**CITCO – Customer Tax BIP Report Change Details**

**Data Models – SQL query**

| **S.No** | **Entity Name** | **CITCO** | **Maximise** | **Comments** |
| --- | --- | --- | --- | --- |
| 1. | Tax  Profile | select  -- zxptp.request\_id ,  -- to\_char(zxptp.last\_update\_date,'DD/MM/YYYY') ,  -- zxptp.last\_updated\_by,  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  PROCESS\_FOR\_APPLICABILITY\_FLAG as PROCESS\_FOR\_APPLICABILITY\_FLAG,  ROUNDING\_LEVEL\_CODE as ROUNDING\_LEVEL\_CODE,  ROUNDING\_RULE\_CODE as ROUNDING\_RULE\_CODE,  TAX\_CLASSIFICATION\_CODE as TAX\_CLASSIFICATION\_CODE,  INCLUSIVE\_TAX\_FLAG as INCLUSIVE\_TAX\_FLAG,  ALLOW\_OFFSET\_TAX\_FLAG as ALLOW\_OFFSET\_TAX\_FLAG,  COUNTRY\_CODE as COUNTRY\_CODE,  REGISTRATION\_TYPE\_CODE as REGISTRATION\_TYPE\_CODE,  REP\_REGISTRATION\_NUMBER as REP\_REGISTRATION\_NUMBER  from  ZX\_PARTY\_TAX\_PROFILE zxptp,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where zxptp.record\_type\_code = 'FUSION\_RI\_UPLOAD'  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxptp.party\_id = hzps.party\_site\_id  and hzps.party\_id = hzp.party\_id  and zxptp.rounding\_level\_code='HEADER'  and zxptp.request\_id = nvl(:load\_request\_id,zxptp.request\_id)  and zxptp.last\_updated\_by = nvl(:lastUpdatedBy,zxptp.last\_updated\_by)  and to\_char(zxptp.last\_update\_date,'DD/MM/YYYY')  = nvl(:last\_update\_date,to\_char(zxptp.last\_update\_date,'DD/MM/YYYY')) | select  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  PROCESS\_FOR\_APPLICABILITY\_FLAG as PROCESS\_FOR\_APPLICABILITY\_FLAG,  ROUNDING\_LEVEL\_CODE as ROUNDING\_LEVEL\_CODE,  ROUNDING\_RULE\_CODE as ROUNDING\_RULE\_CODE,  TAX\_CLASSIFICATION\_CODE as TAX\_CLASSIFICATION\_CODE,  INCLUSIVE\_TAX\_FLAG as INCLUSIVE\_TAX\_FLAG,  ALLOW\_OFFSET\_TAX\_FLAG as ALLOW\_OFFSET\_TAX\_FLAG,  COUNTRY\_CODE as COUNTRY\_CODE,  REGISTRATION\_TYPE\_CODE as REGISTRATION\_TYPE\_CODE,  REP\_REGISTRATION\_NUMBER as REP\_REGISTRATION\_NUMBER  from  ZX\_PARTY\_TAX\_PROFILE zxptp,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where zxptp.record\_type\_code = 'FUSION\_RI\_UPLOAD'  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxptp.party\_id = hzps.party\_site\_id  and hzps.party\_id = hzp.party\_id  and zxptp.request\_id = :load\_request\_id  and zxptp.last\_updated\_by =:lastUpdatedBy | 1.Usage of NVL operation in the below columns – Citco.  and zxptp.request\_id = nvl(:load\_request\_id,zxptp.request\_id)  and zxptp.last\_updated\_by = nvl(:lastUpdatedBy,zxptp.last\_updated\_by)  2.New column used in where condition – Citco  and zxptp.rounding\_level\_code='HEADER'  3.Added last updated date logic in Citco code.  and to\_char(zxptp.last\_update\_date,'DD/MM/YYYY')  = nvl(:last\_update\_date,to\_char(zxptp.last\_update\_date,'DD/MM/YYYY')) |
| 2. | Tax  Registration | select  zxptp.request\_id,  zxr.created\_by ,  to\_char(zxr.creation\_date,'DD/MM/YYYY') creation\_date,  zxr.last\_updated\_by ,  to\_char(zxr.last\_update\_date,'DD/MM/YYYY') last\_update\_date,  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  zxr.TAX\_REGIME\_CODE as TAX\_REGIME\_CODE,  zxr.TAX as TAX,  zxr.TAX\_JURISDICTION\_CODE as TAX\_JURISDICTION\_CODE,  zxr.REGISTRATION\_TYPE\_CODE as REGISTRATION\_TYPE\_CODE,  zxr.REGISTRATION\_NUMBER as REGISTRATION\_NUMBER,  to\_char(zxr.EFFECTIVE\_FROM,'dd/mm/yyyy') as EFFECTIVE\_FROM,  to\_char(zxr.EFFECTIVE\_TO,'dd/mm/yyyy') as EFFECTIVE\_TO,  zxr.DEFAULT\_REGISTRATION\_FLAG as DEFAULT\_REGISTRATION\_FLAG,  zxr.REGISTRATION\_REASON\_CODE as REGISTRATION\_REASON\_CODE,  zxr.REGISTRATION\_STATUS\_CODE as REGISTRATION\_STATUS\_CODE,  zxr.REGISTRATION\_SOURCE\_CODE as REGISTRATION\_SOURCE\_CODE,  zxr.REP\_PARTY\_TAX\_NAME as REP\_PARTY\_TAX\_NAME,  zxr.ROUNDING\_RULE\_CODE as ROUNDING\_RULE\_CODE,  zxr.ROUNDING\_LEVEL\_CODE as ROUNDING\_LEVEL\_CODE,  zxptp.SELF\_ASSESS\_FLAG as SELF\_ASSESS\_FLAG,  zxr.INCLUSIVE\_TAX\_FLAG as INCLUSIVE\_TAX\_FLAG,  zxr.TAX\_POINT\_BASIS,  null as ISSUING\_TAX\_AUTHORITY,  zxr.COLL\_TAX\_AUTHORITY\_ID as COLLECTING\_TAX\_AUTHORITY,  zxr.REP\_TAX\_AUTHORITY\_ID as REPORTING\_TAX\_AUTHORITY,  zxr.VALIDATION\_TYPE,  zxr.SELF\_ASSESS\_DIFF\_TOL\_LMT\_FLAG,  zxr.MIN\_SELF\_ASSESS\_DIFF\_TOL\_LIMIT,  zxr.MAX\_SELF\_ASSESS\_DIFF\_TOL\_LIMIT  from  ZX\_PARTY\_TAX\_PROFILE zxptp,  ZX\_REGISTRATIONS zxr,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where zxptp.record\_type\_code = 'FUSION\_RI\_UPLOAD'  and zxptp.CUSTOMER\_FLAG = 'Y'  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxr.PARTY\_TAX\_PROFILE\_ID = zxptp.PARTY\_TAX\_PROFILE\_ID  and zxptp.party\_id = hzps.party\_site\_id  and hzps.party\_id = hzp.party\_id  and zxptp.request\_id = NVL(:load\_request\_id,zxptp.request\_id)  and zxr.created\_by = NVL(:pcreatedBY,zxr.created\_by)  and to\_char(zxr.last\_update\_date,'DD/MM/YYYY')  = nvl(:last\_update\_date,to\_char(zxr.last\_update\_date,'DD/MM/YYYY')) | select  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  zxr.TAX\_REGIME\_CODE as TAX\_REGIME\_CODE,  zxr.TAX as TAX\_CODE,  zxr.TAX\_JURISDICTION\_CODE as TAX\_JURISDICTION\_CODE,  zxptp.REGISTRATION\_TYPE\_CODE as REGISTRATION\_TYPE\_CODE,  zxptp.REP\_REGISTRATION\_NUMBER as REGISTRATION\_NUMBER,  to\_char(zxr.EFFECTIVE\_FROM,'dd/mm/yyyy') as EFFECTIVE\_FROM,  to\_char(zxr.EFFECTIVE\_TO,'dd/mm/yyyy') as EFFECTIVE\_TO,  zxr.DEFAULT\_REGISTRATION\_FLAG as DEFAULT\_REGISTRATION\_FLAG,  zxr.REGISTRATION\_REASON\_CODE as REGISTRATION\_REASON\_CODE,  zxr.REGISTRATION\_STATUS\_CODE as REGISTRATION\_STATUS\_CODE,  zxr.REGISTRATION\_SOURCE\_CODE as REGISTRATION\_SOURCE\_CODE,  zxr.REP\_PARTY\_TAX\_NAME as REP\_PARTY\_TAX\_NAME,  zxptp.ROUNDING\_LEVEL\_CODE as ROUNDING\_LEVEL\_CODE,  zxptp.SELF\_ASSESS\_FLAG as SELF\_ASSESS\_FLAG,  zxptp.INCLUSIVE\_TAX\_FLAG as INCLUSIVE\_TAX\_FLAG  from  ZX\_PARTY\_TAX\_PROFILE zxptp,  ZX\_REGISTRATIONS zxr,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where zxptp.record\_type\_code = 'FUSION\_RI\_UPLOAD'  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxr.PARTY\_TAX\_PROFILE\_ID = zxptp.PARTY\_TAX\_PROFILE\_ID  and zxptp.party\_id = hzps.party\_site\_id  and hzps.party\_id = hzp.party\_id  and zxptp.request\_id = NVL(:load\_request\_id,zxptp.request\_id)  and zxr.created\_by = NVL(:pcreatedBY,zxr.created\_by) | 1.Additional new columns added in Citco.  zxptp.request\_id,  zxr.created\_by ,  to\_char(zxr.creation\_date,'DD/MM/YYYY') creation\_date,  zxr.last\_updated\_by ,  to\_char(zxr.last\_update\_date,'DD/MM/YYYY') last\_update\_date,  2.Citco is fetching Registeration\_type\_code and Registeration\_Number  From ZX\_REGISTRATIONS table.  Where maximise is fetching Registeration\_type\_code and Registeration\_Number From ZX\_PARTY\_TAX\_PROFILE table.  3.Blue highlight are the new columns added in Citco.  4.New column added in Citco  zxr.ROUNDING\_RULE\_CODE as ROUNDING\_RULE\_CODE.  5. Citco is fetching Rounding\_level\_code from ZX\_REGISTRATIONS table.  Maximise is fetching Rounding\_level\_code from ZX\_PARTY\_TAX\_PROFILE table. |
| 3 | Party Classification | -- Change List  -- ===========  --  -- Date Author Comment  -- ---------- ----------------- -----------------------------------------  -- Michal Arrowsmith Initial Version  --  select  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  hzca.CLASS\_CATEGORY as CLASS\_CATEGORY,  zxft.CLASSIFICATION\_TYPE\_CODE as CLASSIFICATION\_TYPE\_CODE,  zxfttl.CLASSIFICATION\_TYPE\_NAME as CLASSIFICATION\_TYPE\_NAME,  null as TAX\_REGIME\_CODE,  hzca.CLASS\_CODE as CLASS\_CODE,  null as FISCAL\_CLASSIFICATION\_CODE,  to\_char(hzca.START\_DATE\_ACTIVE,'dd/mm/yyyy') as EFFECTIVE\_FROM,  to\_char(hzca.END\_DATE\_ACTIVE,'dd/mm/yyyy') as EFFECTIVE\_TO  from  HZ\_CODE\_ASSIGNMENTS hzca,  ZX\_FC\_TYPES\_B zxft,  ZX\_FC\_TYPES\_TL zxfttl,  ZX\_PARTY\_TAX\_PROFILE zxptp,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where hzca.CREATED\_BY\_MODULE = 'EBTAX\_CONTENT\_UPLOAD'  and hzca.CLASS\_CATEGORY = 'FISCAL CLASSIFICATION'  and zxft.OWNER\_ID\_CHAR = hzca.CLASS\_CATEGORY  and zxfttl.CLASSIFICATION\_TYPE\_ID = zxft.CLASSIFICATION\_TYPE\_ID  and zxptp.party\_tax\_profile\_id = hzca.owner\_table\_id  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxptp.party\_id = hzps.party\_site\_id  --and hzp.party\_id = zxptp.party\_id  and hzps.party\_id = hzp.party\_id  and zxptp.request\_id = NVL(:load\_request\_id,zxptp.request\_id)  and hzca.created\_by = NVL(:pCreatedBy,hzca.created\_by) | select  hzps.PARTY\_SITE\_NUMBER as PARTY\_NUMBER,  hzp.PARTY\_NAME as PARTY\_NAME,  decode(PARTY\_TYPE\_CODE,  'THIRD\_PARTY\_SITE','Customer site',  PARTY\_TYPE\_CODE) as PARTY\_TYPE\_CODE,  hzca.CLASS\_CATEGORY as CLASS\_CATEGORY,  zxft.CLASSIFICATION\_TYPE\_CODE as CLASSIFICATION\_TYPE\_CODE,  zxfttl.CLASSIFICATION\_TYPE\_NAME as CLASSIFICATION\_TYPE\_NAME,  null as TAX\_REGIME\_CODE,  hzca.CLASS\_CODE as CLASS\_CODE,  hzca.CLASS\_CODE as CLASS\_NAME,  to\_char(hzca.START\_DATE\_ACTIVE,'dd/mm/yyyy') as EFFECTIVE\_FROM,  to\_char(hzca.END\_DATE\_ACTIVE,'dd/mm/yyyy') as EFFECTIVE\_TO  from  HZ\_CODE\_ASSIGNMENTS hzca,  ZX\_FC\_TYPES\_B zxft,  ZX\_FC\_TYPES\_TL zxfttl,  ZX\_PARTY\_TAX\_PROFILE zxptp,  HZ\_PARTY\_SITES hzps,  HZ\_PARTIES hzp  where hzca.CREATED\_BY\_MODULE = 'EBTAX\_CONTENT\_UPLOAD'  and hzca.CLASS\_CATEGORY = 'REGISTERED'  and zxft.OWNER\_ID\_CHAR = hzca.CLASS\_CATEGORY  and zxfttl.CLASSIFICATION\_TYPE\_ID = zxft.CLASSIFICATION\_TYPE\_ID  and zxptp.party\_tax\_profile\_id = hzca.owner\_table\_id  and zxptp.party\_type\_code = 'THIRD\_PARTY\_SITE'  and zxptp.party\_id = hzps.party\_site\_id  --and hzp.party\_id = zxptp.party\_id  and hzps.party\_id =hzp.party\_id  and zxptp.request\_id = NVL(:load\_request\_id,zxptp.request\_id)  and hzca.created\_by = NVL(:pCreatedBy,hzca.created\_by) | 1.Class Category values are different, both in maximise and Citco.  2.Pink highlight, change in column names. |