**Condition and Policy Terms (Limits/Deductibles) and Other Account Information**

In OED, ~~almost~~ all condition level term fields start with letters like ‘CondLimit’ or ‘CondDed’ and all Policy level term fields start with letters like ‘PolLimit’ or ‘PolDed’ AK: all condition fields start with Cond and all policy terms start with Pol

SB: made the change.

For transformation, start with tLayerCondition table. Populate OED Account file fields as per the mapping logic provided below. First this will create an OED Account file with same number of records as there are in the tLayerConditon table.

In the pseudo code below, the OED Account file is assumed to be a table with table name ‘Account’. If any of the rules given below is not met, leave the corresponding OED Account file condition fields as blanks. For example, wherever there are texts highlighted in yellow, technically, they can be left blank i.e. without mapping to anything.

**Filling the CondName and CondNumber:**

To Fill CondName and CondNumber, use the same TagNumber table info that was used for filling CondNumber in OED location file. Get directly 2 fields (ContractSID, ApliesToTag) in tLayerCondition and then using this 2 column combination go and look up CondName and CondNumber from TagNumber.

AK: throughout all limit and deductible mappings – can they only be amounts? If not, some of the mappings would need to change.

SB: Yes, we will take care of this, as mentioned in the end of this sheet. But current timeline for POC is not allowing us to take care of this. We will do it in the next iteration (and will note this as a ToDo list)

**Fill Condition level Limit and Deductible terms from CEDE into OED Account File:**

If tLayerCondition.DeductibleTypeCode = ‘N’ and tLayerCondition.OccLimitTypeCode = ‘B’ then map

* + - 1. 0 = Account.CondLimitCode6All
      2. 0 = Account.CondLimitType6All
      3. tLayerCondition.OccLimit1 = Account.CondLimit6All
      4. tLayerCondition.Attachment1 = Account.CondDed6All AK: will it always be zero in this scenario?

SB: CEDE stores both Blanket Attachment and Building Attachment in the same DB field ‘Attachment1’. I have kept Attachment1 mapping for all Limit types while this is more relevant for by single Coverage Limit type.

Are you suggesting that for Blanket Limit, there is very little chance that someone will have an attachment point associated, while defining a sublimit condition?

* + - 1. 0 = Account.CondDedCode6All
      2. 0 = Account.CondDedType6All
      3. 0 = Account.CondMinDed6All
      4. 0 = Account.CondMaxDed6All

AK: note that in OED, min and max deductible fields accept only amounts (i.e. it can’t represent percentages). If e.g. 0.05 is entered it will be treated as, say 5 cents.

SB: AIR also suggests the same in their documentation. But I am not sure there is this validation done during AIR data import process and that we will never encounter a decimal in case of Min/Max.

ElseIf tLayerCondition.DeductibleTypeCode = ‘MI’ and tLayerCondition.OccLimitTypeCode = ‘B’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit1 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible1 = Account.CondMinDed6All
6. 0 = Account.CondDedCode6All
7. 0 = Account.CondDedType6All
8. 0 = Account.CondDed6All
9. 0 = Account.CondMaxDed6All

ElseIf tLayerCondition.DeductibleTypeCode = ‘MA’ and tLayerCondition.OccLimitTypeCode = ‘B’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit1 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible2 = Account.CondMaxDed6All
6. 0 = Account.CondDedCode6All
7. 0 = Account.CondDedType6All
8. 0 = Account.CondDed6All
9. 0 = Account.CondMinDed6All

ElseIf tLayerCondition.DeductibleTypeCode = ‘MM’ and tLayerCondition.OccLimitTypeCode = ‘B’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit1 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible1 = Account.CondMinDed6All
6. tLayerCondition.Deductible2 = Account.CondMaxDed6All
7. 0 = Account.CondDedCode6All
8. 0 = Account.CondDedType6All
9. 0 = Account.CondDed6All

🡺🡺🡺

Unlike CEDE, OED condition fields do not support Excess Limit layer with participation. Policy terms fields do support Layers. To transform Excess type limit into OED, the OccLimit2 has been mapped as Condition Limit instead of mapping OccLimit1. AK: is this the same participation that will then be taken into account at the layer level?

SB: OED has 4 fields “LayerAttachment, LayerLimit, LayerParticipation” and a LayerNumber to define an excess layer at policy level. This applies after Condition terms are already applied. In Condition terms itself in OED we cannot define an Excess Layer. We have the above 4 fields to define a Policy terms only.

However, AIR supports this excess layer definition at 2 levels: at layer level as well as at sublimit level. So, to transform AIR’s sublimit level excess terms, I was following an approximate method by assigning AIR’s sublimit participation amount to OED’s condition site limit amount.

I hope I made it clear.

If tLayerCondition.DeductibleTypeCode = ‘N’ and tLayerCondition.OccLimitTypeCode = ‘E’ then map

* + - 1. 0 = Account.CondLimitCode6All
      2. 0 = Account.CondLimitType6All
      3. tLayerCondition.OccLimit2 = Account.CondLimit6All
      4. tLayerCondition.Attachment1 = Account.CondDed6All
      5. 0 = Account.CondDedCode6All
      6. 0 = Account.CondDedType6All
      7. 0 = Account.CondMinDed6All
      8. 0 = Account.CondMaxDed6All

ElseIf tLayerCondition.DeductibleTypeCode = ‘MI’ and tLayerCondition.OccLimitTypeCode = ‘E’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit2 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible1 = Account.CondMinDed6All
6. 0 = Account.CondDedCode6All
7. 0 = Account.CondDedType6All
8. 0 = Account.CondDed6All
9. 0 = Account.CondMaxDed6All

ElseIf tLayerCondition.DeductibleTypeCode = ‘MA’ and tLayerCondition.OccLimitTypeCode = ‘E’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit2 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible2 = Account.CondMaxDed6All
6. 0 = Account.CondDedCode6All
7. 0 = Account.CondDedType6All
8. 0 = Account.CondDed6All
9. 0 = Account.CondMinDed6All

ElseIf tLayerCondition.DeductibleTypeCode = ‘MM’ and tLayerCondition.OccLimitTypeCode = ‘E’ then map

(2 deductible terms at same coverage level is unlikely to co-exist, unless there is a comparison, but has been mapped below to allow any one of them)

1. 0 = Account.CondLimitCode6All
2. 0 = Account.CondLimitType6All
3. tLayerCondition.OccLimit2 = Account.CondLimit6All
4. tLayerCondition.Attachment1 = Account.CondDed6All
5. tLayerCondition.Deductible1 = Account.CondMinDed6All
6. tLayerCondition.Deductible2 = Account.CondMaxDed6All
7. 0 = Account.CondDedCode6All
8. 0 = Account.CondDedType6All
9. 0 = Account.CondDed6All

🡺🡺🡺

ElseIf tLayerCondition.DeductibleTypeCode = ‘N’ and tLayerCondition.OccLimitTypeCode = ‘CB’ then map

1. 0 = Account.CondLimitCode5PD
2. 0 = Account.CondLimitType5PD
3. 0 = Account.CondDedCode5PD
4. 0 = Account.CondDedType5PD
5. tLayerCondition.OccLimit1 = Account.CondLimit5PD
6. tLayerCondition.Attachment1 = Account.CondDed5PD AK: always zero?

SB: AIR’s old UNICEDE document suggests using this field in case client wants to define an PD attachment amount for a PD Limit.

1. tLayerCondition.OccLimit4 = Account.CondLimit4BI
2. tLayerCondition.Attachment4 = Account.CondDed4BI AK: always zero?

SB: AIR’s old UNICEDE document suggests using this field in case client wants to define an BI attachment amount for a BI Limit under CB LimitType

1. 0 = Account.CondLimitCode4BI
2. 0 = Account.CondLimitType4BI
3. 0 = Account.CondDedCode4BI
4. 0 = Account.CondDedType4BI

🡺🡺🡺

ElseIf tLayerCondition.DeductibleTypeCode = ‘N’ and tLayerCondition.OccLimitTypeCode = ‘C’ then map

1. 0 = Account.CondLimitCode1Building
2. 0 = Account.CondLimitType1Building
3. 0 = Account.CondDedCode1Building
4. 0 = Account.CondDedType1Building
5. 0 = Account.CondMinDed1Building
6. 0 = Account.CondMaxDed1Building
7. 0 = Account.CondLimitCode2Other
8. 0 = Account.CondLimitType2Other
9. 0 = Account.CondDedCode2Other
10. 0 = Account.CondDedType2Other
11. 0 = Account.CondMinDed2Other
12. 0 = Account.CondMaxDed2Other
13. 0 = Account.CondLimitCode3Contents
14. 0 = Account.CondLimitType3Contents
15. 0 = Account.CondDedCode3Contents
16. 0 = Account.CondDedType3Contents
17. 0 = Account.CondMinDed3Contents
18. 0 = Account.CondMaxDed3Contents
19. 0 = Account.CondLimitCode4BI
20. 0 = Account.CondLimitType4BI
21. 0 = Account.CondDedCode4BI
22. 0 = Account.CondDedType4BI
23. 0 = Account.CondMinDed4BI
24. 0 = Account.CondMaxDed4BI AK: what is the reason for not including PD deductibles?

SB: You are correct we should have. But because this is anyways ‘0’, should not really matter! Later, I think we will have to remove all yellow highlighted 0 assignments to leave the corresponding fields as blanks and make this document clean.

1. tLayerCondition.OccLimit1 = Account.CondLimit1Building
2. tLayerCondition.Attachment1 = Account.CondDed1Building AK: always zero? SB: No, this case is to define a building limit and an associated building attachment – so can be Non-ZERO
3. tLayerCondition.OccLimit2 = Account.CondLimit2Other
4. tLayerCondition.Attachment2 = Account.CondDed2Other AK: always zero? SB: No, this case is to define a other limit and an associated other attachment – so can be Non-ZERO.
5. tLayerCondition.OccLimit3 = Account.CondLimit3Contents
6. tLayerCondition.Attachment3 = Account.CondDed3Contents AK: always zero? SB: No, this case is to define a contents limit and an associated contents attachment – so can be Non-ZERO.
7. tLayerCondition.OccLimit4 = Account.CondLimit4BI
8. tLayerCondition.Attachment4 = Account.CondDed4BI AK: always zero? SB: No, this case is to define a time limit and an associated time attachment – so can be Non-ZERO

**=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>**

AK: What does AP type mean? Why are you mapping deductible into Attachment field in OED instead of using PolDed6All field?

SB: AP means Attachment Point Type Deductible in AIR. AIR provides flexibility for client either to define Layer Attachment Point as an Excess Layer term or instead define a blanket deductible amount with type AP with an amount equivalent to Layer Attachment amount. They are treated one and the same and are stored in Deductible1 field in DB. This is just may be a flexibility that AIR gave, how many clients use it that way is something I am not sure.

I note that, as a normal workflow, if a client defines a Layer Attachment amount along with layer limit and participation, the condition below will work fine. Deductible1 is just a DB field name, it holds the Layer Attachment point amount as well.

**Fill Loss Layer terms of from CEDE into Layer Terms in OED Account File:**

If tLayer.OccLimitTypeCode = 'E' then Map 3 Layer Limit Fields of tLayer record to 3 Policy Layer fields of Account file:

* + - 1. tLayer.OccParticipation= Account.LayerParticipation

1. If tLayer.AttachmentPoint= 0 and tLayer.DeductibleTypeCode = ‘AP’ and tLayer.Deductible1 > 0

then tLayer.Deductible1 = Account.LayerAttachment

Else tLayer.AttachmentPoint = Account.LayerAttachment

1. tLayer.OccTotalLimit = Account.LayerLimit

**Fill Layer level Limit terms of AIR into Policy level Limit Terms in OED:**

Layer level limit and deductible Terms in AIR are on Total Coverage basis, so those PolLimit and PolDed fields in Account file need to be filled which have fieldnames ending with 'All'(i.e. PolLimitCode6All, PolLimitType6All, PolLimit6All fields)

Else If tLayer. OccLimitTypeCode = 'B 'then:

1. 0 = Account.PolLimitCode6All
2. 0 = Account. LayerAttachment
3. tLayer.OccTotalLimit = Account.LayerLimit

AK: all layers should be mapped to:

|  |
| --- |
| LayerParticipation |
| LayerLimit |
| LayerAttachment |

PolLimit6All shouldn’t be used for layers. It’s only used for aggregate limits. In this case layerLimit (and possibly LayerParticipation will need to be filled in) – SB: sorry, typo. I did not mean layer; this condition is for Blanket Site Limit.

Based on later comments, mapped Blanket Limit amount to LayerLimit with LayerAttachment = 0

Else tLayer. OccLimitTypeCode = 'N' then

1. 0 = Account.PolLimitCode6All
2. 0 = Account.PolLimitType6All
3. 0 = Account. LayerLimit AK: should be LayerLimit SB: Changed

(Though all ‘0’ assignments above can be left blank, coding this mapping with at least one field mapped as 0, is important to handle if conditions and that may be available in AIR Contract File – I reckon now that ‘0’ assignment fields need not be coded – but will confirm again)

**Fill Layer level Deductible terms of AIR into Policy level Deductible Terms in OED:**

Like for Limits, AIR Layer level Deductible Terms are on Total Coverage basis, so fill only fields with fieldname ending with 'All'.

If tLayer.DeductibleTypeCode = ‘B’ then map

1. 0 = Account.PolDed6All
2. 0 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. tLayer.Deductible1 = Account.PolDed6All
5. 0 = Account.PolMaxDed6All

ElseIf tLayer.DeductibleTypeCode = ‘FR’ then map

1. tLayer.Deductible1 = Account.PolDed6All
2. 2 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. 0 = Account.PolMinDed6All
5. 0 = Account.PolMaxDed6All

ElseIf tLayer.DeductibleTypeCode = ‘PL’ then map

tLayer.Deductible1 = Account.PolDed6All

0 = Account.PolDedCode6All

1 = Account.PolDedType6All

0 = Account.PolMinDed6All

0 = Account.PolMaxDed6All

ElseIf tLayer.DeductibleTypeCode = ‘MI’ then map

1. 0 = Account.PolDed6All
2. 0 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. tLayer.Deductible1 = Account.PolMinDed6All
5. 0 = Account.PolMaxDed6All

ElseIf tLayer.DeductibleTypeCode = ‘MA’ then map

1. 0 = Account.PolDed6All
2. 0 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. 0 = Account.PolMinDed6All
5. tLayer.Deductible2 = Account.PolMaxDed6All

ElseIf tLayer.DeductibleTypeCode = ‘MM’ then map

1. 0 = Account.PolDed6All
2. 0 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. tLayer.Deductible1 = PolMinDed6All
5. tLayer.Deductible2 = PolMaxDed6All

Else tLayer.DeductibleTypeCode = ‘N’ then map

1. 0 = Account.PolDed6All AK: always zero? SB: This is for DeductibleType = ‘N’ means no Deductible
2. 0 = Account.PolDedCode6All
3. 0 = Account.PolDedType6All
4. 0 = Account.PolMinDed6All
5. 0 = Account.PolMaxDed6All

Note: Later in next phase, we need to take care of cases when amount deductible terms are defined as % of Limit and also when these % type deductibles are to be added for calculating combined terms