<div align="center">

Meeting Minutes
PWG MFD Semantic Model Face-to-Face Meeting
Dec. 11, 2007
At Dell Technology Briefing Center, Austin, TX

</div>

On-Site Attendees:

Attendee	Company	Email Address
Ron Bergman	Ricoh Printing Systems	Ron.Bergman@RPSA.RICOH.COM
Nancy Chen	Oki Data	nchen@okidata.com
Peter Cybuck	Sharp	cybuckp@sharpsec.com
Lee Farrell	Canon	lfarrell@cda.canon.com
Mike Fenelon	Microsoft	mike.fenelon@microsoft.com
Rick Landau	Dell	Richard_Landau@Dell.com
Harry Lewis	InfoPrint	harryl@us.ibm.com
Glen Petrie	Epson	glen.petrie@eitc.epson.com
Ole Skov	MPI Tech	osk@mpitech.com
Erhan Soyer-Osman	Microsoft	erhanso@microsoft.com
Jerry Thrasher	Lexmark	thrasher@lexmark.com
Dave Whitehead	Lexmark	david@lexmark.com
Craig Whittle	Sharp	cwhittle@sharplabs.com
Peter Zehler	Xerox	peter.zehler@xeroxlabs.com

Dial-in Attendee:

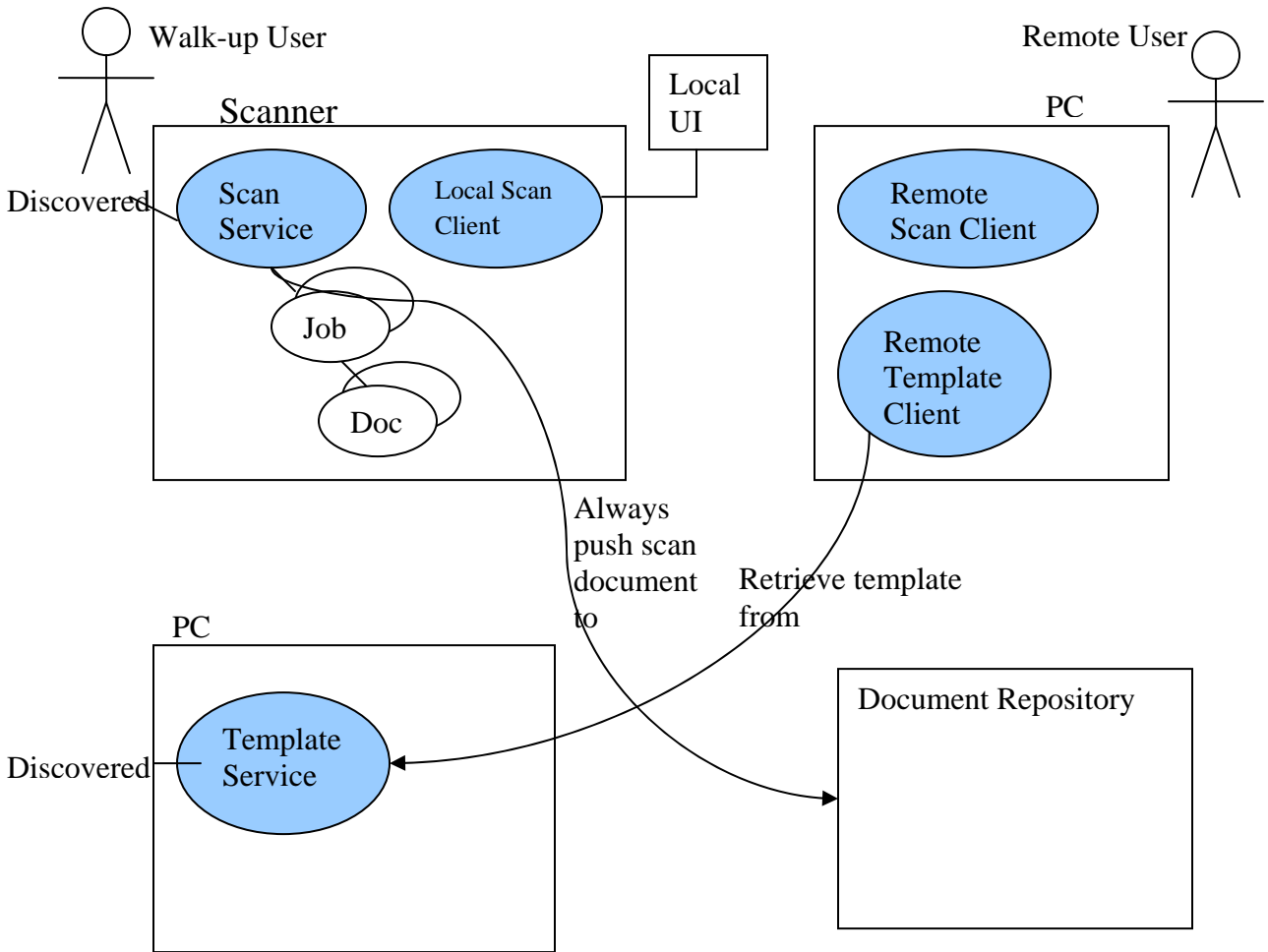| Ira McDonald	High North Inc.	blueroofmusic@gmail.com

1. MFD Chairman's Opening

MFD Working Group Chairman Peter Zehler welcomed attendees and provided the agenda of today's MFD meeting –
- o The purpose is mainly on flushing out the details of Scan Service schema and operations
- o The meeting will start with a briefing of Use Case scenarios, then walk through the Scan Service model schema, followed by review of operations.
- o No objection to the agenda.

2. Briefing of MFD Scan Service Use Case Scenarios

Nancy Chen briefed the attendees the eight use cases used for the Scan Service semantic model.

Peter Zehler drew a high-level overview diagram of the Scan Service framework that satisfies the requirements from these use cases:

Walk-up User

Remote User

Scanner

Local UI

PC

Discovered

Scan Service

Local Scan Client

Remote Scan Client

Job

Remote Template Client

Doc

Always push scan document to

Retrieve template from

PC

Document Repository

Discovered

Template Service

1
2
3   o   The Scan Service could be co-located physically with a Scanner, or connected remotely with a
4   Scanner via network.
5   o   A user could walk up to a Scanner or using a remote Scan Client remotely on a PC to use Scan
6   Service.
7   o   A template is an unbound Scan Job Ticket. One recent change in Scan Service model is we do
8   not allow the use of template by reference. To use a template, the template must be retrieved
9   from the Template Service, then copied into a Scan Job Request to be sent to the Scan Service.
10   The copy of the template can then be modified according to the user's intent.
11 o   The Scan Service and the template service must be able to be discovered.
12 o   The Scan Service has a sequence of jobs, each job contains 0 or more documents.
13 o   A document repository can exist in any number of locations:
14   o   A transport end point in a Client where application is ready to receive the Scan Document
15   o   Co-located with Scan Service
16   o   A ftp site
17   o   A full document management system
18 o   The destination of a Scan Job is identified in the Scan Job Request from the Client.
19 o   The Scan Service always pushes documents to a Document Repository.
20

- o The Local UI can provide user's inputs to the Local Scan Client or the Scan Service. When the Scan documents are to be stored in a document repository, Local UI can get user's credentials to the Scan Service for user authentication to the repository.

3. PWG Semantic Model Version 2

Peter Zehler briefly introduced top level elements of the PWG Semantic Model Version 2.
- o System – contains all kinds of services for a MFD. System counters are the roll-up of device counters, subunit counters, and service counters.
- o Subunit – each has an Id that's unique within the server (MFD).
- o Services in a MFD:
  - o Print Service: well-defined so far, taken from PWG Semantic Model version 1 – IPP.
  - o Scan Service is what we are working on right now.

4. Scan Service Model Discussions

(Note: highlighted in red are changes to the existing model or action items)

The PWG semantic model has multiple Scan Services in a MFD, each is analogous to a print queue in a given server.

A Scan Service has the following top level elements:
- o Scan Service Status : each Scan Service state is maintained by automata, must not be changeable by user or operators, but can be changed by administrative operations such as pause(), resume().
- o Scan Service Description : Descriptive information of Scan Service can be set or changed by user or operator.
- o Default Scan Ticket : the defaults for all the scan ticket information which holds job description, document and job processing instructions. Since there are multiple documents in a job, one can use per document processing instructions to overwrite job level processing instructions at the document level.
- o Scan Service Capabilities : map very closely to Scan Job Ticket, with elements in the same name but different syntax. The capabilities represent the allowed values for a given processing instruction element, and what processing elements are supported.

Scan Jobs are separated into two categories for easy retrieval of job information:
- o Active Jobs and Job History are analogous to Pending Jobs and Completed jobs in IPP.
- o Elements of Scan Job:
  - o Scan Job Status : Job state is maintained by automata, must not be changeable by users or operators.
  - o Scan Job Ticket : has descriptive information for the job, job processing instructions, document processing instructions for the documents within the job.
  - o There are multiple documents per job, but the initial implementation of a scan service needs not support multiple documents.
  - o Scan Job Receipt : represents what is actually done when the job is processed. Elements with unspecified value in Scan Job ticket will have the default values represented in the Scan Job Receipt.

Scan Service Status – contains the following elements:

- o Id : unique within a Scan Server
  - o IsAccptingJob: false whenever the Scan Service is disabled. The service still can accept queries, but no longer be able to accept jobs.
  - o State: the Scan Service working draft has a state diagram covers all states and state transitions. This is not modifiable by user or operator.
  - o State Reason : is an extension to state for additional information about why a job is in a particular state.
  - o State Message : gives localized information for state reasons. Issues:
    - o How is localization handled? This has not been specified in the Scan Service model right now.
    - o In SNMP, there is a locality that can be set, but there is no protection for two people setting the locality though.
    - o In Web services, HTTP allows to query what local language is supported, and local language to be used can be negotiated through HTTP request.
  - o Access Mode : user/group/world read/write/execute. They are part of operation information. This allows you specify some group operations under certain privileged level in a specific domain.
  - o Creation Date : of the service
  - o Current Time : the tick time of the service
  - o Natural Language : the local language currently used in the service (same as the locality of printer MIB)
  - o Owner URI : email / HTTP URI by which you can reach the admin or owner who created the service.
  - o OwnerVcard : owner's business card information
  - o QueueJobCount : number of jobs currently pending in the service, does not include jobs completed. The number matches the number of jobs in the ActiveJobs.
  - o ScanServiceCounters : used for Scan Service, Scan Jobs, and Scan Documents. Each document has its counters and rolled up to the job level counters. The name of this sounter in Scan Job and Scan Document will be changed to Scan Job Counters and Scan Document Counters using the ScanServiceCounter complex type.
  - o FullColorImages counter : Full color in scan is used to separate from black&white, highlight, and color that are used in printing.
  - Action Item : Owner URI and OwnerVcard are administratively set, should be moved to Scan Service Description.

Scan Service Description –
  - o Vcard + OwnerURI. should be here settable by Administrator – this needs to be corrected for WIMS too.
  - o Service type: this element is already in Scan Service Status : remove
  - o ServiceURISupported: the transport endpoint URIs the service supported, can have secure and non-secure URIs (e.g. HTTP, HTTPS…).
    - o This element is mandatory only for meeting LDAP mapping requirement. The service is required to advertise one or more URI to LDAP server for discovery.
  - o ServiceInfo and ServiceName : provide information about the service and user friendly name of the service.

o   ServiceSubunitSupported : this is a list of pointers to subunits that the service uses. The GetScanServiceElement will need to walk down the list to find out exactly what subunits are supported. Would rather to see what are the subunits used – i.e. a configuration element that identifies the actual instances of the subunits the service uses. No objection to use pointers for SNMP mapping.

Action Item: Add a Configuration element at the same level as Scan Service Capabilities, Descriptions, DefaultScanTicket. This will be service specific subunit configuration.

Scan Service Capabilities –

This is the way the service reports what allowed values are for the ScanJobTicket elements. Element names are the same as that in the ScanJobticket, syntax is different. It has the following elements:
o   Scan Document Capabilities -
   o   NumberUp : Number of scanned pages to be put into one image. Prefer to be a list of numbers.
   o   PresentationDirectionNumberUp : the direction of NumberUp pages are ordered in one image.
   o   OrientationRequested : the orientation of how the image will appear in the document. (portrait or landscape).
   o   PageRanges: Boolean now, defer to processing instructions.
   o   AutoExposure :Boolean
   o   Contrast : predefined range of values – doesnot make sense to have a range of pages.
   o   OriginalType : this is document type (scan original document as text,…)
   o   OutputDocumentSize : describe the size of the output image.
   o   Resolution : sequence of resolution in X, Y direction. Coordinate on the page Should use the same terminology (X, Y) across the semantic model. Not to use CrossFeedDirection, FeedDirection. In projector, short and long are used. In print, Y >= X always.
   o   ScanRegion : the bound of region can be scanned, is the input document size. (lower bound/ upper bound range)
   o   Sharpness : this is a term in image processing.
   o   Sides

Scan Job
o   Scan Job Status
   o   Date Time should be used for all the time instead of tick time.
   o   Job ID – an integer unique within the service, not unique within another service collocated.
   o   JobURI – This element need to remain and it must be a global unique identifier for a job. CUPS implementation of IPP in MAC OS uses this to hold a UUID which can be in actual IETF registered URN format for a Global unique identifier of a job. In MAC this gives a more meaningful way of job accounting from job history.

o   ScanJobTicket
   o   ScanJobDescription :
      o   ElementsCharset and ElementsNaturalLanguage : Both are character set in language for system generated character strings. For example, the response of a request to Scan Service can be in the language you specified here.
      o   Fidelity: is marked in favor of the MustHonored attribute in the individual job processing element. This should be deprecated.

- o JobMandatoryElements – This is a list of ticket elements the Scan Service must be honored in order to create a job. It is very useful for a Scan Client to display the must honored element list to a human operator instead of traversing through each element in Scan Job Ticket. This is used in certain protocol mapping, should not be deprecated. This is a request based on JDF to IPP recently.

- o ScanJobProcessing
  Each element has a MustHonored attribute when the value is true, the Scan Service does not process the job unless the element is supported, otherwise the Scan Service processes the job with its best effort.
    - o JobHoldUntil: keywrd from DPA- MustHort tru –job process if can be hnrd
    - o JobHoldUntilTime : allows you to hold a job until a specific time for processing.
    - o JobPriority – this is a range of 1 or 0 to100 from DPA. Change of job priority in Scan Job will require somebody to switch input originals in and out based on priority. This does not seem applicable to Scan Job. Will move this element to Scan Service specific job processing instructions.
    - o JobRecepientName :
    - o BatchMode : Set by the creator of the job to tell the Scan Service this job consists of batch scan inputs originals that contain a hardcopy instruction sheet in font of each input document.
    - o DocumentOutputMode – to tell the Scan Service whether this is a Single or Multiple document job.
    - o ScanServiceTimeout (optional) – this is a scan service timeout in ScanServiceDescription that's only settable by administrators, not by users or operators. When the user did not press the "stop scan" button after multiple set of originals are scanned, this timeout tells the Scan Service to close and schedule the job for processing.
    - o MultipleSetOriginal (optional) – to tell the Scan Service starts to look for multiple sets of input originals from the platen or ADF.

- o Scan Document Processing
  This overwrites Job processing instruction at document level.
    - o MultipleScanRegion – Multiple scan regions on a single page or separate pages, coming from different input source or scanner. The DocumentOutputMode can be used to specify whether each region is a separate document or all in a single image. This is a sequence of document type and the order of the sequence implies the order of scanning. This element is added because the scan region can change as the function of document type and scan resolution required. Currently the model assumes there is separate a scanner subunit for each type of original. To scan different type of document originals, you can specify different input source of the scanner, e.g. platen, ADF, film reader/scanner. This element can add multiple source of inputs to a single page or multiple pages.
    - o CurrentPageOrder : the logical page order.
    - o OrientationRequested: for output img
    - o PageRanges ?– print only pg 3, 7 , 9 for Print Service, does not make sense for only scan pages 3, 7, 9. REMOVE.
    - o AutoExposure: Boolean

- Should Take sharpness/contract/brightness(-100 to +100), Move them to AutoExposure (becomes complex choice to be specified when AutoExposure is off).
- CompressionFactor : compression factor of a particular type of compression.
- InputType : platen, automatic feeder, film reader.
- Magnification – allow to zoom in and zoom out the scanned document. Should change to Scaling. The Scan Service produces the final file ready to be stored in repository. The Unit is % , the value of 100 is no change (no scaling). Value of less than 100 is size reduction, over 100 is magnification. We need to decide the lowest allowed value.
- OutputDocumentColor – This is color space enum value RGB, CMYK, no ICC profile for color correction, can't do much for scanning. Monochrome(binary- 1bit) is not grey scale.
- Resolution : change CrossFeed & Feed to the common keywords used across the model.
- Rotation: 0, 90, 180, 270 degrees
- SaveDocumentFormat – This is document forma for document to be saved in repository, an enumeration of MIME type which is DocumentFormat. This is keywords of MIME type, e.g. Appl/…
  - **Change to pwg:DocumentFormat for the format of document is for print. A descriptive element.**
- ScanDestination –
  - Location to store the document. Choice of Destination/DestinationDirectory
- ScanRegion – hight, width.
- Sides: doc feeder whether to flip page for duplex or single sided
- SinglePageFile : This can simply be determined by document format. – **remove.** It should be controlled at job level not document level.

- ScanJobReceipt – this has everything in Scan Job Ticket, and its schema will be restructured exactly the same as ScanJobTicket.

Configuration of the Scan Service –

This is a new top-level element of the Scan Service added today. It contains subunit elements for a MFD. Below are subunits used by Scan Service:
- Consoles
- Covers
- Finishers - REMOVE
- InputChannels: analogous to the job channel in printer, the protocol channels by which a Scan Job can be submitted.
- InputTray: Platen, ADF
- Add a ScannerType or ScanTechnology : platen or ADF.
- Media path: (two) platen and ADF, a function of input tray.
- We Need to extend the subunit to accommodate all physical characteristics of a scanner: light bulb, etc…

ScanDocumentStatus –
- CurrentPageOrder: the scanning order from top or bottom of the stack of originals in the input tray, only make sense with ADF.

Move the following three elements to Print Service, they are not applicable to Scan Service:
- DocumentFormatDetailsDetected : REMOVE
- DocumentFormatDetected : REMOVE
- DocumentFormatVersionDetected : REMOVE
- DocumentNumber : The unique identifier for a document within a job.
- ErrorCounts : number of errors encountered while processing the job.
- JobURI : The global unique identifier of a job.
- ServiceURI : The link to the service of the job and documents, in an accounting log it let a job point back to the service, make sense to have at job and service level but not document level. REMOVE.
- SystemUptime: how long the system is up, should be at the service level only. REMOVE from document and job level.
- ScanServiceCounters: This is changed to ScanDocumentCounters using the ScanServiceCounters type.

ScanDocumentProcessing -

This is scan job ticket overwritten at document level. Will be changed to ScanDocmentTicket.
- ScanDocumentDescription
  - DigitalSignature – the digital signature type of the output document, just the types – a list of keywords.
  - DocURI – the link to where the scan data is actually stored.
  - LastDocument : the flag for the last doc for a sequence of documents of a job. For positive confirmation when a job is closed.
  - PageOrderRecieved – need to know that from ADF, but not anything else.

ScanDocumentReceipt will be changed to the same structure as ScanJobTicket.

ActiveJobs – jobs pending or processing

5. WSDL – Web Services Definitions
(Note: highlighted in red are changes to the existing operations or action items)

Scan Service Operations -
- CreateScanJob – create a scan job
- There is no AddDocument operation to a job – User can add document via interaction with the scanner now only.
- CloseScanJob -
- ValidateTicket – both in Scan Service and Template Service. The Scan Service does complete validation of whether the ticket will be honored. The Scan Template Service can only check well-formness of the template.
- CancelJob
- HoldJob – prevent a job from being processed.
- ReleaseJob – allow a job which is on hold, submitted by remote client (by password) to be scheduled.
- There is no PauseJob – which allows a current processing job to be interrupted. The Pause Scan Service operation will stop the current scan job processing and allow other type of jobs to be

processed. Then the Resume operation will resume the current processing job that was paused. In order to resume other new scan job not in the job queue, every existing job in Scan Service need to be individually paused to allow interruption of the new job. The group's consensus is to CancelJob then RestartJob to simplify user's scan operation.

- o PromoteJob increases the priority of a job. This should be removed, the priority element of scan job has been removed. An enterprise user should submit a job and put the job on-hold until he walks up to the scanner to release the job then starts scanning/ processing the job.
- o Restart Job – restart and reprocess a completed job with same processing instructions and a new job id.
- o ReprocessJob – restart and reprocess a completed job with same processing instructions without a new job id, with the previous job that was stored.
- o Both RestartJob and ReprocessJob should be removed. Unless they do not require rescan originals (the Scan Service keeps all previously scanned data…), these two operations do not help much – REMOVE. A user can always resubmit a job with the same convenient template.
- o GetJobs – list the jobs available in the service – will change to GetJobHistory & GetActvJobs to get jobs from different job queues.
- o GetJobElements – get all information for the job.
- o GetScannerElement – will be changed to GetScanServiceElement to get all element in scan service.
- o Pause: to pause the Scan Service.
- o PauseAfterCurrentJob: prefer to add a parameter to Pause for this operation (REMOVE).
- o Resume : to resume Scan service.
- o HoldJob -
- o HoldNewJobs – this can useful when requiring no scan before certain time – any job submitted before that will need explicitly be released.
- o ReleaseJob
- o ReleaseNewJobs -
- o Disable – disable Scan Service = no new job is accepted
- o Enable – enable Scan Service, accepting new job again
- o Remove Activate/Deactivate: these are the same as startup and shutdown
- o Restart – the scan service goes through initialization again, REMOVE
- o Shutdown – shout down the scan service
- o Startup – startup the scan service

Template Manager Service Operations -
- o DeleteTemplate – delete a template
- o CreateTemplate - create a template
- o GetTemplate - get a template
- o PutTemplate - store a template with optional overwrite to replace template
- o REMOVE ReplaceTemplate
- o ValidateTemplate - xml syntax validation only
- o ListTemplate – get a list of templates each has a name
- o SetTemplateExperation – allow a template to be held at a location for a set period of time. (REMOVE), no element in template covers the expiration time.
- o SetTemplateAccessRights – set template permissions.

Scan Service Operation Messages -

1.   o CreateScanJobRequest is a ScanJobTicket.
2.   o CreateScanJobResponse is JobId and UnsupportedElements which is the list of processing
3.   instructions with best efforts that are not going to be honored.
4.   o CloseJobRequest is JobID.
5.   CloseJobResponse : nothing needs to be returned
6.   o Current schema allows us to add document instructions to process a document. But currently there is
7.   no protocol that can pass document instructions along with a CreateScanJobRequest which can only
8.   pass job level processing instructions. Do we need to explicitly model the creation of a document ?
9.   The only way we can do right now is to use a hardcopy instruction sheet in Batch Scan Mode. We
10.   need a way that remotely adds document level processing instructions for each document. At
11.   ScanJobCreation, the ScanJobTicket only have the default document processing instruction for the
12.   entire job. We can create a job with HoldUntil instruction that will create document sequence
13.   structure, then use AddDocument to add document instruction for each document. Similarly, the
14.   local UI of MFD needs to have AddDocument interface. Therefore the following two operations
15.   need to be added:
16.   o AddDocumentRequest – is JobId and DocumentTicket
17.   AddDocumentResponse - is a DocumentNumber and UnsupportedElements.
18.   o ScanJobRequest – REMOVE, a residual from previous model.
19.
20.   o ValidateTicketRequest is a ScanJobticket
21.   ValidateTicketResponse is the UnsupportedElements.
22.   o CanclJobRequest is a JobId
23.   CanclJobRepsonse is a success or a fault with fault description.
24.
25.   o REMOVE the followings : PromoteJobRequest, PromoteJobResponse, RestartJobRequest,
26.   RestartJobResponse, ReprocessJobRequest, ReprocessJobResponse.
27.
28.   o GetScanServiceElementsRequest is a list of RequestedElements which is a list of names. No
29.   restriction of what the names should be. In IPP there are predefined keywords to use.
30.
31.   o GetScanServiceElementResponse is the choice of entire Scan Service top level elements, i.e.
32.   ScanServiceStatus, ScanServiceDescription, DefaultScanTicket, Configuration,
33.   ScanServiceCapabilities . The names and types of the direct children of the Scan Service. The
34.   NMTOKEN of the 5 top level elements. ActiveJobs and JobHistory are not returned.
35.   ALL keyword will retrieve the entire ScanService.
36.
37.   o GetJobRequest is a list of RequestedElements names that will have the enumeration of keywords of
38.   the top level elements of the Scan Job. ALL will get all the Scan Job elements.
39.   GetJobResponse is preferably a summary list of top level elements (e.g user friendly name) of a job
40.   and JobId. For security, one should have a server policy implemented to restrict user only retrieve
41.   his own job.
42.
43.   o GetJobElementsRequest is a JobId and a list of element interested.
44.   GetJobElementsResponse is the top level elements of Scan Job complex type, status, ticket, receipt.
45.   This will not return ScanDocument.
46.

- o The following optional document level operations need to be added: GetDocumentElementsRequest and GetDocumentElementsResponse at document level.
- o PauseRequest currently has no parameter.

- o Remove PauseAferCurrentJobRequest and Response

- o Resume has no parameter.
- o Disable/Enable has no parameter.

- o Remove Deactivate/Activate/Restart request/response.

- o Shutdown/Startup has no parameter.

- o HoldJobRequest is a JobId
  HoldJobResponse is nothing and will have details of operation if you told a job to hold after it has begun processing – e.g. a fault message.
- o RelaseJobRequest is a JobID.
  HoldJobResponse is nothing.

- o Remove HoldNewJobs/ReleaseNewJobs Request/Response

Scan Template Manager Operation Messages -
- o DeleteTemplateRequest is a TemplateId or TemplateName.
  DeleteTemplateResponse is nothing.

- o The ScanJobTemplate schema needs to have TemplateId along with TemplateName.

- o The following Template Manager Service requests/responses need to be flushed out:
  GetTemplateRequest is a TemplateId or TemplateName.
  GetTemplateResponse needs to return an instance of Template.
  ListTemplateRequest/Response
  PutTemplateRequest /Response
  ValidateTemplateRequest/Response

- o REMOVE:
  ReplaceTemplateRequest/Response, SetTemplateExpirationRequest/Response

6. Next Steps
- o Working draft needs to be updated
- o Members please provide comments in email list.
- o Comments will be discussed in the next conference.

7. Next teleconference is January 10, Thursday, 2008 at 3pm EDT