

Content Upload Guide (Day 1)
Oracle Banking APIs
Release 20.1.0.0.0

Part No. F30660-01

May 2020

ORACLE®

Content Upload Guide (Day 1)

May 2020

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2006, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1. Preface	1-1
1.1 Intended Audience.....	1-1
1.2 Documentation Accessibility.....	1-1
1.3 Access to Oracle Support.....	1-1
1.4 Structure	1-1
1.5 Related Information Sources.....	1-1
2. Content Upload Configuration (Day 1)	2-1
2.1 Creating folder for holding files.....	2-1
2.2 Creating Global Database Variable	2-1
2.3 Creating Procedure.....	2-1
2.4 Verifying DB State Before Procedure Execution	2-5
2.5 Executing Procedure	2-6
2.6 Verifying DB State After Procedure Execution	2-11
2.7 Delete Created Directory, Variable And Procedure.....	2-12
3. ATM/Branch Locator.....	3-1
3.1 Maintaining Default Radius.....	3-1

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=acc&id=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=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=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:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking APIs Release 20.1.0.0.0, 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. Make 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:
create or replace directory CONTENT_FILES as '/scratch/contents';
3. Granting access to schema (OBAPISHEMA – Example- OBAPI_DBAUTH181IT) for variable created in step 2 of 2.2.

```
GRANT READ, WRITE ON DIRECTORY CONTENT_FILES TO &&schema_name;
```

2.3 Creating Procedure

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

- **For Origination Disclosure Configuration:**

```
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_FL
AG) values

```

```

(l_max_val,'1',p_disclosure_name,p_mime_type,null,'FILE',6,null,'obdxuser',sysdate,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_file_content);

```

```

dbms_lob.close(l_blob);

```

```

commit;

```

```

end;

```

```

/

```

- **For User Print Document Configuration:**

```

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,SHARED_FL
AG) values

```

```

(l_max_val,'1',p_document_name,p_mime_type,null,'FILE',6,null,'obdxuser',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,CON
TENT_TYPE,FILE_CONTENT,CONTENT_SIZE,SHARED_FLAG,OBJECT_VERSION_NUMBE
R,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATE_DATE,OBJECT_ST
ATUS) 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,CON
TENT_TYPE,FILE_CONTENT,CONTENT_SIZE,SHARED_FLAG,OBJECT_VERSION_NUMBE
R,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATE_DATE,OBJECT_ST
ATUS) 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,CON
TENT_TYPE,FILE_CONTENT,CONTENT_SIZE,SHARED_FLAG,OBJECT_VERSION_NUMBE
R,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATE_DATE,OBJECT_ST
ATUS) 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:**

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

- **For User Print Document Configuration:**

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

- **For Feedback**

```
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 (OBAPISHEMA).
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:**

SUB-CLASS	CODE	Name	Desc	MIME Type	Proc Execution Statement
AUTOLOANFLL	ESIGN DISCLOSURE	E-SIGN Disclosure	E-SIGN Disclosure	application/pdf	begin add_disclosure('CONTENT_FILE S', 'fileName1.pdf' , 'application/pdf', 'ESIGNDISCLOS URE', 'E-SIGN Disclosure', 'E- SIGN_Disclosure' , 'AUTOLOANFLL'); end; /
AUTOLOANFLL	PRIVACY POLICY	Privacy Policy	Privacy Policy	application/pdf	begin add_disclosure('CONTENT_FILE S', 'fileName2.pdf' , 'application/pdf', 'PRIVACYPOLIC Y', 'Privacy Policy', 'Privacy_Policy', 'AUTOLOANFLL'); end; /

SUB-CLASS	CODE	Name	Desc	MIME Type	Proc Execution Statement
AUTOLOANFLL	LOANACCOU NTAGREEME NT	Loan Account Agreement	Loan_ Account_A greement	applicatio n/pdf	begin add_disclosure('CONTENT_FILE S', 'fileName3.pdf' , 'application/pdf', 'LOANACCOUNT AGREEMENT', 'Loan Account Agreement', 'Loan_Account_A greement', 'AUTOLOANFLL'); end; /
PAY DAY	ESIGNDISCLO SURE	E-SIGN Disclosure	E-SIGN_ Disclosure	applicatio n/pdf	begin add_disclosure('CONTENT_FILE S', 'fileName4.pdf' , 'application/pdf', 'ESIGNDISCLOS URE', 'E-SIGN Disclosure', 'E- SIGN_Disclosure' , 'PAYDAY'); end; /

SUB-CLASS	CODE	Name	Desc	MIME Type	Proc Execution Statement
PAY DAY	PRIVACYPOLICY	Privacy Policy	Privacy_Policy	application/pdf	begin add_disclosure('CONTENT_FILES', 'fileName5.pdf' , 'application/pdf', 'PRIVACYPOLICY', 'Privacy Policy', 'Privacy_Policy', 'PAYDAY'); end; /
PAY DAY	LOANACCOUNT AGREEMENT	Loan Account Agreement	Loan_Account_Agreement	application/pdf	begin add_disclosure('CONTENT_FILES', 'fileName6.pdf' , 'application/pdf', 'LOANACCOUNT AGREEMENT', 'Loan Account Agreement', 'Loan_Account_Agreement', 'PAYDAY'); end; /

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

- **For User Print Document Configuration:**

Name	MIME Type	Proc Execution Statement
Promotional Offer	application/pdf	<pre>begin add_content('CONTENT_FILES', 'PromotionalOffer.pdf', 'application/pdf', 'Promotional Offer'); end; /</pre>
Terms and Conditions	application/pdf	<pre>begin add_content('CONTENT_FILES', 'TnC.pdf', 'application/pdf', 'Terms and Conditions'); end; /</pre>
Welcome Letter	application/pdf	<pre>begin add_content('CONTENT_FILES', 'WelcomeLetter.pdf', 'application/pdf', 'Welcome Letter'); end; /</pre>

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

- **For Feedback**

Proc Execution Statement:

File Name	Proc Execution Statement
Heart-fill.svg	<pre>begin load_file_to_content('CONTENT_FILES','heart-fill.svg', '\${contentId1}'); end; /</pre>
Star.png	<pre>begin load_file_to_content('CONTENT_FILES','star.png ', '\${contentId2}'); end; /</pre>
Thumbs-up.svg	<pre>begin load_file_to_content('CONTENT_FILES','thumbs-up.svg ', '\${contentId3}'); end; /</pre>

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:**

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

- **For User Print Document Configuration:**

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

- **For Feedback**

```
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;
```

[Home](#)

3. ATM/Branch Locator

3.1 Maintaining Default Radius

Default Radius of ATM/Branch Search is 2.5 Kms which can be changed by updating the value of DEFAULT_SEARCH_RADIUS in digx_fw_config_all_b. The updated value will be in effect post server restart.

[Home](#)