Table of Contents

1. Preface................................................................................................................................................. 4
   1.1 Intended Audience .......................................................................................................................... 4
   1.2 Documentation Accessibility ........................................................................................................... 4
   1.3 Access to Oracle Support ............................................................................................................... 4
   1.4 Structure ......................................................................................................................................... 4
   1.5 Related Information Sources .......................................................................................................... 4

2. Content Upload Configuration (Day 1) ............................................................................................... 5
   2.1 Creating folder for holding files ....................................................................................................... 5
   2.2 Creating Global Database Variable ............................................................................................... 6
   2.3 Creating Procedure ......................................................................................................................... 7
   2.4 Verifying DB State Before Procedure Execution .............................................................................. 10
   2.5 Executing Procedure ...................................................................................................................... 11
   2.6 Verifying DB State After Procedure Execution ............................................................................... 15
   2.7 Delete Created Directory, Variable And Procedure ........................................................................ 16
1. Preface

1.1 Intended Audience
This document is intended for the following audience:

- Customers
- Partners

1.2 Documentation Accessibility
For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=accandid=docacc.

1.3 Access to Oracle Support
Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=accandid=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=accandid=trs if you are hearing impaired.

1.4 Structure
This manual is organized into the following categories:
Preface gives information on the intended audience. It also describes the overall structure of the User Manual.
The subsequent chapters describes following details:

- Purpose
- Configuration / Installation.

1.5 Related Information Sources
For more information on Oracle APIs Banking Release 19.1.0.0.0, refer to the following documents:

- Oracle Banking APIs Installation Manuals
2. **Content Upload Configuration (Day 1)**

2.1 **Creating folder for holding files**

1. Create a directory (Example - '/scratch/contents') on server where db is installed. This directory will hold files which are to be uploaded.
2. Copy files in the directory created in above step.

**Note:** We are not providing files out of the box. Please provide files for different documents as per your requirement. Makes sure that proper file names use during procedure execution (Refer to section 2.5.3 column 'Proc Execution Statement' for more detail)
2.2 Creating Global Database Variable

1. Login into database using sys user.
2. Create db variable (CONTENT_FILES) as mentioned below:
   ```sql
   create or replace directory CONTENT_FILES as '/scratch/contents';
   ```
3. Granting access to schema (OBAPISCHEMA – Example- OBAPI_DBAUTH181IT) for variable created in step 2 of 2.2.
   ```sql
   GRANT READ, WRITE ON DIRECTORY CONTENT_FILES TO &&schema_name;
   ```
2.3 Creating Procedure

1. Login into database using schema user (OBAPISCEMA).
2. Compile following procedure.

- For Origination Disclosure Configuration:

```sql
create or replace procedure add_disclosure(
p_file_path in varchar2,
p_file_name in varchar2,
p_mime_type in varchar2,
p_disclosure_code in varchar2,
p_disclosure_name in varchar2,
p_disclosure_desc in varchar2,
p_product_subclass in varchar2)
as
    l_blob blob;
    l_file_content bfile := bfilename(p_file_path, p_file_name);
    l_max_val number;
begin
    select NVL((max(ID)+1),1) into l_max_val from DIGX_CM_CONTENT;
    insert into DIGX_CM_CONTENT
    (ID, DOCUMENT_CHECKLIST_ID, TITLE, MIME_TYPE, CONTENT_CATEGORY, CONTENT_TYPE, CONTENT_SIZE, OBJECT_VERSION_NUMBER, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATE_DATE, OBJECT_STATUS, PARTY_ID, SHARED_FLAG) values
    (l_max_val, '1', p_disclosure_name, p_mime_type, null, 'FILE', 6, null, 'obapi user', sysdate, null, null, null, null, null, null);

    insert into DIGX_OR_SUBMISSION_DISCLOSURES (PRODUCT_CLASS, PRODUCT_SUBCLASS, NAME, CONTENT_ID, DESCRIPTION, REFERENCE_ID, DISCLOSURE_CODE) values ('LOANS', p_product_subclass, p_disclosure_name, l_max_val, p_disclosure_desc, l_max_val, p_disclosure_code);

    update DIGX_CM_CONTENT set FILE_CONTENT = empty_blob() WHERE ID = l_max_val;
    SELECT FILE_CONTENT into l_blob FROM DIGX_CM_CONTENT WHERE ID = l_max_val;
    dbms_lob.open(l_file_content, dbms_lob.lob_readonly);
    dbms_lob.open(l_blob, dbms_lob.lob_readwrite);
    dbms_lob.loadfromfile(dest_lob => l_blob, src_lob => l_file_content, amount => dbms_lob.getlength(l_file_content));

    dbms_lob.close(l_blob);
```
commit;
end;
/

- **For User Print Document Configuration:**

```sql
create or replace procedure add_content(
p_file_path in varchar2,
p_file_name in varchar2,
p_mime_type in varchar2,
p_document_name in varchar2
)
as
    l_blob        blob;
    l_file_content             bfile := bfilename(p_file_path, p_file_name);
    l_max_val number;
begin
    select NVL((max(ID)+1),1) into l_max_val from DIGX_CM_CONTENT;
    insert into DIGX_CM_CONTENT
    (ID,DOCUMENT_CHECKLIST_ID,TITLE,MIME_TYPE,CONTENT_CATEGORY,CONTENT_TYPE,CONTENT_SIZE,OBJECT_VERSION_NUMBER,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATE_DATE,OBJECT_STATUS,PARTY_ID,SHA_RED_FLAG) values
    (l_max_val,'1',p_document_name,p_mime_type,null,'FILE',6,null,'obapiuser',sysdate,null,null,null,null,null);
    insert into DIGX_UM_USERPRINTDOCS(CODE, NAME)values (l_max_val, p_document_name);
    update DIGX_CM_CONTENT set FILE_CONTENT = empty_blob() WHERE ID = l_max_val;
    SELECT FILE_CONTENT into l_blob FROM DIGX_CM_CONTENT WHERE ID = l_max_val;
    dbms_lob.open(l_file_content, dbms_lob.lob_readonly);
    dbms_lob.open(l_blob, dbms_lob.lob_readwrite);
    dbms_lob.loadfromfile(dest_lob      => l_blob,
                         src_lob       => l_file_content,
                         amount        => dbms_lob.getlength(l_file_content));
    dbms_lob.close(l_file_content);
    dbms_lob.close(l_blob);
    commit;
end;
/
```
For Feedback

Pre-requisites before compiling below procedure:

Execute below steps to make Scale(Rating) icons available for Feedback Module.

Note: The images can be found at below location: clip/trunk/core/channel/images/feedback

Insert Query for DIGX_CM_CONTENT Table

Insert into DIGX_CM_CONTENT
(ID, PARTY_ID, DOCUMENT_CHECKLIST_ID, TITLE, MIME_TYPE, CONTENT_CATEGORY, CONTENT_TYPE, FILE_CONTENT, CONTENT_SIZE, SHARED_FLAG, OBJECT_VERSION_NUMBER, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATE_DATE, OBJECT_STATUS) values ('${contentId1}', null, null, 'heart-fill.svg', 'image/svg', null, null, null, '', 'Y', 1, 'superadmin', sysdate, 'superadmin', sysdate, null);

Insert into DIGX_CM_CONTENT
(ID, PARTY_ID, DOCUMENT_CHECKLIST_ID, TITLE, MIME_TYPE, CONTENT_CATEGORY, CONTENT_TYPE, FILE_CONTENT, CONTENT_SIZE, SHARED_FLAG, OBJECT_VERSION_NUMBER, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATE_DATE, OBJECT_STATUS) values ('${contentId2}', null, null, 'star.png', 'image/jpeg', null, null, null, '', 'Y', 1, 'superadmin', sysdate, 'superadmin', sysdate, null);

Insert into DIGX_CM_CONTENT
(ID, PARTY_ID, DOCUMENT_CHECKLIST_ID, TITLE, MIME_TYPE, CONTENT_CATEGORY, CONTENT_TYPE, FILE_CONTENT, CONTENT_SIZE, SHARED_FLAG, OBJECT_VERSION_NUMBER, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATE_DATE, OBJECT_STATUS) values ('${contentId3}', null, null, 'thumbs-up.svg', 'image/jpeg', null, null, null, '', 'Y', 1, 'superadmin', sysdate, 'superadmin', sysdate, null);

"Kindly Refer this folder location for all the three icons to be used in below steps:
[clip/core/channel/images/feedback]"

Update the content in digx_fd_scale table.

update digx_fd_scale set content_id='${contentId1}' where determinant_value in ('*', 'OBDX_BU') and id='1';
update digx_fd_scale set content_id='${contentId2}' where determinant_value in ('*', 'OBDX_BU') and id='2';
update digx_fd_scale set content_id='${contentId3}' where determinant_value in ('*', 'OBDX_BU') and id='3';

Procedure:

create or replace procedure load_file_to_content( p_file_path in varchar2, p_file_name in varchar2, p_content_id in varchar2 )
as
  l_blob blob;
  l_file_content bfile := bfilename(p_file_path, p_file_name);
begin
    update DIGX_CM_CONTENT set FILE_CONTENT = empty_blob() WHERE ID = p_content_id;
    SELECT FILE_CONTENT into l_blob FROM DIGX_CM_CONTENT WHERE ID = p_content_id;
    dbms_lob.open(l_file_content, dbms_lob.lob_readonly);
    dbms_lob.open(l_blob, dbms_lob.lob_readwrite);
    dbms_lob.loadfromfile(dest_lob => l_blob, src_lob => l_file_content, amount => dbms_lob.getlength(l_file_content));
    dbms_lob.close(l_file_content);
    dbms_lob.close(l_blob);
    commit;
end;
/

2.4 **Verifying DB State Before Procedure Execution**

1. Login into database using schema (OBAPISCHEMA) user.
2. Confirm data by executing below queries; before executing day1 scripts.

- **For Origination Disclosure Configuration:**

  ```sql
  select * from DIGX_CM_CONTENT;
  select * from DIGX_OR_SUBMISSION_DISCLOSURES;
  ```

- **For User Print Document Configuration:**

  ```sql
  select * from DIGX_CM_CONTENT;
  select * from DIGX_UM_USERPRINTDOCS;
  ```

- **For Feedback**

  ```sql
  select file_content from DIGX_CM_CONTENT where ID IN ('${contentId1}', '${contentId2}', '${contentId3}');
  ```

  **Note:** All cells should be null or with empty BLOB
2.5 Executing Procedure

1. Login into database using schema user (OBAPISCHEMA).
2. Template for procedure call is as given below

- **For Origination Disclosure Configuration:**
  ```
  begin
  add_disclosure(
  <Variable name for File's registered path>,
  <Files' Name>,
  <CONTENT'S MIME TYPE>,
  <DISCLOSURE CODE>,
  <DISCLOSURE NAME>,
  <DISCLOSURE DESCRIPTION>,
  <PRODUCT SUBCLASS>
  );
  end;
  /
  ```

- **For User Print Document Configuration:**
  ```
  begin
  add_content(
  <Variable name for File's registered path>,
  <Files' Name>,
  <CONTENT'S MIME TYPE>,
  <DOCUMENT NAME>
  );
  end;
  /
  ```

- **For Feedback**
  ```
  begin
  load_file_to_content(
  <Variable name for File's registered path>,
  <Files' Name>,
  <Content ID>
  );
  end;
  /
  ```
3. Execute commands given under column ‘Proc Execution Statement’

- For Origination Disclosure Configuration:

<table>
<thead>
<tr>
<th>SUB-CLASS</th>
<th>CODE</th>
<th>Name</th>
<th>Desc</th>
<th>MIME Type</th>
<th>Proc Execution Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOLOAN</td>
<td>ESIGN</td>
<td>E-SIGN Disclosure</td>
<td>E-SIGN Disclosure</td>
<td>application/pdf</td>
<td>begin add_disclosure('CONTENT_FILES', 'fileName1.pdf', 'application/pdf', 'ESIGNDISCLOSURE', 'E-SIGN Disclosure', 'E-SIGN_Disclosure', 'AUTOLOANFLL'); end; /</td>
</tr>
<tr>
<td>FLL</td>
<td>PRIVACY</td>
<td>Privacy Policy</td>
<td>Privacy Policy</td>
<td>application/pdf</td>
<td>begin add_disclosure('CONTENT_FILES', 'fileName2.pdf', 'application/pdf', 'PRIVACYPOLICY', 'Privacy Policy', 'Privacy_Policy', 'AUTOLOANFLL'); end; /</td>
</tr>
<tr>
<td>FLL</td>
<td>LOANACCOUNT</td>
<td>Loan Account Agreement</td>
<td>Loan_Account_Agree ment</td>
<td>application/pdf</td>
<td>begin add_disclosure('CONTENT_FILES', 'fileName3.pdf', 'application/pdf', 'LOANACCOUNTAGREEMENT', 'Loan Account Agreement', 'Loan_Account_Agreement', 'AUTOLOANFLL'); end; /</td>
</tr>
</tbody>
</table>
### Content Upload Configuration (Day 1)

<table>
<thead>
<tr>
<th>PAY DAY</th>
<th>ESIGNDISCLOSURE</th>
<th>E-SIGN Disclosure</th>
<th>E-SIGN Disclosure</th>
<th>application/pdf</th>
<th>begin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>add_disclosure('CONTENT_FILES', 'fileName4.pdf', 'application/pdf', 'ESIGNDISCLOSURE', 'E-SIGN Disclosure', 'PAYDAY');</td>
<td>end;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAY DAY</th>
<th>PRIVACYPOLICY</th>
<th>Privacy Policy</th>
<th>Privacy Policy</th>
<th>application/pdf</th>
<th>begin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>add_disclosure('CONTENT_FILES', 'fileName5.pdf', 'application/pdf', 'PRIVACYPOLICY', 'Privacy Policy', 'Privacy_Policy', 'PAYDAY');</td>
<td>end;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAY DAY</th>
<th>LOANACCOUNT AGREEMENT</th>
<th>Loan Account Agreement</th>
<th>Loan Account Agreement</th>
<th>application/pdf</th>
<th>begin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>add_disclosure('CONTENT_FILES', 'fileName6.pdf', 'application/pdf', 'LOANACCOUNTAGREEMENT', 'Loan Account Agreement', 'Loan_Account_Agreement', 'PAYDAY');</td>
<td>end;</td>
</tr>
</tbody>
</table>

#### Note:
- fileName1.pdf - fileName6.pdf must be present at the location mentioned in step 1.1.2.

- For User Print Document Configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>MIME Type</th>
<th>Proc Execution Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Promotional Offer</td>
<td>application/pdf</td>
<td>begin add_content('CONTENT_FILES', 'PromotionalOffer.pdf', 'application/pdf', 'Promotional Offer'); end;</td>
</tr>
</tbody>
</table>
### Name | MIME Type | Proc Execution Statement
---|---|---
Terms and Conditions | application/pdf | begin
|  |  | add_content('CONTENT_FILES', 'TnC.pdf', 'application/pdf', 'Terms and Conditions');
|  |  | end;
|  |  | /

Welcome Letter | application/pdf | begin
|  |  | add_content('CONTENT_FILES', 'WelcomeLetter.pdf', 'application/pdf', 'Welcome Letter');
|  |  | end;
|  |  | /

**Note:** - PromotionalOffer.pdf - TnC.pdf- WelcomeLetter.pdf must be present at the location mentioned in step 1.1.2.

- **For Feedback**
  
  **Proc Execution Statement:**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Proc Execution Statement</th>
</tr>
</thead>
</table>
| Heart-fill.svg | begin
|  | load_file_to_content('CONTENT_FILES', 'heart-fill.svg', '${contentId1}');
|  | end;
|  | /

| Star.png       | begin
|  | load_file_to_content('CONTENT_FILES', 'star.png', '${contentId2}');
|  | end;
|  | /

| Thumbs-up.svg  | begin
|  | load_file_to_content('CONTENT_FILES', 'thumbs-up.svg', '${contentId3}');
|  | end;
|  | /
2.6 Verifying DB State After Procedure Execution

1. Login into database using schema user (OBAPISHEMA).
2. Confirm date if the data is properly inserted into DB by executing below queries.

- **For Origination Disclosure Configuration:**

```sql
select * from DIGX_CM_CONTENT;
select * from DIGX_OR_SUBMISSION_DISCLOSURES;
```

- **For User Print Document Configuration:**

```sql
select * from DIGX_CM_CONTENT;
select * from DIGX_UM_USERPRINTDOCS;
```

- **For Feedback**

```sql
select file_content from DIGX_CM_CONTENT where ID IN ('${contentId1}', '${contentId2}', '${contentId3}');
```

**Note:** - All cells should contain BLOB
2.7 **Delete Created Directory, Variable And Procedure**

1. Delete the directory variable, created in step 2 of 1.2 (with sys user).
   
   drop directory CONTENT_FILES;

2. Delete the directory, created in step 1 of 1.1.

3. Drop the procedure, created in step 1.3.

- **For Origination Disclosure Configuration:**
  
  drop procedure add_disclosure;

- **For User Print Document Configuration:**
  
  drop procedure add_content

- **For Feedback**
  
  drop procedure load_file_to_content;