

# BANKA SLOVENIJE

## TEHNIČNA POJASNILA GLEDE IZPOLNJEVANJA ITS POROČIL

Tip dokumenta	<b>IT-TN</b>
Oznaka dokumenta	<b>BS-IT-RA-TN-ITS-dod</b>
Verzija dokumenta	<b>1.4</b>
Status dokumenta	
Datum zadnje spremembe dokumenta	<b>23.11.2017</b>

### Zgodovina sprememb dokumenta:

Verzije dokumenta	Datum potrditve	Opis sprememb dokumenta
1.0	13.12.2013	Prva verzija dokumenta
1.1	30.01.2014	"OkvirPorocila" in "PodlagaPorocanja" nadomeščena z "ModulPorocila", popravek izhodnega formata procenta
1.2	05.05.2014	Verzija DPM 2.1: uvedba podstolpcev, ključi, verzioniranje tabel, modulov, validacijska pravila
1.3	06.01.2016	Dodan podatkovni tip r
1.4	23.11.2017	Popravki zaradi preнове spletne strani BS

# BANKA SLOVENIJE

## Kazalo

1.	Osnovni pojmi in definicije .....	3
2.	Osnova za poročanje .....	4
3.	Obveznost postavk poročila .....	4
4.	Oznake listov .....	5
5.	Podvrstice in podstolpci .....	6
6.	Podatkovni tipi .....	7
7.	Verzije taksonomij, modulov, tabel.....	8
8.	Validacijska pravila.....	10

## 1. Osnovni pojmi in definicije

**Taksonomija** = Najširši nabor zahtevanih elementarnih podatkov FINREP in COREP za ITS poročanje in njihovih medsebojnih povezav, hierarhije, validacije

**DPM** = formalna predstavitev taksonomije (taksonomija v accessovi bazi) – za komunikacijo med uporabniki (e.g. between business and IT experts, or between various groups of business experts).

**XBRL taxonomija** = tehnični format za izmenjavo podatkov taksonomije

**obrazec** = Obrazec je vsebinsko (uporabniško) definiran v excel template obliki. V DPM-ju je lahko razbit v več tabel (npr. obrazec F 20.05 ima tabeli F 20.05.a in F 20.05 .b v DPM)

**tabela** = oznaka iz DPM. Če vzamemo prvih 7 mest oznake tabele, dobimo 'uporabniško' oznako obrazca.

**modul** = podmnožica vseh tabel za določen okvir (FINREP/COREP) in podlago (konsolidirana/individualna). Znotraj modula je lahko nabor tabel posamezne poročevalke tudi manjši glede na obveznost poročanja posameznih tabel (obrazcev).

**poročilo** = vsi podatki tabel (obrazcev) za poročevalko/datum podatkov/modul (oznaka modula po vsebini vsebuje informacije o okviru in podlagi)

Veljavni moduli 2.12.2013 (EBA) za naše poročevalke:

- COREP\_Con,
- COREP\_Ind,
- COREP\_LCR\_Con,
- COREP\_LCR\_Ind,
- COREP\_LE\_Con,
- COREP\_LE\_Ind,
- COREP\_NSFR\_Con,
- COREP\_NSFR\_Ind,
- FINREP\_Con\_IFRS

Nabor vseh obrazcev (tabel), ki jih zajema posamezni modul, je v ITS\_Moduli\_tabele\_povezave.csv.

**Posamezno poročilo predstavlja enoto (osnovo) za izvajanje kontrol/potrjevanje/zavračanje!**

### Data model (povzeto po objavah na EBA)

Prior to the development of an XBRL taxonomy (which is a technical format used for data exchange), information requirements need to be identified by specifying reportable pieces of information. This is usually done in the form of data models. Data models organize the data for communication purposes (e.g. between business and IT experts, or between various groups of business experts). In the case of CRR reporting, the inputs for creation of the data model are Implementing Technical Standards, consisting of the main provisions covering the reporting requirements, the reporting templates, i.e. tabular representation of information requirements, the instructions associated with these templates, and the related validation formulae. These templates, provisions, instructions and underlying regulations are analysed according to the Data Point Modelling methodology in order to create a Data Point Model format. In the case of the ITS data model, the DPM format consists primarily of a structured Microsoft Access database, the content of which is also documented via two Microsoft Excel workbooks:

Dictionary - defining properties (and their classifications/breakdowns) that can be used to describe each exchanged piece of information, and hierarchical relations between them.

Table Layout and Data Point Categorisation - Annotated tables where each row/column/sheet is associated with a property or a set of properties defined in the dictionary.

As a result, the DPM database defines a set of reportable cells (data points) in tables by specifying all of the properties (according to the content of the dictionary) required to convey their full meaning.

The preparatory taxonomy was created by (automated) translation of the DPM database format into XBRL syntax based on the rules described in this document.

## 2. Osnova za poročanje

### Verzija v2.0 - 02.12.2013

EBA je objavila dokumente v zvezi z ITS poročanjem:

<http://www.eba.europa.eu/-/eba-publishes-xbrl-taxonomy-for-remittance-of-supervisory-reporting-by-competent-regulatory-authorities>

Iz dokumentacije na EBA-strani, ki sta osnova za sprejem podatkov ITS s strani bank v BS:

1. DPM Documents.zip => **Annotated DPM Table Layout and Data Point Categorisation v2.0.0 with z axes unrolled.xlsx**
2. DPM Documents.zip => **Annex XV (DPM Validation formulae) - v2.0.0.xlsx**

Podatki iz DPMja nam bodo služili za osnovno kontrolo pri sprejemu podatkov. Iz DPM access baze z dne 2.12.2013 smo pripravili **dpm\_postavke\_20131202.csv** - podatke o celicah vseh tabel (tabela, list, vrstica, stolpec, datatype, opis stolpca, opis vrstice, ali je row key, tableid, tablevid, datapointid, datapointvid). Vseh celic je 35.516 v 161 elementarnih tabelah, ki predstavljajo 120 uporabniških obrazcev.

Dne 16.4.2014 je bil objavljen nov Annex XV (DPM Validation formulae) z odklopljenimi kontrolami za verzijo 2.0 ( <http://www.eba.europa.eu/-/eba-publishes-list-of-incorrect-its-validation-rules> )

### Verzija v2.1 - 21.3.2014

EBA: <http://www.eba.europa.eu/-/eba-consults-on-a-revised-xbrl-taxonomy-for-supervisory-reporting>

## 3. Obveznost postavk poročila

Vsako posamezno poročilo ima isti DatumPodatkov in isto ReferencnaStevilkaPorocila.

Znotraj enega poročila mora biti celoten nabor predvidenih obrazcev za poročevalka/datum/ModulPorocila. Če podatki posameznih postavk (celic) ne obstajajo, se ne posredujejo (nikakor se ne sme poročati ničel). Ravno tako ne posredujete obrazcev, za katere ne presegate praga. (Podobno kot v obstoječem Corep poročanju bomo za vsako posamezno banko glede na obstoječe stanje opredelili, katere obrazce banka pošilja in katerih ne. V primeru, da mora banka zaradi širitve poslov posredovati dodatna poročila, se poročanje obrazca "odpre").

Znotraj enega poročila lahko sklope pošiljate v ločenih datotekah, vendar se morajo vsi deli sklicevati na eno poročilo (ista ReferencnaStevilkaPorocila, StDelnihPorocil = število datotek, ZapStDelnegaPorocila, npr. za tri datoteke 1/3,2/3,3/3).

Podatek VrstaPorocila je le informativne narave (navadno ali popravki ali po reviziji), ravno tako Opomba.

Celotno ITS poročilo mora vsebovati vse obrazce, ki jih je poročevalec dolžan poročati in ima podatke, ki so zahtevani na obrazcu. Če za posamezen obrazec ali list ali celico nima podatkov, tega ne poroča.

DPM vsebuje najširši možen nabor celic za poročanje FINREP/COREP taksonomije. Nekaterih celic IFRS poročevalke (vse naše banke so IFRS poročevalke) ne smejo posredovati. Zato bomo dodatno označili celice tabel z oznako, ki bo označevala, ali je poročanje le-te obvezno, neobvezno ali prepovedano (seznam je v pripravi).

## 4. Oznake listov

Če je v DPMju obrazec brez oznake lista, se posreduje vrednost lista '000', sicer pa tako, kot je označeno v DPMju. Kljub temu, da ima obrazec samo en list, ima lahko v DPM-ju svojo oznako lista (večina takih je brez oznake lista). Tabele z enim listom z oznako lista 010 :

C 51.00.a	010
C 51.00.b	010
C 52.00.a	010
C 52.00.b	010
C 52.00.c	010
C 52.00.d	010
C 53.00.a	010
C 53.00.b	010
C 53.00.c	010
C 54.00.a	010
C 60.00.a	010
C 60.00.b	010
C 61.00.a	010
C 61.00.b	010

Nekatere tabele imajo vnaprej definirane vse možne oznake listov. Posamezna oznaka lista se navezuje na točno določeno vsebino lista. Primer: tabela C 21.00 ima možnih 42 listov: list 001=total, 002=Austria... 042=Other.

Če bodo posredovani podatki take tabele, bo kontrola, da se poleg lista z npr 002=Austria posreduje tudi 001=Total. Posredovanje samo postavk lista 001=Total brez dodatnega lista ni dovoljeno.

Kadar je obrazec označen kot '**open sheet**' (=open Z, vrednost lista v DPM='999'), se v vrednost lista v xml vpiše šifra, na katero se nanaša glede na šifrant za ta obrazec ([glej Seznam dovoljenih vrednosti celic in listov – popravek 2.4.2014](#))

Primer: tabela C 09.01.a: za oznako lista za Albanijo se posreduje AL (MemberCode), za Avstrijo AT. Lista vrednosti vsebuje 250 držav in je podmnožica memberjev domene 250. Select iz Access-a:

```
SELECT Hierarchy.HierarchyCode, Hierarchy.HierarchyLabel, TableVersion.TableVersionCode, Member.MemberID, Member.DomainID, Member.MemberCode, Member.MemberLabel, Member.IsDefaultMember, Member.MemberXbrlCode, OpenAxisValueRestriction.AxisID, Axis.TableVID, Axis.AxisOrientation, Axis.AxisLabel, Axis.AxisOrder, Axis.IsOpenAxis
FROM TableVersion INNER JOIN (Axis INNER JOIN (Member INNER JOIN ((Hierarchy INNER JOIN HierarchyNode ON Hierarchy.HierarchyID = HierarchyNode.HierarchyID) INNER JOIN OpenAxisValueRestriction ON HierarchyNode.HierarchyID = OpenAxisValueRestriction.HierarchyID) ON Member.MemberID = HierarchyNode.MemberID) ON Axis.AxisID = OpenAxisValueRestriction.AxisID) ON TableVersion.TableVID = Axis.TableVID;
```

V xml shemi je dolžina oznake lista daljša kot bi glede na šifrante lahko bila.

# BANKA SLOVENIJE

## 5. Podvrstice in podstolpci

V primeru poročanja ponavljajočih vrstic (v DPMju z oznako vrstice '999', open Y) ali podstolpcev (v DPMju open X) se za vsako celico te podvrstice/podstolpca v xml vpiše enolično oznako.

V posamezni tabeli s podstolpci/podvrsticami je lahko enoličen ključ sestavljen iz več celic (sestavljen ključ). Oznaka podvrstice/podstolpca lahko vsebuje vrednost ključa ali drugo enolično oznako.

Ključni so vsebinsko: matična številka, ISIN koda, sekvenca, ... (glej dpm\_IsRowKey\_postavka.xlsx).

AxisOrdnate
OrdinateLabel
Entity code
Group code
Code
Holding company code
Security code
ROW NUMBER
Obligor grade

Pri posredovanih podatkih mora biti:

- oznaka podvrstice/podstolpca enolična (lahko vsebuje vrednost ključa oz. vrednosti ključev, če je sestavljen ključ ali poljubno enolično oznako)
- v posamezni podvrstici/podstolpcu morajo biti izpolnjene vse celice, ki so ključ (isrowkey='da'): npr C 29.00 ima dva ključa: c010 in c020)
- vrednosti celic ključev posamezne podvrstice/podstolpca (sestavljen ključ) morajo biti enolične za vse poročane podvrstice/podstolpce

Primer (izbrane vrednosti so zgolj za lažjo predstavo):

C 27.00 - Identification of the counterparty				Columns							
				COUNTERPARTY IDENTIFICATION							
				Code	Name	LEI code	Residence of the counterparty	Sector of the counterparty	NACE code	Type of counterparty	
				010	020	030	040	050	060	070	
Oznaka Podvrstice	Row	Open	999	a	aaa	xxx	xxx	xxx	xxx	xxx	Pravilno?
1	Row	Open	999	b	aaa	xxx	xxx	xxx	xxx	xxx	✓
2	Row	Open	999	c	aaa	xxx	xxx	xxx	xxx	xxx	✓
3	Row	Open	999	d	aaa	xxx	xxx	xxx	xxx	xxx	✓
3	Row	Open	999	d	aaa	xxx	xxx	xxx	xxx	xxx	narobe (podvr. ni enolična)
5	Row	Open	999	a	aaa	xxx	xxx	xxx	xxx	xxx	narobe (ključ je že v podvr 1)

## 6. Podatkovni tipi

Posamezna celica ima datatype, ki pomeni za izpolnjevanje v XML naslednje:

- DATATYPE b Boolean vrednost DA ali NE
- DATATYPE d Date format YYYY-MM-DD
- DATATYPE i Integer celo število
- DATATYPE p Percentage ločilo za decimalno mesto je pika, max. 4 decimalke, večinoma vrednost med 0 in 1 (0.1234 za 12,34%, 0.1 za 10%)
- DATATYPE m Monetary celo število v tisočih, predznak spredaj, brez ločil za tisoč, brez decimalk
- DATATYPE s String string do dolžine 255
- DATATYPE e Code string do dolžine 255
- DATATYPE r Decimal ločilo za decimalno mesto je pika
- celice, ki nimajo oznake datatype (key value ='DA') string do dolžine 255

## 7. Verzije taksonomij, modulov, tabel

V verziji access v2.1 so meta podatki za poročanje po verziji v2.0 in v2.1.

Za nove verzije Taksonomij veljavne od-do so ustvarjeni novi id-ji modulov (ModulCode, ki se poroča, je ostal nespremenjen). Za vsakokratno veljaven ModulID je potrebno poiskati veljavne TableVID (TableVersionCode je nespremenjen). Posamezen TableVID ima tudi datum od-do in svoj nabor celic (postavk). Katere postavke vsebuje posamezen TableVID, dobimo preko poizvedb CellOrdinates in TableStructure (glej v nadaljevanju).

V nadaljevanju popravki poizvedb (spremenili so imena nekaterih kolon tabel) in prikaz povezav med tabelami:

### CellOrdinates

```
TRANSFORM First(CellPosition.OrdinateID) AS FirstOfOrdinateID
SELECT TableVersion.TableVersionCode, TableVersion.TableVID, TableCell.CellID, TableCell.DataPointVID, TableCell.IsShaded
FROM TableVersion INNER JOIN (TableCell INNER JOIN ((Axis INNER JOIN AxisOrdinate ON Axis.AxisID = AxisOrdinate.AxisID) INNER JOIN
CellPosition ON AxisOrdinate.OrdinateID = CellPosition.OrdinateID) ON TableCell.CellID = CellPosition.CellID) ON (TableVersion.TableVID =
TableCell.TableVID) AND (TableVersion.TableVID = Axis.TableVID)
GROUP BY TableVersion.TableVersionCode, TableVersion.TableVID, TableCell.CellID, TableCell.DataPointVID, TableCell.IsShaded
PIVOT Axis.AxisOrientation;
```

### TableStructure

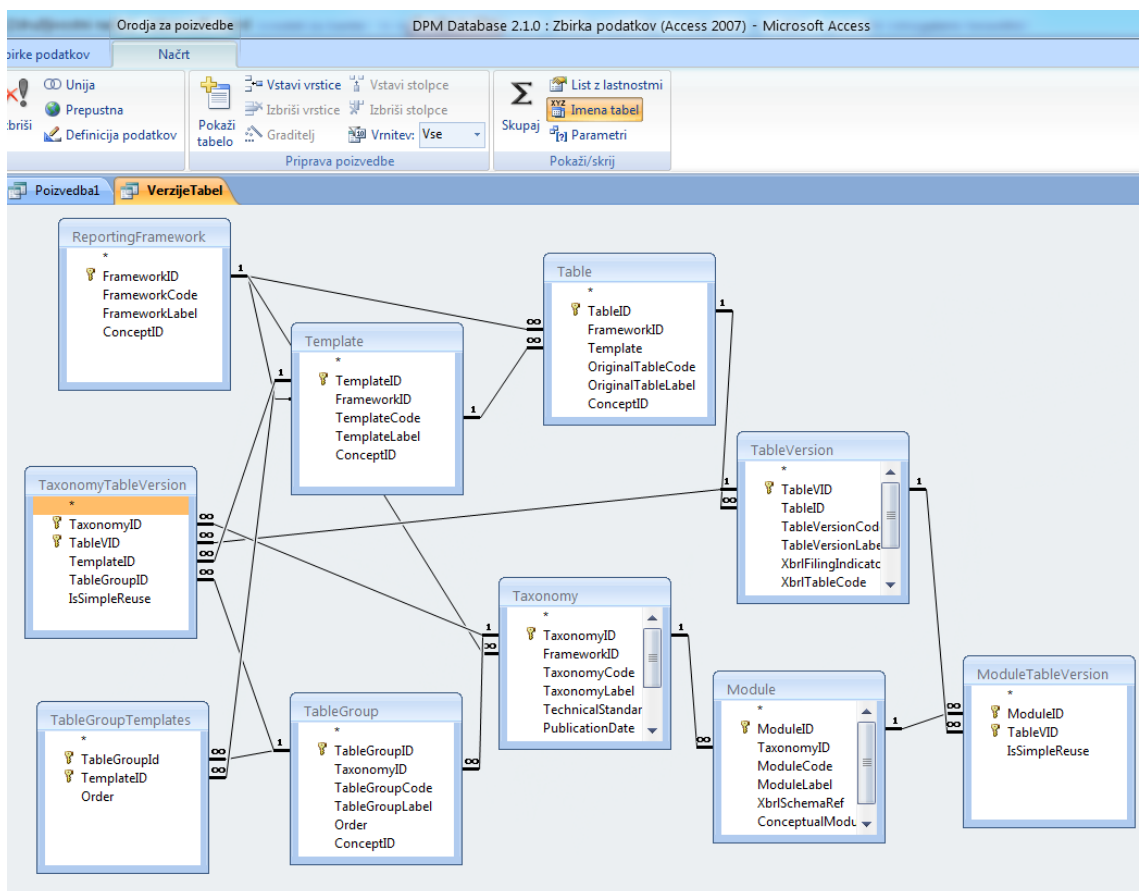
```
SELECT CellOrdinates.TableVID, CellOrdinates.TableVersionCode, CellOrdinates.CellID, CellOrdinates.DataPointVID, CellOrdinates.IsShaded,
Column.OrdinateCode, [Column].[OrdinateCode]+" - "+[Column].[OrdinateLabel] AS [Column], Row.OrdinateCode, [Row].[OrdinateCode]+" -
"+[Row].[OrdinateLabel] AS Row, Sheet.OrdinateCode, [TableVersionCode]+" - "+[Sheet].[OrdinateLabel] AS Sheet
FROM ((CellOrdinates LEFT JOIN [Column] ON CellOrdinates.X = Column.OrdinateID) LEFT JOIN Row ON CellOrdinates.Y = Row.OrdinateID) LEFT
JOIN Sheet ON CellOrdinates.Z = Sheet.OrdinateID;
```

### VerzijeTabel

```
SELECT Taxonomy.TaxonomyID, ReportingFramework.FrameworkCode, ReportingFramework.FrameworkLabel, ReportingFramework.ConceptID,
Taxonomy.FrameworkID, Taxonomy.TaxonomyCode, Taxonomy.TaxonomyLabel, Taxonomy.TechnicalStandard, Taxonomy.PublicationDate,
Taxonomy.Version, Taxonomy.ConceptID, Taxonomy.FromDate, Taxonomy.ToDate, TableGroup.TableGroupID, TableGroup.TableGroupCode,
TableGroup.TableGroupLabel, TableGroup.Order, TableGroup.ConceptID, TableGroupTemplates.Order, Template.TemplateCode, Template.TemplateLabel,
Template.ConceptID, TableVersion.TableVID, TableVersion.TableID, TableVersion.TableVersionCode, TableVersion.TableVersionLabel,
TableVersion.XbrlFilingIndicatorCode, TableVersion.XbrlTableCode, TableVersion.FromDate, TableVersion.ToDate, TableVersion.ConceptID,
Table.OriginalTableCode, Table.OriginalTableLabel, Table.ConceptID
FROM (((ReportingFramework INNER JOIN Template ON ReportingFramework.FrameworkID = Template.FrameworkID) INNER JOIN [Table] ON
(Template.TemplateCode = Table.Template) AND (ReportingFramework.FrameworkID = Table.FrameworkID)) INNER JOIN Taxonomy ON
ReportingFramework.FrameworkID = Taxonomy.FrameworkID) INNER JOIN TableVersion ON Table.TableID = TableVersion.TableID) INNER JOIN
([Module] INNER JOIN ModuleTableVersion ON Module.ModuleID = ModuleTableVersion.ModuleID) ON (Taxonomy.TaxonomyID =
Module.TaxonomyID) AND (TableVersion.TableVID = ModuleTableVersion.TableVID)) INNER JOIN ((TableGroup INNER JOIN TableGroupTemplates ON
TableGroup.TableGroupID = TableGroupTemplates.TableGroupId) INNER JOIN TaxonomyTableVersion ON TableGroup.TableGroupID =
TaxonomyTableVersion.TableGroupID) ON (Template.TemplateID = TaxonomyTableVersion.TemplateID) AND (Template.TemplateID =
TableGroupTemplates.TemplateID) AND (Taxonomy.TaxonomyID = TaxonomyTableVersion.TaxonomyID) AND (Taxonomy.TaxonomyID =
TableGroup.TaxonomyID) AND (TableVersion.TableVID = TaxonomyTableVersion.TableVID);
```



# BANKA SLOVENIJE



Iz verzije 2.0 na 2.1 so se spremenile postavke na tabelah:

verzije_tabel
XbrlTableCode
C_07.00.a
C_09.01.a
F_09.01
C_07.00.d
C_07.00.c
F_20.05.a
F_20.07
F_06.00
F_20.04
F_41.02
F_40.01
F_08.01.a
F_08.01.b
C_02.00
C_25.00
F_31.01
F_30.02

## 8. Validacijska pravila

Tipi validacijskih pravil:

- Identity (enolična vrednost dveh celic na različnih tabelah – enak datapointVID)
- Sign (predznak vrednosti celice)
- Hierarchy (hierarhične vsote podrejenih celic)
- Manual
- Unique identifier (enoličnost ključev – xbrl)
- Allowed values for metric (kontrola na seznam dovoljenih vrednosti za celice datatype=e, xbrl)
- Coherence check (xbrl)

V fazi K0 izvajamo:

Unique identifier  
Allowed values for metric

V fazi K1 izvajamo (pravila, ki zadevajo modul – preko ValidationRuleSet):

Identity  
Sign  
Hierarchy  
Manual

Coherence check ne izvajamo, saj podatke parametrskih tabel sami napolnimo preko drugih podatkov (modul, banka poročevalka).

Medmodulskih pravil v access ni (so samo v excel-u). Zaenkrat ob sprejemu kontroliramo le kontrole znotraj podatkov prejetega poročila (modula). Napake v kontrolah med moduli bodo lahko povzročile, da bo potrebno ponovno posredovati popravke podatkov že sicer potrjenega poročila.

Medmodulska pravila (iz excel-a, v2.0, 16.4.2014):

v0655_m	y	Manual	C 28.00	C 04.00	{C 28.00, c230} = {C 28.00, c210} / {C 04.00, r220, c010} * 100	treat as zero
v0656_m	y	Manual	C 28.00	C 04.00	{C 28.00, c350} = {C 28.00, c330} / {C 04.00, r220, c010} * 100	treat as zero
v0774_m	y	Manual	F 01.02	C 01.00	{F 01.02, r260, c010} >= {C 01.00, r330, c010} + {C 01.00, r360, c010} + {C 01.00, r410, c010} + {C 01.00, r490, c010}	treat as zero
v0781_m	y	Manual	F 01.03	C 01.00	{F 01.03, r040, c010} >= {C 01.00, r060}	treat as zero
v0782_m	y	Manual	F 01.03	C 01.00	abs({F 01.03, r090, c010}) > abs({C 01.00, r180, c010})	treat as zero
v1536_m	y	Manual	C 61.00.a	C 01.00	{C 61.00.a, r010, c050} = {C 01.00, r015, c010}	treat as zero
v1537_m	y	Manual	C 61.00.a	C 01.00	{C 61.00.a, r020, c050} = {C 01.00, r750, c010}	treat as zero
v1538_m	y	Manual	C 60.00.a	C 51.00.a	{C 60.00.a, r010, c010} = {C 51.00.a, r010, c030}	treat as zero
v1539_m	y	Manual	C 60.00.a	C 51.00.a	{C 60.00.a, r020, c010} = {C 51.00.a, r020, c030}	treat as zero
v1540_m	y	Manual	C 60.00.a	C 51.00.a	{C 60.00.a, r030, c010} = {C 51.00.a, r030, c030}	treat as zero
v1544_m	y	Manual	C 61.00.b	C 52.00.c	xsum({C 61.00.b, (r070, r140, c010-050)}) >= xsum({C 52.00.c, (r120-920, c020, c070)})	treat as zero
v1545_m	y	Manual	C 61.00.b	C 52.00.c	xsum({C 61.00.b, (r080, r150, c010-050)}) >= xsum({C 52.00.c, (r120-920, c040, c090)})	treat as zero
v1546_m	y	Manual	C 61.00.b	C 52.00.c	xsum({C 61.00.b, (r090, r160, c010-050)}) >= xsum({C 52.00.c, (r120-920, c060, c110, c120)})	treat as zero
v1547_m	y	Manual	C 61.00.b	C 52.00.a	sum({C 61.00.b, r240, (c010-050)}) >= {C 52.00.a, r1080, c020}	treat as zero
v1551_m	y	Manual	C 61.00.x	C 52.00.y	xsum({C 61.00.x, (r070, r140, c010-050)}) >= xsum({C 52.00.y, (r120-920, c020, c070)})	treat as zero
v1552_m	y	Manual	C 61.00.x	C 52.00.y	xsum({C 61.00.x, (r080, r150, c010-050)}) >= xsum({C 52.00.y, (r120-920, c040, c090)})	treat as zero
v1553_m	y	Manual	C 61.00.x	C 52.00.y	xsum({C 61.00.x, (r090, r160, c010-050)}) >= xsum({C 52.00.y, (r120-920, c060, c110, c120)})	treat as zero
v1554_m	y	Manual	C 61.00.x	C 52.00.w	sum({C 61.00.x, r240, (c010-050)}) >= {C 52.00.w, r1080, c020}	treat as zero
v1555_m	y	Manual	C 60.00.w	C 51.00.w	{C 60.00.w, r010, c010} = {C 51.00.w, r010, c030}	treat as zero
v1556_m	y	Manual	C 60.00.w	C 51.00.w	{C 60.00.w, r020, c010} = {C 51.00.w, r020, c030}	treat as zero
v1557_m	y	Manual	C 60.00.w	C 51.00.w	{C 60.00.w, r030, c010} = {C 51.00.w, r030, c030}	treat as zero
v3330_j	y	Identity	C 01.00	C 61.00.a	{C 01.00, r015, c010} = {C 61.00.a, r010, c050, s010}	do not run rule
v3331_j	y	Identity	C 01.00	C 61.00.a	{C 01.00, r750, c010} = {C 61.00.a, r020, c050, s010}	do not run rule

Pravila, ki so označena v excelu »Not implemented in XBRL = y«, so tista, ki se navezujejo na podatke med moduli, ali na preteklo obdobje (vsebujejo T-1) ali pa so bolj komplicirane in jih XBRL ne podpira. XBRL validator (podobno kot XML) lahko validira le v okviru ene datoteke (=modul/datum podatkov/poročevalka). V access DPM je ta podatek v EXPRESSION.EXPRESSIONTYPE.

# BANKA SLOVENIJE

Validacijska pravila imajo v 2.1 verziji nove ValidationID (za pravila iz verzije 2.0 velja, da imajo v ValidationRule.Severity='Error', nove pa imajo prazno vrednost).

Preko tabele ValidationRuleSet pridemo do nabora formul, ki jih izvajamo znotraj enega ModulID (dobimo iz modulCode na določen datum veljavnosti).

## verzije\_tabele\_formule

```
SELECT Taxonomy.TaxonomyID, ReportingFramework.FrameworkCode, ReportingFramework.FrameworkLabel, ReportingFramework.ConceptID, Taxonomy.FrameworkID, Taxonomy.TaxonomyCode, Taxonomy.TaxonomyLabel, Taxonomy.TechnicalStandard, Taxonomy.PublicationDate, Taxonomy.Version, Taxonomy.ConceptID, Taxonomy.FromDate, Taxonomy.ToDate, TableGroup.TableGroupID, TableGroup.TableGroupLabel, TableGroup.Order, TableGroup.ConceptID, TableGroupTemplates.Order, Template.TemplateCode, Template.TemplateLabel, Template.ConceptID, TableVersion.TableVID, TableVersion.TableID, TableVersion.TableVersionCode, TableVersion.TableVersionLabel, TableVersion.XbrlFilingIndicatorCode, TableVersion.XbrlTableCode, TableVersion.FromDate, TableVersion.ToDate, TableVersion.ConceptID, Table.OriginalTableCode, Table.OriginalTableLabel, Table.ConceptID FROM ValidationRule INNER JOIN (((((ReportingFramework INNER JOIN Template ON ReportingFramework.FrameworkID = Template.FrameworkID) INNER JOIN [Table] ON (Template.TemplateCode = Table.Template) AND (ReportingFramework.FrameworkID = Table.FrameworkID) INNER JOIN Taxonomy ON ReportingFramework.FrameworkID = Taxonomy.FrameworkID) INNER JOIN TableVersion ON Table.TableID = TableVersion.TableID) INNER JOIN (([Module] INNER JOIN ValidationRuleSet ON Module.ModuleID = ValidationRuleSet.ModuleID) INNER JOIN ModuleTableVersion ON Module.ModuleID = ModuleTableVersion.ModuleID) ON (Taxonomy.TaxonomyID = Module.TaxonomyID) AND (TableVersion.TableVID = ModuleTableVersion.TableVID)) INNER JOIN ((TableGroup INNER JOIN TableGroupTemplates ON TableGroup.TableGroupID = TableGroupTemplates.TableGroupID) INNER JOIN TaxonomyTableVersion ON TableGroup.TableGroupID = TaxonomyTableVersion.TableGroupID) ON (Template.TemplateID = TaxonomyTableVersion.TemplateID) AND (Template.TemplateID = TableGroupTemplates.TemplateID) AND (Taxonomy.TaxonomyID = TaxonomyTableVersion.TaxonomyID) AND (Taxonomy.TaxonomyID = TableGroup.TaxonomyID) AND (TableVersion.TableVID = TaxonomyTableVersion.TableVID)) ON ValidationRule.ValidationId = ValidationRuleSet.ValidationRuleId;
```

