DBSENTRY
Corp.

Samhita

PeopleSoft Integration Guide

Samhita
Document Manager

Samhita PeopleSoft Integration Guide

Contributors: DBSentry Corp. Product Development

This documentation is subject to change without notice, and DBsentry does not warrant that the material contained in this documentation is free of errors. Any errors found in this documentation should be reported to DBsentry in writing.

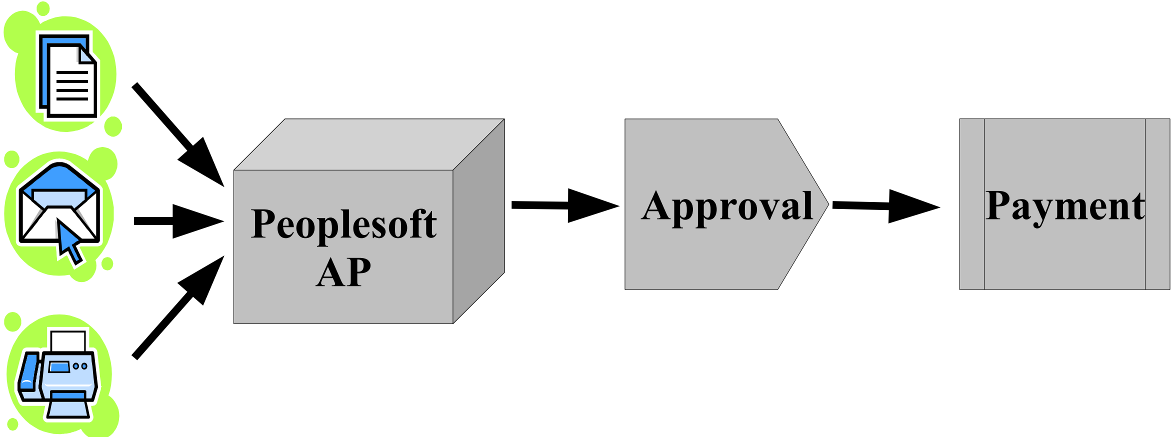
This documentation that accompanies the software is licensed for use only in strict accordance with the applicable license agreement, which should be read carefully, as it governs the terms of use of this documentation and the related software, including the disclosure thereof.

Table of Contents

0.0 Introduction:

Samhita, a comprehensive and user-friendly framework that can quickly and cost-effectively leverage your technology investments enhancing your existing business process by storing documents, vouchers and invoices, linking them to the appropriate records. Making information accessible with a single click from your existing software application enhancing work processes in areas such as accounts payable, procurement, customer service, compliance and many others.

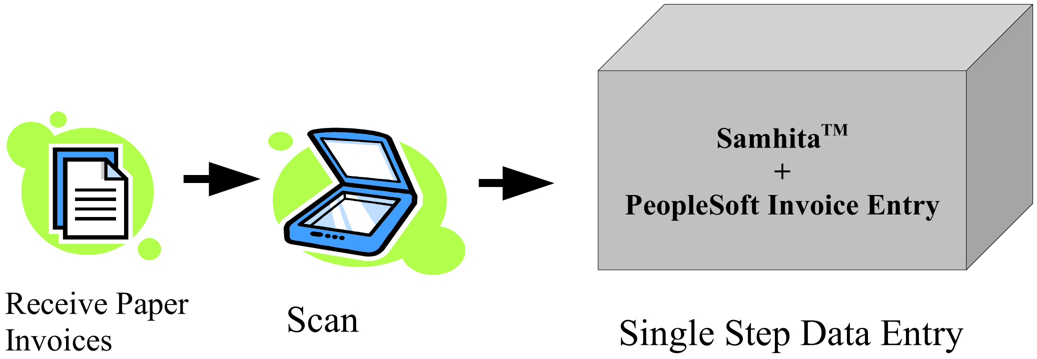
Samhita comes with various adapters that make integration of samhita with other application easier. One of such adapter is Peoplesoft Financial adapter. This adapter can be used to automate some of the key business processes in the Peoplsoft Financials. One of such process is Accounts Payable invoice processing. **Samhita** allows you to streamline invoice processing by removing common paper-related bottlenecks. Samhita lets staff bind receipts and invoices electronically to their respective transactions in your accounts payable software and access those documents instantly from a central repository. A strong workflow ensures routing of items through the approval cycle more efficiently.



This streamlined invoice processing is achieved by following few steps. The first step is to import and capture invoice data. Paper invoices are in multiple forms and need to be transformed into images via scanning and then submitted to rigorous image preprocessing, or cleanup, to ensure that data is easily readable and accurate as it moves through the system.

The invoice captured in the first step, then is store in Samhita system in a logical folder structure. At this point a link is generated to this stored invoice, which would allow one to access the invoice.

Link generated in the second step is incorporated in the Accounts Payable(AP) system. This would enable a filing clerk or an auditor to access the physical invoice by clicking on the link.

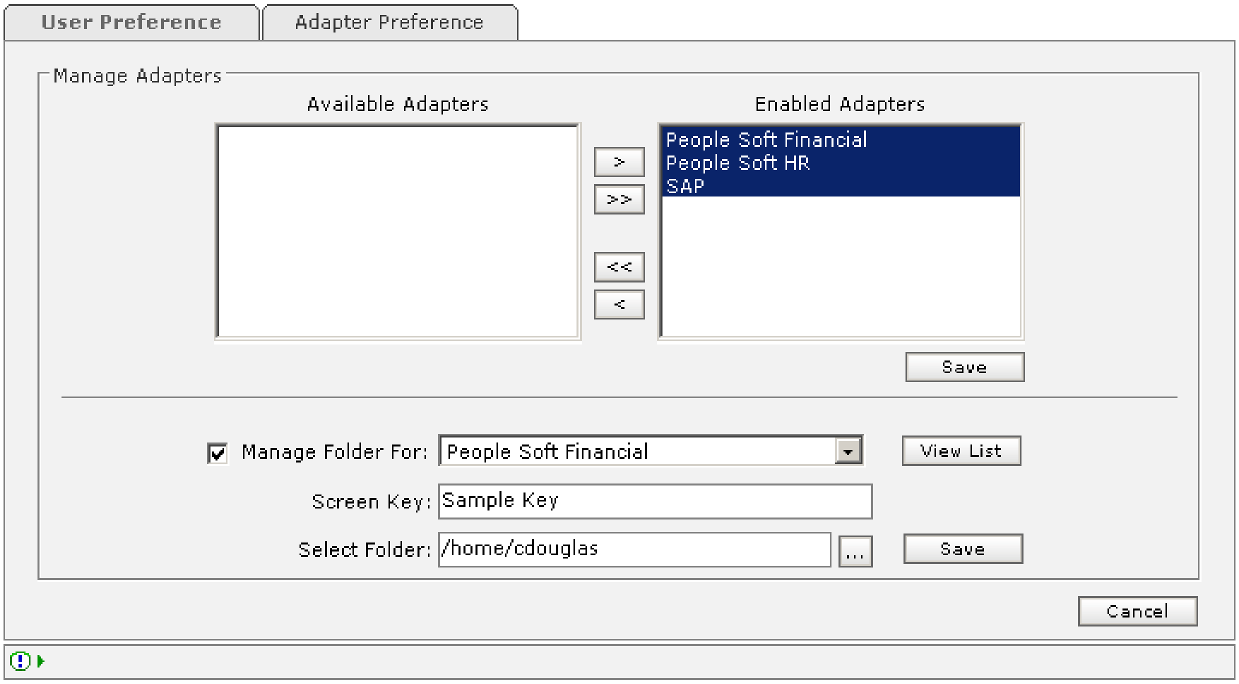


Integration of samhita and Peoplesoft can be achieved using following steps:

1) Enable PeopleSoft Financial Adapter in Samhita
2) Set folders for PeopleSoft Financial Adapter in Samhita
3) Enable/configure Integration Broker in PeopleSoft
4) Create PeopleSoft Objects
5) Sample PeopleCode
6) Samhita PeopleSoft Adapter Message format

1.0 Enable PeopleSoft Financial Adapter:

- To perform this operation you need to have systems administrator privileges in Samhita
- Login to Samhita
- Click on Preference
- Select Adapter Preference



Manage Adapters

Available Adapters

Enabled Adapters
People Soft Financial
People Soft HR
SAP

☑ Manage Folder For: People Soft Financial

Screen Key: Sample Key

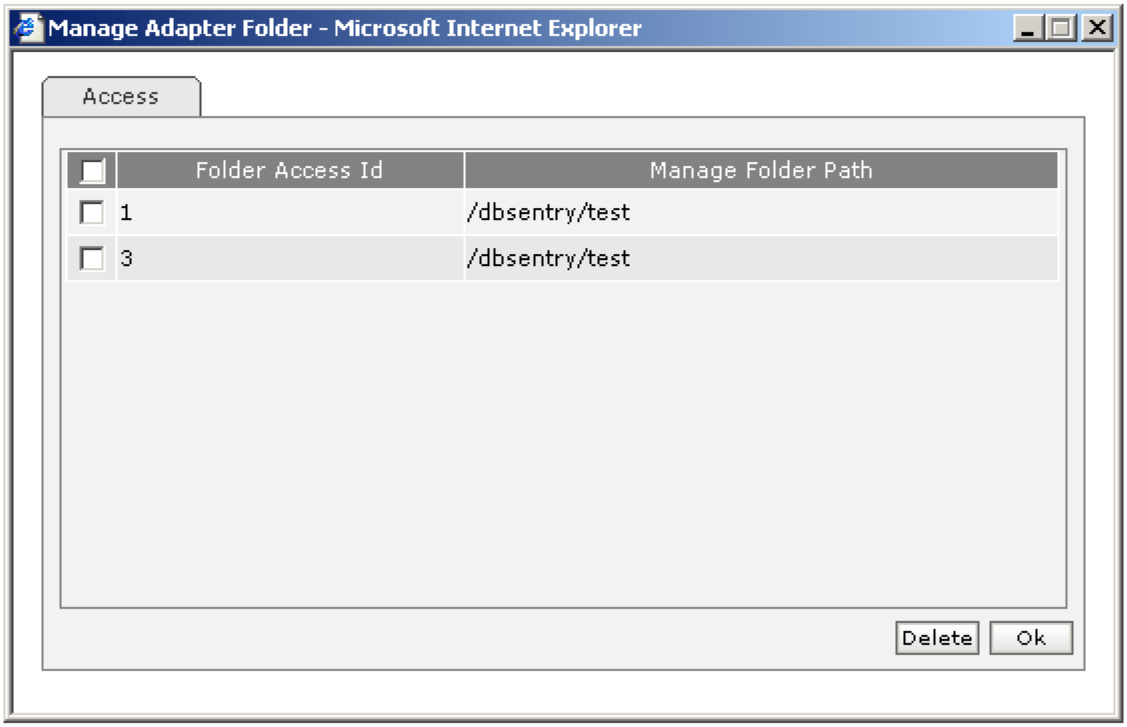
Select Folder: /home/cdouglas

- Select "PeopleSoft Financial" adapter in the "Available Adapters" and move it to "Enabled Adapters".
- Click on "Save"

2.0 Set folders for PeopleSoft Financial Adapter:

To create a new screen key for an enabled adapter:

- Select the adapter from the dropdown next to "Manage Folder For" as shown in fig 1
- Select "Manage Folder For" option as shown in fig 1



□	Folder Access Id	Manage Folder Path
□	1	/dbsentry/test
□	3	/dbsentry/test

- Enter a new and unique Screen Key
- Select the desired folder to which this key should correspond using the Look In... button
- Once the screen key and folder path have been entered, Click "Save" adjacent to the Look In... button in fig 1 to create and save the key

To view the list of screen key(s) for an enabled adapter:

- Select the adapter from the dropdown next to "Manage Folder For"
- Click "View List" to view the list of screen keys and the corresponding folder paths

To delete screen key(s) for an enabled adapter:

- Select the adapter from the dropdown next to "Manage Folder For"
- Click "View List" to view the list of screen keys and the corresponding folder paths as shown in illustration.

- Select the key(s) to be deleted and press "Delete" to delete the same
  - Please note that deleting key(s) does not delete the corresponding folder(s)

3.0 Configure PeopleSoft Integration Broker:

1) Activate Pub/Sub server
   - PeopleTools >> IB >> Monitor >> Monitor Message >> Domain Status
   - Set status of domain to active.
2) Set IB gateway URL for the default local gateway
   - PeopleTools >> IB >> Gateways
   - Set URL to "http://server:port/PSIGW/PeopleSoftListeningConnector" (for e.g. "http://psfin.dbsentry.com:8811/PSIGW/PeopleSoftListeningConnector"
3) Click on "Load" button to load and register installed target connecters if they are not already present.
4) Configure integrationGateway.properties file
   - Click on "properties", userid=administrator, default password=password
   - Set JOLT section (Use the values for JOLT from psappsrv.cfg file). Also, make sure that the password is encrypted.
   - For e.g
   - ig.isc.serverURL=//psfin.dbsentry.com:9020
   - ig.isc.userid=VP1
   - ig.isc.password=bPfdjXLse0Q=
   - ig.isc.toolsRel=8.44
5) Configure nodes.
   - PeopleTools >> IB >> Node Definitions
   - Create a new node "Samhita"
   - Set auth to "None"
6) Create Messages using APP. Designer
   - DBS_VOUCHER_REQ1
   - DBS_VOUCHER_RESP1
7) Create Message Channel using App. Designer.
   - DBS_VOUCHER
8) Associate DBS_VOUCHER_REQ1 and DBS_VOUCHER_RESP1 to DBS_VOUCHER message channel.
9) Create a transaction for node "Samhita"
   - OutSync with DBS_VOUCHER_REQ1 as request and DBS_VOUCHER_RESP1 as response.
   - Set routing type to explicit.
   - Activate the transaction.

4.0 Create PeopleSoft Objects

1) Create following fields

1.	DBS_REQ_VOUCHER	(Char 1)
2.	DBS_MNG_VOUCHER	(Char 1)
3.	DBS_SAMHITA_LINK	(Long Char 800)
4.	DBS_SAMHITA_DOC_ID	(Char 20)
5.	DBS_PARAM_KEY	(Char 40)
6.	DBS_PARAM_VALUE	(Char 100)

2) Modify record “VOUCHER” and add fields defined in step (1).
3) Add FieldChange peoplecode for DBS_REQ_VOUCHER and DBS_MNG_VOUCHER.
4) Create a record DBS_PARAM with DBS_PARAM_KEY and DBS_PARAM_VALUE as record fileds.
5) Add SavePostChange peoplecode to DBS_SAMHITA_LINK filed
6) Add three buttons in page VCHR_HDR_SBP
    1. “Get Next Image” button: =Associate with peoplecode on VOUCHER.DBS_REQ_VOUCHER.
    2. “Manage Image” button :Associate with peoplecode on VOUCHER.DBS_MNG_VOUCHER.
    3. “Show Image” button : Associate with file VOUCHER.DBS_SAMHITA_LINK. Set the property for link to open in new window.
7) Add “Show Image” button in VCHR_HDR_SBP4 as in step (6.3)
8) Add “Show Image” button in VCHR_HDR_SBP4 as in step (6.3)

5.0 Sample Peoplecode:

DBS_REQ_VOUCHER:FieldChange

```
/******************************************************************************
*                        Confidentiality Information                          *
*                                                                             *
* This module is the confidential and proprietary information of             *
* DBSentry Corp.; it is not to be copied, reproduced, or transmitted in any *
* form, by any means, in whole or in part, nor is it to be used for any     *
* purpose other than that for which it is expressly provided without the    *
* written permission of DBSentry Corp.                                        *
*                                                                             *
* Copyright (c) 2004-2005 DBSentry Corp.  All Rights Reserved.                *
*                                                                             *
*******************************************************************************
* Author: Manish Gupta                                                        *
******************************************************************************/
Local XmlDoc &xmlRequestDoc;
Local XmlDoc &xmlResponseDoc;

Local string &responseAction;
Local string &responseURL;
Local string &responseDocumentId;

Local string &xmlString = "<?xml version='1.0'?><root/>";
Local string &reqUserId = “”;
Local string &reqPasswd = “”;

Local string &reqDataType1 = “ImageDocumentId”;
Local string &reqDataType2 = “ImageURL”;
Local string &reqDataValue1 = “”;
Local string &reqDataValue2 = “”;

Local Rowset &paramrowset;
Local Row    &paramrow;
Local Record &paramrecord;

/* Get parameters */

&paramrowset = CreateRowset(RECORD.DBS_PARAM);
&paramrowset.Fill(“where dbs_param_key = :1”, “USERID”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqUserId = &paramrecord.GetField(FIELD.DBS_PARAM_KEY).Value;

&paramrowset.Fill(“where dbs_param_key = :1”, “PASSWD”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqPasswd = &paramrecord.GetField(FIELD.DBS_PARAM_VALUE).Value;

/* Create XML Request for the next Voucher */
&xmlRequestDoc = CreateXmlDoc(&xmlstring);
&xmlRequestDoc.DocumentElement.AddAttribute("Request", "SendNextImage");
&xmlRequestDoc.DocumentElement.AddAttribute(“UserId”, &requserId);
&xmlRequestDoc.DocumentElement.AddAttribute(“Passwd”, &reqPasswd);
```

```
&xmlRequestDoc.DocumentElement.AddAttribute(&reqDataType1, &reqDataValue1);
&xmlRequestDoc.DocumentElement.AddAttribute(&reqDataType1, &reqDataValue1);
&xmlRequestDoc.DocumentElement.AddAttribute(&reqDataType2, &reqDataValue2);

/* Send Request */
&xmlResponseDoc = SyncRequestXmlDoc(&xmlRequestDoc, Message.DBS_VOUCHER_REQ1,
Node.SAMHITA);

/* Receive Response */
&responseAction = &xmlResponseDoc.DocumentElement.GetAttributeValue(1);
&responseURL = &xmlResponseDoc.DocumentElement.GetAttributeValue(2);
&responseDocumentId = &xmlResponseDoc.DocumentElement.GetAttributeValue(3);

If &responseAction = “NextImage” Then
    VOUCHER.DBS_SAMHITA_LINK = &responseURL;
    VOUCHER.DBS_SAMHITA_DOC_ID = &responseDocumentId;
    GetField().Enabled = False;
    ViewContentURL(&responseURL);
Else
    If &responseAction = “QueueEmpty” Then
        MessageBox(0,””,0,0, “Voucher Queue empty, please inform your
supervisor about it.”);
    End-if;
    End-if;
```

DBS_MNG_VOUCHER:FieldChange

```
/****************************************************************************
*                       Confidentiality Information                         *
*                                                                           *
* This module is the confidential and proprietary information of            *
* DBSentry Corp.; it is not to be copied, reproduced, or transmitted in any *
* form, by any means, in whole or in part, nor is it to be used for any     *
* purpose other than that for which it is expressly provided without the    *
* written permission of DBSentry Corp.                                      *
*                                                                           *
* Copyright (c) 2004-2005 DBSentry Corp.  All Rights Reserved.              *
*                                                                           *
*****************************************************************************
* Author: Manish Gupta                                                      *
****************************************************************************/
Local string &reqUserId = “”;
Local string &reqPasswd = “”;
Local string &mgmtURL = “”;
Local string &samhita_doc_id;

/* Get parameters */

&paramrowset = CreateRowset(RECORD.DBS_PARAM);
&paramrowset.Fill(“where dbs_param_key = :1”, “USERID”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqUserId = &paramrecord.GetField(FIELD.DBS_PARAM_VALUE).Value;

&paramrowset.Fill(“where dbs_param_key = :1”, “PASSWD”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqPasswd = &paramrecord.GetField(FIELD.DBS_PARAM_VALUE).Value;

&paramrowset.Fill(“where dbs_param_key = :1”, “MGMTURL”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&mgmtURL = &paramrecord.GetField(FIELD.DBS_PARAM_VALUE).Value;

&samhita_doc_id=VOUCHER.DBS_SAMHITA_DOC_ID;

/* Create URL */
```

```
&mgmtURL = &mgmtURL | “?USERID=” | &reqUserId | “&PASSWD=” | &reqPasswd |
“&DOCUMENTID=” | &samhita_dic_id;

/* Send Request */
ViewContentURL(&mgmtURL);
```

DBS_SAMHITA_LINK:SavePostChange

```
/****************************************************************************
*                       Confidentiality Information                        *
*                                                                           *
* This module is the confidential and proprietary information of            *
* DBSentry Corp.; it is not to be copied, reproduced, or transmitted in any *
* form, by any means, in whole or in part, nor is it to be used for any     *
* purpose other than that for which it is expressly provided without the    *
* written permission of DBSentry Corp.                                      *
*                                                                           *
* Copyright (c) 2004-2005 DBSentry Corp.  All Rights Reserved.              *
*                                                                           *
*****************************************************************************
* Author: Manish Gupta                                                      *
****************************************************************************/
Local XmlDoc &xmlRequestDoc;
Local XmlDoc &xmlResponseDoc;

Local string &responseAction;
Local string &responseURL;
Local string &responseDocumentId;

Local string &xmlString = "<?xml version='1.0'?><root/>";
Local string &reqUserId = “”;
Local string &reqPasswd = “”;

Local string &reqDataType1 = “Voucher_No”;
Local string &reqDataValue1 = VOUCHER.VOUCHER_ID;

/* Get parameters */

&paramrowset = CreateRowset(RECORD.DBS_PARAM);
&paramrowset.Fill(“where dbs_param_key = :1”, “USERID”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqUserId = &paramrecord.GetField(FIELD.DBS_PARAM_KEY).Value;

&paramrowset.Fill(“where dbs_param_key = :1”, “PASSWD”);
&paramrow = &paramrowset.GetRow(1);
&paramrecord = &row.GetRecord(RECORD.DBS_PARAM);
&reqPasswd = &paramrecord.GetField(FIELD.DBS_PARAM_VALUE).Value;


/* Create XML Request for the next Voucher */
&xmlRequestDoc = CreateXmlDoc(&xmlstring);
&xmlRequestDoc.DocumentElement.AddAttribute("Request", "ImageProcessed");
&xmlRequestDoc.DocumentElement.AddAttribute(“UserId”, &requserId);
&xmlRequestDoc.DocumentElement.AddAttribute(“Passwd”, &reqPasswd);
&xmlRequestDoc.DocumentElement.AddAttribute(&reqDataType1, &reqDataValue1);

/* Send Request */
&xmlResponseDoc = SyncRequestXmlDoc(&xmlRequestDoc, Message.DBS_VOUCHER_REQ1,
Node.SAMHITA);

/* Receive Response */
&responseAction = &xmlResponseDoc.DocumentElement.GetAttributeValue(1);

If &responseAction <> “ImageProcessedSuccess” Then
    MessageBox(0,””,0,0, “New Voucher upload failed. Please try again.”);
End-if;
```

6.0 Samhita PeopleSoft Adapter Message Format:

Messages between PeopleSoft Financials and Samhita's PeopleSoft Adapter are in XML format. The interaction starts with the request from external application and ends with response from Samhita Adapter. The external application (in this case PeopleSoft Financials) sends request using plain HTTP POST messages. The XML messages, typically have the following format:

```
<?xml version="1.0"?>
<root Request="GetNextImage" UserId=”XXX” Password=”XXX”
AclName=”PSF_NORTH” ImageDocumentID=”” ImageURL=""/>
```

```
<?xml version="1.0"?>
<root Request="NextImage" ImageDocumentID=”12345”
ImageURL="http://www.google.com"/>
```

The following table gives all the request and response combination developed for PeopleSoft Financials Adapter:

PeopleSoft (Request)	***Samhita Gateway Servlet (Response)***

Request=GetNextImage
AclName=<Value>
UserId=<Value>
Password=<value>
ImageDocumentId=””
ImageURL=””
	Response=NextImage
	ImageDocumentId=<Value>
	ImageURL=<Value>
	OR
	Response=QueueEmpty
	ImageDocumentID=””
	ImageURL=””

Request=ImageProcessed
AclName=<Value>
UserId=<Value>
Password=<Value>
ImageDocumentId=””
VoucherId=<Value>
	Response=ImageProcessedSuccess

Request=DeleteImage
AclName=<Value>
UserId=<Value>
Password=<value>
ImageDocumentID=<Value>
	Response=DeleteImageSuccess

Request=GetManagedImage
UserId=<Value>
Password=<Value>
ImageDocumentId=””
VoucherId=<Value>
Key=<Value>
	Response=ManagedImage
	ImageDocumentId=<Value>
	ImageURL=<Value>