dashboard





Reference information

- More information on sweep is available here:
  - In the Knowledge Centre
    - Using Sweep Topic

      http://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc175.htm
    - Improving Sweep Performance Topic

      http://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.performance.doc/p8ppt237.htm

      Some clarifications on **CopyRetainedContent** property in the tech note
  - Tech note on copying retained content off Fixed Content Devices

    http://www.ibm.com/support/docview.wss?uid=swg27043967

Questions

Customer feedback

- Would like a Java Script tool kit to make it easier to develop the Action Handler
- By default, job sweeps and policy sweeps retrieve a database row before it evaluates the filter conditions for that row. Using the FilteredQueryTimeout property, you can optimize sweep performance. Documented here: https://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc178.htm

  By default the timeout is set to 0 and not shown on the wizard. Generally it makes sense to create job or policy as disabled, set that property on Properties tab and then enable the job or policy.

  Also look at this topic:
  https://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc203.htm

- FDM doesn't fully support import/export of sweep jobs and polices. When building Export manifest it allows to select only Sweep Policies. When policy objects are exported, associated Custom Sweep Actions are also automatically exported.
  - It would be nice if FDM supported exporting Sweep Jobs as well. Often you create and test Sweep Job in a DEV/TEST environment and want to do same in PROD. If you could put an enhancement request for this - it would be great.
  - We found a workaround to this, but it's somewhat of a hack. Basically, you can create a Sweep Policy and export it with FDM. Then go into generated XML file and change class name and ID(s) to correspond to CmSweepJob class. FDM then can actually properly import this XML and create Sweep Job instead of Policy.

Questions

- Can I use the sweep framework to delete folders – will it automatically unfile the content in the folder?

  Yes, you can use the sweep framework to delete folders, and it will unfile any content.

Questions

- **Why does sweep examine all objects in search query**

  The filter condition completely determines which objects are acted upon. The only thing that differs is how that filtering is accomplished:

  - By default, all instances of the relevant class(es) are retrieved from the database (essentially by SELECT * with no WHERE clause) and are then filtered in memory
  - If the Filtered Query Timeout is set to a non-zero value, that part of the filter expression which is valid in an ad hoc search is "pushed down" into the WHERE clause of the database query. If the whole filter expression can be pushed down, the result of the database query would be exactly the objects that would be acted upon, but otherwise further in-memory filtering is required to yield the final set of targeted objects.
  - An example of a filter clause that cannot be pushed down is Owner='foo@bar.co.uk', because the Owner property is not queryable. So if the whole filter expression was, say, VersionStatus=1 AND Owner='...' then the database query would be issued with just the VersionStatus condition and the Owner part would be applied in memory.
  - The basic principle is that by pushing the filter down into the database query, we're getting the DB to do more of the work, reducing the amount of data (number of rows) passed back from the DB to CPE and the amount of processing the CPE has to do.
  - Once filter pushdown is enabled, there's nothing different here than optimizing any other search through database tuning/indexing. Unoptimized, the pushdown query is more than likely to be a table scan or at best a range scan on the primary key index, so it is going to make the DB work pretty hard (over anything other than a fairly trivial number of rows). Adding a covering index can make it go a lot faster, with the tradeoff being the usual one of index maintenance cost.
  - Using the FilteredQueryTimeout property, you can optimize sweep performance. Documented here: https://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc178.htm
  - By default the timeout is set to 0 and not shown on the wizard in ACCE. You can also use this property to limit the number of times a sweep is run in a day too.

Questions

- Can you have Sweep dedicated to particular server
  - While we don't really support this. You could achieve this by disabling the sweep dispatcher on a CPE server – but this would prevent all sweeps running on that server.
  - The sweep framework has been designed from the outset to take maximum advantage of a system with multiple servers, exploiting that to provide resilience and scalability. It does that by processing the candidate objects in chunks, retrieving say 500 objects at a time, filtering them and acting upon, then saving a “bookmark” of where it has got to. That allows another server to take over the next chunk starting from that bookmark and in some cases even to do that while the first server is still processing the previous chunk. This also provides a failover mechanism, should a server crash or be shut down mid-chunk, another server can take it over after a timeout has expired indicating that the chunk has been abandoned.
  - To limit impact on other processes, use the scheduler, limit the number of workers used, and find a suitable batch size that can be processed efficiently.

Questions

- How do you pause a sweep?

  To pause a sweep, simply set the IsEnabled property to false.

  - Sweep objects have a RecordFailures boolean property. If this is set to true, each failure will create a Sweep Result object which records the identity of the failing object and other diagnostic information. The result objects are linked to the sweep object through the Sweep Results property, and I believe ACCE will allow you to see them.

  - A move content operation copies the source content to the destination storage area in essentially the identical fashion as uploading new content. So for a fixed storage area it goes into the staging directory, for file storage into the inbound directory and for advanced storage directly to the final location(s) [plural if synchronously replicating]. Then on committal of the move content transaction the content will be finalized in the normal way.

Background Search Question

- Through ACCE can set up a background search, but cannot automate the re-running of the search. Instead, the timeslots just indicate when the one instance can be run….once all the objects have been returned, the search stops.
  - So for a long running query, the timeslots indicate when then search can be run until all the results have been returned.
  - Currently there isn't a way to re-run a background in automated fashion through the UI....so you'll have to set it up via the APIs instead and your own scheduler....see the following topics:

    https://www.ibm.com/support/knowledgecenter/SSNW2F_5.5.0/com.ibm.p8.ce.dev.ce.doc/backgroundsearch_snip3.htm

Sweep performance questions

- Why the bulk move sweep job only uses one CPE server.

  A sweep job guarantees that it will examine all objects that match the filter expression. To do that it can't advance to the next batch of objects until it has completed the current batch, so it can't distributed the processing (this is different than a distributed policy sweep, which makes continuous passes over the objects and can tolerate missing part of a batch since it will get to on the next pass).

  A single job sweep runs on a single CPE instance. A job sweep obtains a batch of objects and distributes the objects across multiple workers on the single CPE instance. If you need to distribute move content processing across all CPE instances, you can use a Content Migration Policy. You would need to manually monitor it to determine when all the content has been moved - then disable the policy.

- What difference is there between sweep job and sweep policy on how many threads are active?

  A sweep job is not distributed by design - I.e. it only runs on one CPE instance. The reason is that a job guarantees that it will handle all objects that match the filter expression. To be able to do that without maintaining state somewhere, it can't process batch #n+1 until batch #n is done. On the other hand, policy and queue sweeps distribute the sweep workload across all CPE instances that have a sweep dispatcher enabled. (I'm aware that this isn't documented, and we're working on getting the documentation updated).

Sweep performance questions

- Does reducing the FilteredQueryTimeout value improve performance.

  This will only help if the queries are taking longer than the timeout value. To determine if it's the sweeping that's slow, or accessing the source that is slow, you can collect Sweep Framework Moderate tracing. Look for lines in the trace like '[sweep name] obtained n objects from [class-name] in n milliseconds'. This will tell you how long the core sweep query is taking. You can also look for the time between these statements, to get an idea of how long it's taking to process the objects of the batch (i.e. how long to perform the move content).

- The acting on behalf of the initiator has been one of things folks have tripped over in general. So, I was wondering what would happen if the document had been locked down so that only the owner had access and the owner is no longer in the LDAP -- there is no way for that document to be "tidied up"....correct?

  That situation can be correctly manually, by someone with WRITE_ANY_OWNER permission to the object store, by first taking ownership of the document, then adjusting the ACL as necessary. I'm not proposing that the security sweep should mimic that manual behaviour, it would just produce an error record for such a document. (Same goes for the disposal sweep).