



Consumer Data Standards Decision Proposal 338 Feedback

Stay or Go Pty Ltd

15 December 2024

1 General Feedback

As a user of Product Reference Data (PRD), Stay or Go Pty Ltd (StayorGo) is supportive of [DP338](#) and the ongoing improvement of the data standards – we believe such improvements are essential to the success of the CDR. Current limitations and ambiguity within the standards are hindering the opportunity to apply PRD to use cases, such as product comparison, and creating risk that consumers are misled. Uplifting the standards will lead to improved data quality, a stated priority for the CDR, and enable others to begin applying PRD to the advantage of consumers and the industry.

We recommend phasing the implementation of DP338/DP306 in to two parts:

- Phase 1 – those changes that are obvious improvements requiring little refinement to finalise.
- Phase 2 – the set of changes where the standard still requires refinement and further consultation, such as the introduction of feeCategory.

2 Specific Feedback – Get Products

2.1 Addition of the cardOption object which may be considered an update to the existing cardArt object.

Feedback: We highly support this change as the new BankingProductCardOption object allows for a more structured representation of card options, which are currently either completely omitted from the data, or represented as features.

Suggested Phase: Phase 1

2.2 [NBL] Addition of the BUY_NOW_PAY_LATER query parameter and product category value.

Feedback: With the addition of ‘Buy Now, Pay Later’ products to the PRD, which do not fit under any of the existing product categories, this is a welcomed change.

Suggested Phase: Phase 1

3 Specific Feedback – Get Product Detail

3.1 Addition of the cardOption object (from the BankingProduct schema used by the Get Products endpoint) which may be considered an update of the existing cardArt object.

Feedback: As per [2.1](#), we highly support this change as the new BankingProductCardOption object allows for a more structured representation of card options, which are currently either completely omitted from the data, or represented as features.

Suggested Phase: Phase 1

3.2 *Change to the description of the features and constraints properties, and addition of new feature and constraint types.*

Feedback: We support the change to descriptions of the *features* and *constraints* properties. Regarding new and existing *featureTypes* and *constraintTypes*, we believe the below improvements could be made.

[BankingProductFeatureV3](#)

- The descriptions for the new MAX_BALANCE, MAX_LIMIT, MIN_BALANCE and MIN_LIMIT *featureType* values are currently a 'copy and paste' from *constraints*. These descriptions should be modified to ensure they are appropriate for the purpose. For example, MIN_LIMIT should include the potential use for transaction limits, rather than loan limits. The description should state that these are for operational limitations, as opposed to application limitations covered under *constraints*.

Suggested Phase: Phase 1

[BankingProductConstraintV2](#)

- It would be beneficial to define the purpose of each *constraintType* value more clearly, as there appears to be some confusion for data holders. Using RESIDENTIAL_MORTGAGES as an example; there are several data holders who are using MIN_BALANCE to represent the minimum loan amount, rather than MIN_LIMIT, and similarly, MAX_BALANCE to represent the maximum loan amount, rather than MAX_LIMIT. There are instances of this across different product categories.

Suggested Phase: Phase 1

- Many lending products have minimum and/or maximum loan terms that are advertised on provider websites. It would make sense to add two new *constraintType* values – MIN_TERM and MAX_TERM – so that data holders can disclose this information in a structured format. These would also be applicable to term deposits.

Suggested Phase: Phase 2

3.3 *Change to the fees schema, including a feeCategory and additional fee types.*

Feedback: We are very supportive of improvements to the fees schema, to improve the quality of structured data, however, we believe the proposed changes require further refinement and consultation.

[BankingProductFeeV2](#)

- *feeCategory* and *feeType*, as currently defined, are not distinct concepts, for example:
 - both contain TRANSACTION;

- *feeCategory* contains TELEGRAPHIC_TRANSFER (which is a type of transaction), while *feeType* contains PURCHASE, PAYMENT, and CASH_ADVANCE (which are also types of transaction);
- *feeCategory* includes channels (such as ATM, BRANCH), but also includes other concepts such as TRANSACTION, CLOSURE.
- Further, they do not allow for common fees to be unambiguously defined. For example, when defining common credit card fees, it's often not clear which combination of *feeCategory* and *feeType* to use:

Common Credit Card Fee	Possible feeCategory	Possible feeType
Account Fee	SERVICE, CARD	PERIODIC
Reward Program Fee	SERVICE, THIRD_PARTY	PERIODIC
Additional Cardholder Fee	SERVICE, CARD	PERIODIC
Foreign Exchange Fee	FOREIGN_EXCHANGE	PAYMENT, TRANSACTION, PURCHASE, EVENT
Late Payment Fee	SERVICE, CARD, OTHER	PAYMENT_LATE
Over Limit Fee	SERVICE, CARD, OTHER	TRANSACTION, PURCHASE, EVENT, OTHER
Dishonour Fee	TRANSACTION	DISHONOUR
Cash Advance Fee	ATM, CARD, TRANSACTION	CASH_ADVANCE

We recommend amending the proposal to a set of distinct fee properties that allow fees to be accurately, unambiguously, and consistently categorised. For example:

Property / Value	Description
feeType	<i>Mandatory</i>
UPFRONT	A fee charged as part of the account origination process
PERIODIC	A fee charged on a periodic basis
EVENT	A fee charged when a specific event occurs after account origination and prior to account closure
EXIT	A fee charged as part of the account closure process
feeCategory (or feeEvent)	<i>Mandatory</i>
<i>Transaction Events</i>	
CASH_ADVANCE	A cash advance from a line of credit, such as a credit card
CHEQUE	A withdraw was made using a cheque
DEPOSIT	A deposit (or payment) into the account
PAYMENT	A payment made from the account, for example, to another account or via BPAY
PURCHASE	An online or POS purchase made from the account
TELEGRAPHIC_TRANSFER	A telegraphic transfer made from the account
WITHDRAWAL	A withdrawal from the account, for example, at an ATM or branch
OTHER_TRANSACTION	Another type of transaction
<i>Exception Events</i>	
HONOUR	A payment from the account is honour, despite the account being overdrawn
DISHONOUR	A payment from the account is dishonoured due to insufficient funds
PAYMENT_LATE	A loan repayment is not received by the due date
OVERLIMIT	A loan is overdrawn beyond its credit limit
OTHER_EXCEPTION	Another type of exception
<i>Service Events</i>	

ENQUIRY	An account enquiry
REPLACEMENT	A physical account feature, such as a card, is replaced
CORRESPONDENCE	Correspondence about the account
VARIATION	A variation to the account, such as a limit increase
VALUATION	An asset, such as a residential property, is valued
OTHER_SERVICE	Another type of service
<i>Upfront Event</i>	
APPLICATION	An account application is processed
<i>Exit Event</i>	
EARLY_TERMINATION	The account is closed before the term has ended, for example a loan or a term deposit
<i>Periodic Event</i>	
ACCOUNT	Account service
CARD	A card service, such as provision of an optional card, incremental to the account service
ADDITIONAL_CARD	Additional card service
REWARD_PROGRAM	Reward program membership
channel	The channel through which the fee was incurred
ATM	An ATM was used
BRANCH	A branch was used
POS	A POS terminal was used
ONLINE	An online channel was used, such as internet banking
TELEPHONE_BANKING	Telephone Banking was used

The above example should not be considered as our final recommendation, but rather an example of how a clearer, unambiguous standard could be developed.

Before finalising the proposed change, we recommend mapping all common fees to the new standard to ensure they can be clearly and unambiguously represented. StayorGo would be happy to support the DSB with this exercise.

Further to the above, we recommend the following:

- The standard should define a set of valid *feeType* / *feeCategory* value combinations.
- Ambiguity between *feeTypes* PURCHASE, TRANSACTION, EVENT, PAYMENT should be removed by introducing clear descriptions as to when each should be used (they are currently used interchangeably by data holders).
- Many fees have logic conditions, such as AND, OR, MIN and MAX, for example, Foreign Currency Exchange Fees on Credit and Debit Cards are often the maximum of a fixed fee (e.g. \$3) or a percentage of the transaction (e.g. 3%). Currently, *additionalInfo* is the only place such conditions can be included. The standard requires a solution that enables data holders to represent these logic conditions within the structured data.

Suggested Phase: Phase 2

[BankingProductDiscount](#)

- It would make sense for the structure of BankingProductDiscount objects to align with the newly structured BankingProductFeeV2 objects. For example:

```

{
  "description": "string",
  "discountType": "BALANCE",
  "discountMethodUType": "fixedAmount",
  "fixedAmount": {
    "amount": "string"
  },
  "rateBased": {
    "balanceRate": "string",
    "transactionRate": "string",
    "accruedRate": "string",
    "feeRate": "string"
  },
  "additionalValue": "string",
  "additionalInfo": "string",
  "additionalInfoUri": "string",
  "eligibility": [
    {
      "discountEligibilityType": "BUSINESS",
      "additionalValue": "string",
      "additionalInfo": "string",
      "additionalInfoUri": "string"
    }
  ]
}

```

Note the addition of the *discountMethodUType* property, and nested *fixedAmount* and *rateBased* properties.

Suggested Phase: Phase 1

3.4 [\[depositRates\] Removal of the BUNDLE_BONUS and INTRODUCTORY depositRateType values.](#)

Feedback: With the addition of the *adjustmentBundle* field, and the use of *additionalValue* to indicate the discount period, we agree that these are no longer needed.

Suggested Phase: Phase 1

3.5 [\[depositRates\] Addition of the adjustmentToBase and adjustmentBundle fields.](#)

Feedback: We are highly supportive of this change as it resolves ambiguity around which base rate a given adjustment rate should be applied to.

Suggested Phase: Phase 1

3.6 [\[depositRates\] Addition of the applicationType field to support the applicationFrequency field.](#)

Feedback: We support this change and believe it is a robust solution that will greatly improve the quality of the PRD, particularly in relation to rates that are applied at maturity, for which there are currently multiple different implementations in the data.

Suggested Phase: Phase 1

3.7 *[depositRates] Addition of the applicabilityConditions field, including inside the tiers object.*

Feedback: We are very supportive of the addition of the *applicabilityConditions* field to BankingProductDepositRateV2 and BankingProductLendingRateV3 objects. Currently this information is disclosed in unstructured strings, which makes it difficult for data recipients to identify whether a given rate is available to a consumer. The addition of *applicabilityConditions* have, however, raised a couple of concerns which will require further consultation and refinements.

BankingProductRateTierV4

- With the addition of *applicabilityConditions* within the BankingProductDepositRateV2 and BankingProductLendingRateV3 objects, it seems redundant to also have *applicabilityConditions* within each BankingProductRateTierV4 object. Could you provide a use case where *applicabilityConditions* would need to be defined at a *tier* level? If there is no obvious use case, we'd recommend removing this option.

Suggested Phase: Phase 1

BankingProductRateConditionV2

- The standard should clearly state whether applicability conditions are conjunctive or disjunctive.
 - In line with eligibility conditions, which are “exclusive and additive”, our assumption is that applicability conditions would also be additive (i.e., conjunctive), however, this needs to be clearly defined.
 - Alternatively, the addition of a new property (e.g., *applicabilityConditionsMethod*), with a value of ‘AND’ or ‘OR’, would allow for data holders to define the relationship between applicability conditions themselves.

Suggested Phase: Phase 1 for the former; Phase 2 for the latter.

3.8 *[depositRates] Clarification that PERCENT unitOfMeasure values should be specified using the RateString format.*

Feedback: We support this change; the absence of a defined format for PERCENT *unitOfMeasure* values in the current standards is resulting in different interpretations and implementations by data holders.

Suggested Phase: Phase 1

3.9 *[lendingRates] Removal of the BUNDLE_DISCOUNT_FIXED, BUNDLE_DISCOUNT_VARIABLE and INTRODUCTORY lendingRateType values.*

Feedback: As per [3.4](#), with the addition of the *adjustmentBundle* field, and the use of *additionalValue* to indicate the discount period, we agree that these are no longer needed.

Suggested Phase: Phase 1

3.10 *[lendingRates] Addition of the BALANCE_TRANSFER and FEE lendingRateType values.*

Feedback: We support this change, particularly the addition of the BALANCE_TRANSFER *lendingRateType* value, as some data holders are using the INTRODUCTORY *lendingRateType* to represent balance transfer rates for credit cards, and others are providing the information under *features*.

Suggested Phase: Phase 1

3.11 *[lendingRates] Addition of the referenceRate field.*

Feedback: We support this change, however, believe it should be made clear that the *referenceRate* cannot be used in place of a BankingProductLendingRateV3 object to define the index rate. In other words, there should always be a lending rate object with a *rate* that matches the *referenceRate*.

Suggested Phase: Phase 1

3.12 *[lendingRates] Clarification of the use of the comparisonRate field.*

Feedback: Within the current set of adjustment rates published in the PRD, there are instances of data holders providing full comparison rates, adjustment comparison rates, and some are omitting the comparison rate altogether (which is a separate issue). We welcome this change as it clearly defines one implementation which is expected of data holders.

Suggested Phase: Phase 1

3.13 *[lendingRates] Addition of the revertRate and revertProductId fields.*

Feedback: We are supportive of requiring data holders to disclose, in a structured format, which rate a given rate reverts to (if applicable). This information is required for more complex product comparisons and will ultimately improve the quality of the PRD. The addition of the *revertProductId* property is a welcomed change, however, we have the following concern about the *revertRate* property.

[BankingProductLendingRateV3](#)

- Although it is unlikely multiple BankingProductLendingRateV3 objects with different parameters will have identical rates, it is possible. In this case, the *revertRate* alone may not be enough to uniquely identify which rate a given rate reverts to. Consider the DISCOUNT rate in the example below; after the 2-year discount period is up, it is not clear from the structured data alone whether it reverts to the FIXED rate for the remaining year, before reverting to the VARIABLE rate, or if it reverts straight to the VARIABLE rate.


```

[
  {
    "lendingRateType": "VARIABLE",
    "rate": "0.05",
    "repaymentType": "PRINCIPAL_AND_INTEREST",
    "loanPurpose": "OWNER_OCCUPIED"
  },
  {
    "lendingRateType": "FIXED",
    "rate": "0.05",
    "revertRate": "0.05",
    "repaymentType": "PRINCIPAL_AND_INTEREST",
    "loanPurpose": "OWNER_OCCUPIED",
    "additionalValue": "P3Y"
  },
  {
    "lendingRateType": "DISCOUNT",
    "rate": "0.01",
    "revertRate": "0.05",
    "adjustmentToBase": "FIXED",
    "repaymentType": "PRINCIPAL_AND_INTEREST",
    "loanPurpose": "OWNER_OCCUPIED",
    "additionalValue": "P2Y"
  }
]

```

A potential solution for this would be to add a *rateId* property to *BankingProductLendingRateV3*, which both *revertRate* and *referenceRate* can reference.

Suggested Phase: Phase 2

3.14 *[lendingRates] Addition of the adjustmentToBase and adjustmentBundle fields.*

Feedback: As per [3.5](#), we are highly supportive of this change as it resolves ambiguity around which base rate a given adjustment rate should be applied to.

Suggested Phase: Phase 1

3.15 *[lendingRates] Addition of the applicationType field to support the applicationFrequency field.*

Feedback: As per [3.6](#), we support this change and believe it is a robust solution that will greatly improve the quality of the PRD, particularly in relation to rates that are applied at maturity, for which there are currently multiple different implementations in the data.

Suggested Phase: Phase 1

3.16 *[lendingRates] Addition of the applicabilityConditions field, including inside the tiers object.*

Feedback: Please see [3.7](#).

Suggested Phase: Phase 1

3.17 *[lendingRates] Clarification that PERCENT unitOfMeasure values should be specified using the RateString format.*

Feedback: As per [3.8](#), we support this change; the absence of a defined format for PERCENT *unitOfMeasure* values in the current standards is resulting in different interpretations and implementations by data holders.

Suggested Phase: Phase 1

3.18 *[lendingRates] Addition of the PRINCIPAL_AND_FEE repayment Type.*

Feedback: We agree this is a necessary addition to compliment the addition of the FEE *lendingRate* type.

Suggested Phase: Phase 1

3.19 *Change to the use of the additionalValue fields associated with the PENSION_RECIPIENT and STUDENT eligibility types.*

Feedback: We support this change as it will facilitate disclosure of more granular product eligibility conditions.

Suggested Phase: Phase 1

3.20 *[NBL] Addition of the BUY_NOW_PAY_LATER query parameter and product category value.*

Feedback: As per [2.2](#), with the addition of 'Buy Now, Pay Later' products to the PRD, which do not fit under any of the existing product categories, this is a welcomed change.

Suggested Phase: Phase 1

3.21 *[NBL] Addition of the instalments object to support 'Buy Now, Pay Later' (BNPL) products.*

Feedback: We believe this is a necessary addition to support BNPL products.

Suggested Phase: Phase 1

4 Other Feedback – Get Product Detail

[BankingFeeRate](#)

- The language “One of *balanceRate*, *transactionRate* and *accruedRate* is mandatory” can be interpreted as either “At least one of...” or “One and only one of...”. It would be beneficial to clarify the correct interpretation within the relevant property descriptions.

This applies to the *amount*, *balanceRate*, *transactionRate*, *accruedRate* and *feeRate* properties within *BankingProductDiscount* as well.

Suggested Phase: Phase 1

[BankingProductLendingRateV3](#)

- When the *lendingRateType* is one of the adjustment rate types, the use of the *additionalValue* field is defined as “The period of time for the (discounted | penalty) rate if applicable”. The standard should clarify how discounts or penalties that are applicable for the life of the loan are expected to be represented.

Suggested Phase: Phase 2

- Given a *BankingProductLendingRateV3* object with an *INTEREST_ONLY* *repaymentType*, where should the available interest only terms be defined? This information is usually available on provider websites but is largely absent from the product reference data. Note that the available interest only terms are sometimes discontinuous (e.g., 1-5, 7 or 10 years). This issue also applies to *repaymentType* = null, which indicates the rate is applicable to all repayment types.

Suggested Phase: Phase 2, assuming industry consultation may be required.

- Ideally the standard would disallow *repaymentType* = null. This would eliminate the issue where data holders only disclose *repaymentType* in *additionalInformation*. If the rate applies to both IO and P&I, then the data holder should include two rate objects. The same applies for *loanPurpose*.

Suggested Phase: Phase 2