HP TRIM 7.2 Records Authority Integration Using SDK

A practical guide for developers

Technical white paper

Table of contents

Introduction

This developer guide helps you integrate with the Records Authority feature of HP TRIM 7.2 or later. The Records Authority feature allows you to take advantage of TRIM's rich disposition management features to safely and legally dispose of documents stored in separate document archive systems.

The feature allows you to create a single record in a TRIM Records Authority database and use it to manage a whole collection of items stored in a separate system. When the TRIM record's retention schedule triggers its destruction it will generate a delete job for all the externally archived items, execute the deletion, and store a detailed audit trail of it back into the TRIM Records Authority database.

In order to implement the feature within your archive system, you need to develop code to integrate with the TRIM Records Authority function. This involves developing software to support:

1. Tagging of the items in the external archive system. This involves making a call to TRIM to request an authority record and then assigning its records authority ID to the items.
2. Deleting items when record destruction is due. This involves writing a deletion component implementing the ITrimDeleteJobAddIn interface, which will be called by the TRIM event processor.

Archive System Requirements

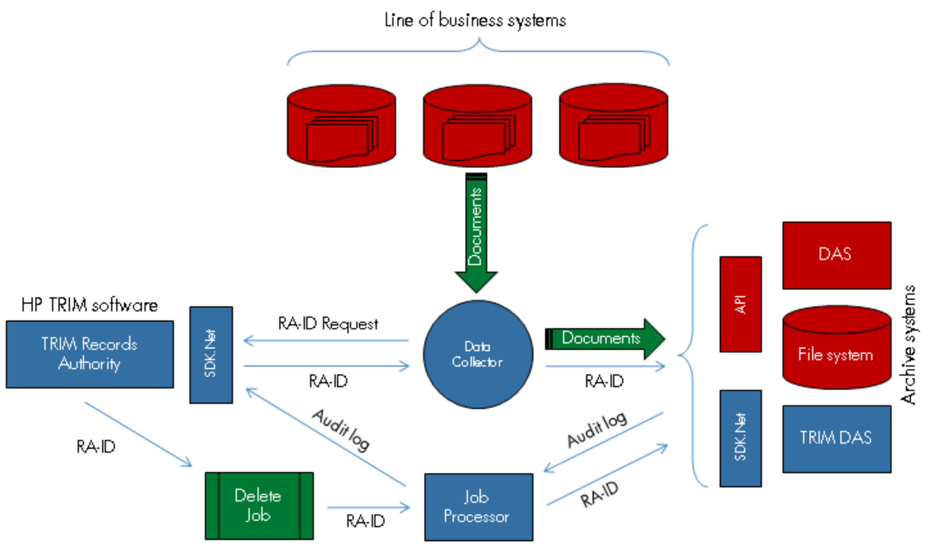
In order to support the TRIM Records Authority feature, your archive system must fulfil the following requirements:

- Tag each archived document with a records authority ID (50 character string)
- Search for and find documents based on this records authority ID
- Prevent documents from being deleted unless instructed by the records authority deletion component

Architecture and Basic Concepts

Figure 1 shows the architecture of a records authority implementation where the tagging of items is performed as part of the data collection from a document source at the time of moving the documents into an archive system. Instead of tagging the documents as part of a move you could, of course, also assign the records authority ID to already stored items identified through a query.

Figure 1: Records Authority Architecture



Architecture Components

TRIM Records Authority System (RAS)

This is a TRIM dataset that has been configured to manage authority records that, in turn, manage externally stored items. This involves the creation of retention schedules to manage the disposal, record type(s) to enable creation of authority records, external system objects to maintain the connectivity to the externally stored items, and records authority origin policies that define when and how a record needs to be created for a particular data source.

Document Archiving System (DAS)

A document archiving system is a system for storing documents for a period of time. For the discussion, it is assumed that it provides the features outlined in the introduction. Typically, the system will have a minimal set of metadata and usually has the purpose of reducing storage cost. The intent of the Records Authority feature is to bring a legally compliant process along to assist with the deletion of documents from the archive. In the remainder of this document, a general Document Archiving System is referred to by the acronym DAS. In implementing the Records Authority functionality, we decided to also add new features to TRIM to allow it to behave as a DAS. A TRIM database configured in this way will be referred to as a TRIMDAS in this paper.

Data Collector (DC)

A data collector is a general name given to a process that stores documents within a DAS. This guide outlines how to configure a system based on the idea that it will be applied to new documents entering the system. Adapting existing systems with documents in situ is also possible, but is not discussed here. It is presumed that enough details are provided in this guide to enable you to proceed further into designing that adaptation.

Job Processor Add-In (JPA)

This is a software component that responds to delete requests generated by the RAS. TRIM will create a new delete job whenever an authority record is due for deletion. The TRIM event processor implements a Job Processor which reads and executes the delete jobs—this is done by making a call to an external software component that implements the TRIM ITrimDeleteJobAddIn interface (published in the HP TRIM Software Developer's Kit (SDK)). Essentially, you need to develop a software component that implements this interface to ensure that items are deleted from your particular DAS.

Sample Document Flow

Documents will flow through system as follows:

- Documents are identified by the DC for entry into the system
- DC requests an authority record from RAS (providing a records authority ID)
- DC adds the documents to the DAS, tagging each with the records authority ID of the authority record

At this stage, there is a record in the RAS that represents the documents in the DAS. Typically, and key to the design goal, the same records authority record could be returned for several continuous requests by the DC; this behaviour can be configured as part of the records authority origin policy. Once stored, the DAS should prevent the documents from being deleted if they have a records authority ID attached.

The RAS is now responsible for calculating when documents should be deleted by applying rules to the authority record. When it determines that an authority record is due to be destroyed, it assumes that the documents under its authority should be deleted as well. There is one proviso to this assumption—the records authority record types can be configured for individual destruction, which will cause the delete job to include additional date-based selection criteria for the deletion of the documents.

Continuing the flow:

- The RA determines that a records authority record is due for destruction
- The RA generates a delete job including the records authority ID (RA-ID) and optional date-based selection criteria, as well as the connection details for the DAS
- The Job Processor executes the delete job by initiating the JPA for the DAS
- The JPA finds the documents in the specified DA and deletes them
- The JPA logs the destruction to a file which is stored back into the RA by the Job Processor
- Once all documents for an authority record have been deleted the record is marked as "destroyed"

Configuring the TRIM Records Authority Dataset (RAS)

As part of the initial setup of TRIM to provide RAS features, you need to undertake the following steps:

1. Register the DA system(s) that the TRIM RA will support. This can be done using the TRIM user interface (see **Tools – HP TRIM Administration – External Document Archive System**). Each system needs to have a unique ID (this will also be passed to the RAC component), a description, and three optional query URLs. The main query URLs is used in the TRIM UI to provide a hyperlink capability to display DAS items under a TRIM authority. The URL should contain the substitution string "%authorityID%", which will be replaced with the ID of the currently selected TRIM records authority record within the TRIM UI. The two other URLs provide an additional substitution parameter of "%date%" to allow the display of a subset of records based on their being older than the specified date. Depending on your process flow, you may decide to register the DA system through the SDK as part of the DC integration.
2. Create one or more record types that implement the Records Authority behaviour. The record type also determines whether the individual deletion option will be used when notifying the DA of 'due for destruction' events.
3. Optionally, you can set up one or more Origin objects to record policy information to determine how records authority records are created or selected. The 'Origin Type' of Records Authority provides the following policy parameters:

    - **AuthorityTitleStub**: This is some text that will be added to the start of the title when creating a records authority record.
    - **AuthorityStyle**: An enumeration indicating how records authority records should be allocated when using this policy. The options are to simply create a new authority record every time you run an import, or to re-use the same authority record over and over again (with some options for automatically creating a new one; see rollover items below).
    - **AuthorityRecord**: This property is used when the policy is to re-use an existing authority record.
    - **RolloverDays**: When re-using an existing authority record, you can optionally specify to automatically create a new authority record whenever the date registered of the current authority record is more than a number of days in the past.
    - **RolloverItems**: When re-using an existing authority record, you can optionally specify to automatically create a new authority record whenever the managed item count of the existing authority record exceeds the entered number.
    - **NeedsAuthorisation**: You can turn on a flag to indicate that before any items get deleted from the DAS, someone needs to "Approve" the deletion manually.
    - **Authoriser**: If you turn on the `NeedsAuthorisation` flag, you also have to specify the person who will be responsible for authorising the deletion of the DAS items.

    The Origin also provides a special method, **GetAuthorityRecord**, to setup the records authority record according to the above policies. Using this policy approach, an archive system co-ordinator or records manager can specify a number of policies for Records Authority, thus allowing the developer of the DC system a simple task of selecting the appropriate Origin for the run.

Configuring TRIM as a Document Archive System (DAS)

TRIM can also be configured to act as a Document Archive System. Only an SDK application can create externally managed documents in a TRIMDAS, as the SDK is the only way to specify the RecordAuthorityID for a TRIM record. Once a TRIM record has a RecordAuthorityID attached (you can see this in the TRIM UI by displaying the External Records Authorities property of a record), TRIM will safeguard the record from being deleted or destroyed in the normal course of events.

To ensure that the Job Processor can undertake the authorised deletion of these records, a special user needs to be created in TRIM. This user should be setup as an Administrator, but also have the extra permission of "Archive Externally Managed Documents". Using the Administrator as a template, create a user as a "Custom User" and add this extra permission. The JPA component will need to connect as this user in order to be granted the permission to delete these "externally managed" records.

When registering a TRIMDAS as an external archive system in the TRIM RAS, you will notice that there are extra parameters available to allow you to configure which account will be used to connect to the TRIMDAS database and additional settings to assist in formulating the Query URLs. These parameters are:

- **Workgroup**: This is the name of the TRIM Workgroup Server to use to connect to the TRIMDAS dataset.
- **WorkgroupPort**: This is the TCP/IP port number to use for the TRIM Workgroup Server connection.
- **BackupWorkgroup:** The name of a fallback Workgroup server to use if the main Workgroup is not available.
- **BackupWorkgroupPort:** The TCP/IP port number of the fallback Workgroup server.
- **DatasetId:** The two-character TRIM dataset ID of the dataset which will be acting as the TRIMDAS.
- **DatasetName:** The descriptive name for this dataset.
- **WebClientURL:** The root URL for the TRIM Web Client that has been configured for the TRIMDAS system (optional). If provided, the TRIM SDK will automatically setup the appropriate default values for the query URLs based on this root.
- **DateFromMapping:** Indicates which TRIM record property to use when resolving a "delete all records with DateFrom older than a specified date" request.
- **DateToMapping:** Indicates which TRIM record property to use when resolving a "delete all records with DateTo older than a specified date" request.

The JPA for TRIMDAS is included in the standard `HP.HPTRIM.SDK` assembly file. Simply make sure that the **AssemblyPath** and **AssemblyPathType** properties of the external archive system definition point to a valid installation of the `dll`, so that the Job Processor can initialize it when it needs to run a delete job.

Developing the Data Collector (DC)

The DC needs to be able to create a records authority record in the RA and then store one or more documents in the DAS, ensuring that each of these documents has the correct records authority ID. This component is not part of the standard TRIM product; the HP TRIM Software Developer's Kit (SDK) provides a sample application that acts as a DC.

A DC will generally involve moving a set of documents or other items from a source location into a DAS; along the way requesting an authority record from the RAS to manage the eventual destruction of the DAS documents. It is expected that the DC will use an Origin (`type=RecordsAuthority`) from the RAS, which stores details of how the authority record is created in the TRIM RAS.

Similarly, configuration information useful for connecting to the DAS needs to be stored in an ArchiveSystem object in the RAS—using the ConfigurationString property for custom archive systems, or the special settings described above for a TRIMDAS. Whether you set the ArchiveSystem object for an import at run-time or whether you retrieve it based on prior RAS configuration depends upon which component you regard as the authoritative source of this information. In either case, make sure that the authority record is linked to the correct ArchiveSystem object in order to maintain the link between it and the managed items.

To create the TRIM authority record, you first select the appropriate Origin object to match your data source. This could be done by reading a saved setting from a configuration file matching on, for example, Origin Name, or perhaps selected by the user—the implementation details are up to you. Indeed, you don't have to use the Origin policy feature and could simply create a record based on a record type that implements the Records Authority behaviour. However, using the Origin.**GetRecordAuthority** method does provide some highly desirable features and is therefore recommended.

An additional feature in the RAS is to display some status counters (Items Managed/Items Destroyed) for the external DA document set. To enable this to work, the DC must notify the TRIM authority record of items added to the DA using the Record.**UpdateAuthorityItemsCount** method.

Once the TRIM RA record has been created, you obtain the records authority ID from this record using the Record.**RecordAuthorityID** property. This is a string that uniquely identifies the record, and this string should be attached to all the documents that are added to the DAS that need to be managed by this authority record. Note, when using TRIM as the DAS, you can associate the RecordAuthorityID with the TRIMDAS record by using the Record.**SetRecordAuthority** method.

Developing the Job Processor Add-In (JPA) for a DAS

The JPA is really the key component in the Records Authority processing and is responsible for acting upon the delete instructions provided by the RAS to the Job Processor.

TRIM implements a Records Authority Event Processor which needs to be configured to run on the RAS dataset using TRIM Enterprise Studio. This event processor will examine each authority record in the RAS to determine if its destruction due date has been reached. If so, it will create a new Job object based on that authority record. This job entry will have all the details needed to instigate deletion action within the DAS. A separate Job Processing Event Processor, also configured in TRIM Enterprise Studio, will then process each job, loading the JPA .NET assembly that has been defined in the ArchiveSystem object for which the job has been created. Note that if the destruction requires approval, it will be marked accordingly and will require the Approver to log into TRIM and approve the job before it can be processed. (An email will be sent to the person responsible notifying them that a new job awaits their approval)

For TRIMDAS implementations, all the basic JPA functionality is part of the `HP.HPTRIM.SDK.dll` and you would only need to write a customized JPA if you wanted to implement a more sophisticated processing model, e.g., with asynchronous or remote deletion of items in the TRIMDAS.

Job Processor Add-In

For any other DAS, you need to write a .NET assembly that implements the ITrimDeleteJobAddIn interface. If your DAS API is in a different language, you may need your .NET assembly to connect to it via Web Services. In either case, the TRIM Job Event Processor will only be able to initialize .NET assemblies.

The **ITrimDeleteJobAddIn** implements the following methods:

- **Initialise:** is called the first time the event processor loads your assembly prior to issuing any StartDelete calls. The Database object parameter is the TRIM Records Authority database for which the event processor runs.
- **StartDelete:** is called when new delete has arrived in the job queue. It is expected that you will commence an asynchronous process which will then examine items within the archive system and delete them if they match the job delete criteria. It is anticipated that deletes may take some time to execute and may even need to be restarted. For this reason, you should check to see if the Delete job is not currently running. The event processor provides a handle for this job which is used for other calls requesting status and other information relating to the job, which you need to use to execute deletion. See the section on "Executing the Deletion" below.
- **GetItemsAtStartOfRun:** is called by the Job Controller to get a counter of items that were associated with this job when the job was commenced. It is important to gather this information when your delete job starts as this will help with troubleshooting any failures during processing, as well as with calculating the number of items that have been deleted.
- **GetDeleteStatus:** is called by the event processor to get an update to the job status for running jobs. If you notify the controller that the function is completed, the event processor will call back to get the log file and a completed items count, and use these values to update the TRIM authority record properties.
- **GetErrorMessage:** should return a reason for failure message if GetDeleteStatus returned a failure.
- **GetItemsAtDeletedDuringRun:** is called by the Job Controller to get a counter of items that were deleted by this job during processing. If your process doesn't keep a running counter you could use a query to count the remaining items and use the delete job's **ItemsAtStart** property to calculate it.
- **GetLogFileName:** is called by the job event processor when a job has been completed. The event processor will save the log file back to TRIM. The format of the log file should be detailed enough to provide sufficient evidence of the deletion according to your company's records management policies.
- **ReleaseJob:** is called by the event processor after receiving a failure or completion notification. It indicates that it will no longer poll for status on the job or ask for associated properties such as a LogFile or an ErrorMessage. You can therefore release any memory currently associated with this job.

- **Shutdown:** is called when the event processor is about to exit and wants all add-ins to commence shutting down any threads. It will be called regularly until it returns a value of "`true`", indicating that all threads are now closed.

You must implement all of these in your assembly. How you implement the underlying code is up to you and is largely dictated by the capabilities of your DAS and its APIs.

Executing the Deletion

Process Overview

Once the `StartDelete` method in your add-in is called you need to go through the following steps:

1. Persist the job handle so that you can report the status back at any time (even after a crash)
2. Open a log file for the deletion job
3. Connect to the DAS
4. Execute the query to find all items with the records authority ID for the job, keep a count
5. If necessary, refine the query with date-based criteria to identify all items that actually can be deleted
6. Delete each item, logging the outcome
7. Keep a count of deleted items
8. Close the log file
9. Return a value of "`true`"; if you do asynchronous processing you may want to return a value of "`true`" right at the start of the process. In that case, you can return a status of "`failed`" when the event processor calls GetDeleteStatus. In this case, make sure that you have an error string to return to the GetErrorMessage call.

The TRIM Job object that is passed to you on the StartDelete call contains several properties that you need to use in order to execute the deletion when your JPA's StartDelete method is called:

Connecting to the DAS

Job.**ForRecord** contains the Record object for which the delete job has been created. You need to know this because the Record.**ArchiveSystem** property contains the details of the DAS and how to connect to it.

Building the Query to Identify Items Managed and Items to be Deleted

Job.**DeleteDateType** indicates whether you need to include date-based criteria in your query to identify items to be deleted. NoDateCheck indicates that you can query just by the records authority ID, CheckFromDate and CheckToDate require you to include date-based criteria.

Job.**ForRecord** also contains the Record.**RecordAuthorityID** property which allows you to identify all the items managed by the record for which the job was created. You need this to be able to return the count to the GetItemsAtStartOfRun call.

You can also use the Record.**ArchiveSystem** property to get the details of the DAS, in particular the ArchiveSystem.**DateFromMapping** and ArchiveSystem.**DateToMapping** to build the date-based part of the delete query. The Job.**DeleteMonths** contains the number of months the "mapped" date needs be in the past in order for the item to be eligible for deletion.

For More Information

To read more about HP TRIM go to http://www.hp.com/go/hptrim.

To learn more about the HP TRIM SDK refer to the document TRIM7.2_.NetSDK.pdf, available on the installation media or the support portal.

4AA3-9158ENW, Created December2011