Prudential – Alliance Call Recording Archive

Author: Clifford Grimm

Date: 10/28/2014 10:31 AM

Revision: 1.1

DISCLAIMER: This document alone is not a formal contract between HDS and Prudential, but is intended to describe the proposal for the deliverable.

# Table of Contents

1 Introduction 1

2 Delivery Overview 2

2.1 Data Layout 4

2.2 Object Metadata 5

2.3 Content Class Definition 7

2.4 HCP Setup 7

2.5 Search Interface 7

2.6 Processing Efficiencies 8

3 Questions/Issues 9

4 Future Considerations 9

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Revision** | **Description** |
| 10/28/2014 | Clifford Grimm | 1.1 | Minor updates based on actual implementation. |
| 10/2/2014 | Clifford Grimm | 1.0 | Added information about name of the wav files. And added Channel/Server to the metadata. |
| 10/2/2014 | Clifford Grimm | 0.2 | Fixing up with comments from conference call. |
| 9/30/2014 | Clifford Grimm | 0.1 | Initial Revision |

# Introduction

Prudential has approximately 1TB of Alliance call recordings consisting of approximately 700,000 WAV files to be stored on HCP for long term storage. Along with the storage, the desire is to be able to search on metadata associated with the call recordings.

The call recordings were originally created via the CMX recording software. The call recordings are being extracted from the CMX system and deposited onto an external drive. The content will be ingested into HCP along with its metadata created by the extraction to enable the search capabilities.

This document will outline the proposed approach for HCP ingestion, call metadata addition, and search capabilities.

# Delivery Overview

The delivery will be a joint effort between Prudential and HDS. As an overview, Prudential will be responsible for transferring the call recordings to HCP with guidance from HDS. HDS will be responsible for adding metadata to the call recordings to allow for robust search and providing basic training for use of the search capabilities.

The following picture is an overview of the approach.



Figure 1: Delivery Overview

With guidance from HDS, Prudential will move the content from the external Drive to an HCP Namespace using the HCP-DM tool. HCP-DM is a multi-threaded application that can migrate content to HCP in the most efficient manner. Prudential will be responsible to ensure that the content on the external drive has been correctly written to HCP using any level of content verification they deem appropriate.

The migration may take a week or more to accomplish as the expected bandwidth of the whole process will likely be limited by the external drive. In default mode, HCP-DM executes 20 threads per HCP node to achieve the most efficient HCP ingest; all the HCP-DM threads will be accessing the external drive at the same time causing excessive disk thrashing. It might be beneficial to throttle back HCP-DM to only have a few simultaneous threads.

The layout of the content on HCP will be unchanged from the external drive. The full form is described in the Prudential Interface Control Document (ICD) named “ICD\_OSI\_CXM\_CallRecordings\_Extract\_vs1.0\_2.pdf“. To summarize, there are two folders inbound and outbound. In each of these folders, there will be the WAV files and a single XML file. The XML file contains the metadata to be stored with the objects in HCP as custom metadata.

HDS will develop specialized modules for Prudential that will understand how to retrieve the custom metadata based on the content uploaded to HCP from the external drive. These specialized modules will be used with a general purpose tool called COMET (Customizable Metadata Extraction Tool). This tool has the logic for iterating over content, calling the extraction module, and writing the generated custom metadata to HCP.

Since the content will have been migrated to HCP by HCP-DM, the COMET tool will access the content in the HCP namespace using NFS from a Linux system provided by Prudential. COMET will walk the content using the NFS protocol and add the custom metadata via the HTTP REST interface.

During the addition of metadata, HDS will monitor the progress via remote access provided by Prudential.

Final analysis will be performed by HDS and information will be provided to Prudential:

1. Ensure that all WAV files have Custom Metadata. Spot check metadata for some WAV files to ensure it is what is expected. Any WAV files without custom metadata will be researched and root cause determined.
2. Once indexing has been complete, perform searches against generated Custom Metadata to ensure proper indexing.
   1. Perform a date based search on date/time fields.
   2. Perform search based on other field like CallID, Agent Name, etc.

For validation purposes, HDS will provide all log files produced by the COMET tool and provide an HCP generated query result providing evidence of the expected WAV files have Custom Metadata attached.

A brief training session will be provided either via WebEx or on-site for the core consumer of the HCP search capabilities and can occur any time after some content has been indexed. The expected duration of the training is likely to be about an hour. The training will cover the basics of performing searches using the HCP Search Console against the actual CMX recording content. If necessary, the training may also consist of developer focused training for utilizing the API to perform content searches.

## Data Layout

The layout of the content on the external drive is very flat. It consists of two folders at the top-level: *InboundCalls* and *OutboundCalls*. There is approximately 624,000 inbound calls and 83,000 outbound calls. Accompanying the WAV files in each folder will be a single XML file containing the call information for all the WAV files.



Figure 2: Folder/File Layout

Although the layout is not optimal given there is essentially no folder structure, it does not reach the HCP number of objects per folder guidelines. Therefore, the content will be written in the same folder layout on HCP. While there is no expectation that this layout will impact the performance of HCP for search and retrieval, there is expected performance impacts performing and processing full folder listings if that is desired for later processing.

The WAV files have a generated name that consists of the following layout:

MMddyyyyHHmmssCCCBB.wav

where,

* MM – month 01-12
* dd – day of the month
* yyyy – four digits of the year.
* HH – hour of the day in 24 hour time (0-23)
* mm – minutes in the hour (00-59)
* ss – seconds for the time (00-59)
* CCC – CXM channel number
* BB – CXM box number

## Object Metadata

Each WAV file will have metadata associated with it. The metadata will be used to provide a search index for the WAV files. The metadata will be extracted from the XML file residing in the same folder for the WAV file.

The following table outlines the XML schema that will be used to add metadata to the WAV files in HCP:

|  |  |  |  |
| --- | --- | --- | --- |
| XML Tag | Description | Sample Value(s) | Notes |
| <CallRecording type=”wav” > | Top level tag with call recording type. |  | Will always have the type attribute with the type of “wav”. |
| <CallID> | CMX Call Reference Number | 1381068105 | [Integer] |
| <Channel> | CMX Channel Number of call | 432 | [Integer] Derived from the 14-16 digits of the file name. |
| <Direction> | Direction of call initiation. | Inbound Outbound | [String] Derived from file path. |
| <DateTime>  <Start> | UTC Date/Time of start of call | 2014-08-04T12:36:58-0000 | [DateTime]  yyyy-MM-dd HH:mm:ssZ Input is in EST/EDT |
| <DateTime>  <End> | UTC Date/Time of end of call | 2014-08-04T12:37:48-0000 | [DateTime]  yyyy-MM-dd HH:mm:ssZ  Computed from start time and duration from WAV file. |
| <DialedNumber> | Dialed Phone Number | 9785551212 | [String]  For Outbound calls only. Derived from Phone tag in CMX metadata. |
| <ANI> | Customer Phone Number | 80080000101 | [Integer]  For Inbound calls only. |
| <DNIS> | Dialed Number | 4044 | [Integer]  For Inbound calls only. |
| <Duration> | Duration in minutes | 72 | [Integer]  Generated from file. |
| <Agent> | Agent Name | Smith, John | [String] Create first/last content property. |
| <Server> | CMX Box Number | 32 | [Integer] Derived from the last two digits of the file name. |
| <Skill> | Agent Skill | Sales | [String]  For Inbound calls only. |
| <Extension> | Agent Extension | 6527 | [Integer] |

The XML scheme shown here is not exactly the same as the XML of the source content. The reason for the difference is to blend the prior NICE call data fields with the new fields from this project. The following are the differences:

* Top tag: <Recording> changed to <CallRecording> to match prior XML top tag.
* <DateTime> extended to enclosing <DateTime> with child <Start> to match prior XML structure.
* Added <DateTime> with child <End> to be computed from call start time and WAV file duration information.
* <WAVPath> dropped as it is available as HCP system metadata as utf8Name/ObjectName.
* Added <Direction> to indicate whether it was Inbound or Outbound. Derived from file path and will help simplify search by end-user.
* The <AgentId> in the source metadata file will always be blank and it will not be carried forward to metadata in HCP.

<CallRecording type=”wav”>

<CallID>#######</CallID>

<Channel>###</Channel>

<Server>##</Server>

<Direction>**Inbound**</Direction>

<DateTime>

<Start>yyyy-MM-ddTHH:mm:ssZ</Start>

<End>yyyy-MM-ddTHH:mm:ssZ</End>

</DateTime>

<ANI>####</ANI>

<DNIS>####</DNIS>

<Duration>####</Duration>

<Agent>Last, First</Agent>

<Skill>AAAAAAA</Skill>

<Extension>#####</Extension>

</CallRecording>

Figure 3: Inbound Call Metadata Template

<CallRecording type=”wav”>

<CallID>#######</CallID>

<Channel>###</Channel>

<Server>##</Server>

<Direction>**Outbound**</Direction>

<DateTime>

<Start>yyyy-MM-ddTHH:mm:ssZ</Start>

<End>yyyy-MM-ddTHH:mm:ssZ</End>

</DateTime>

<DialedNumber>#########</DialedNumber>

<Duration>####</Duration>

<Agent>Last, First</Agent>

<Extension>#####</Extension>

</CallRecording>

Figure 4: Outbound Call Metadata Template

## Content Class Definition

For prior NICE call recording upload, a CallInfo content class was created to define how to index the call recording metadata. This content class will be extended to have the following definition:

|  |  |  |  |
| --- | --- | --- | --- |
| Property Name | XPath Definition | Data Type | Notes |
| Agent | /CallRecording/Agent | String | Alliance Only |
| AgentFirstName | normalize-space(substring-after(/CallRecording/Agent, “,”)) | String | Alliance Only. *Generated*. |
| AgentLastName | normalize-space(substring-before(/CallRecording/Agent, “,”)) | String | Alliance Only. *Generated*. |
| AgentID | /CallRecording/AgentId | String | NICE Only. |
| ANI | /CallRecording/ANI | String | Alliance Only. |
| CallID | /CallRecording/CallID | Integer | Alliance Only. |
| Channel | /CallRecording/Channel | Integer |  |
| CLSCallID | /CallRecording/CLSCallID | String | NICE Only |
| DialedNumber | /CallRecording/Phone | String | Alliance Only |
| Direction | /CallRecording/Direction | String | Alliance Only |
| DNIS | /CallRecording/DNIS | Integer | Alliance Only |
| Duration | /CallRecording/Duration | Integer | Alliance Only |
| Extension | /CallRecording/Extension | Integer | Alliance Only |
| EndDateTime | /CallRecording/DateTime/End | DateTime |  |
| FileType | /CallRecording/@type | String |  |
| Logger | /CallRecording/Logger | Integer | NICE Only |
| Server | /CallRecording/Server | Integer |  |
| Skill | /CallRecording/Skill | String | Alliance Only |
| Session | /CallRecording/Session | Integer | NICE Only |
| StartDateTime | /CallRecording/DateTime/Start | DateTime |  |

Table 1: CallInfo Content Class Definition

## HCP Setup

The following is a list of configuration that will need to be performed on HCP as part of this delivery:

* Create a namespace in the existing tenant to host the Alliance call recordings. Search must be turned on for the namespace.
* Update the existing CallInfo Content Class to have the additional fields.
* Assign the Content Class to the Alliance namespace.

## Search Interface

HCP has a built in Search Console for performing adhoc searches. For this delivery, this HCP Search Console will be expected to be used by the end user. The user will enter in the criteria using the Structured Query tab and select the criteria from the drop-down list. Once a criteria has been matched, the user will be able to view the metadata about each call and click on the call WAV file to listen to the recording.

After delivery of call recording data with metadata, training via WebEx will be available to assist in the understanding of performing core queries against the call metadata. Advanced usage of the Search Console can be obtained from the online HCP documentation.

## Processing Efficiencies

Each XML file containing the call metadata contains approximately 700,000 records and do not appear to be in any sorted order. This will make for finding the metadata for a single call recording extremely resource intensive. The following are a few areas that can be focused on to help.

* Break up XML files into multiple files before processing. Each XML file will have a range of WAV files based on the WAV file name. Each XML file should probably have less than 50,000 entries per file; however, even this is a bit large. Will have to write a bit of software to process this. Will require special logic in module to know where to look based on WAV file name encountered.
* Disable sorting of items for the file system scanner in COMET this will require too much processing time with likely little gain.
* Configure multiple instances of COMET to work on a range of WAV files using the input file filtering. This will keep the COMET memory demands down as not every file will be stored in the internal arrays. However still concerned about how the Java file system utilities will behave encountering a folder with that many files in it.
* Consider having caching of the XML entries in the COMET XML generator. However, unless the items are processed in a sorted order, the hit ratio is likely to be low.
* **[IMPLEMENTED]** Convert XML file into ‘|’ separated value files. The converted file will be used by a specialized COMET scanner to provide the list of files to be ingested/updated. This alternative is to using the file system scanner that requires doing folder listing of all files. In addition, the scanner will store the metadata for each element for usage by the COMET generator code for creating the XML metadata.

# Questions/Issues

None at this time.

# Future Considerations

Provide a better representation of the data to break up the size of the folders and the XML files containing metadata.

While there is no anticipation that this flat folder structure will have an impact on the usage of search and retrieval as outlined in this document, any attempts at listing content in these folders via any supported protocol will result in extreme sized listings. Caution must be taken in avoiding doing any folder listing and may be excessively slow presenting and processing.