

Data Standards Body

Technical Working Group

Decision 154 – Enhanced Error Handling

Contact: Mark Verstege

Publish Date: 27th May 2021

Decision Approved By Chairman: 1st June 2021

Context

This document summarises the changes to be made resulting from consultation across the series of Enhanced Error Handling. Most recently, three Decision Proposals were published for community consultation and feedback. They were:

- [Decision Proposal 154 – Enhanced Error Handling: Error Code Payload Structure and Transition Arrangements](#)
- [Decision Proposal 155 - Enhanced Error Handling: Error Code Catalogue](#)
- [Decision Proposal 156 - Enhanced Error Handling: Custom Error Code Discovery Service](#)

This document presents the set of changes to support enhanced error handling incorporating feedback received from community consultation.

Background

Well-defined error handling is critical to the successful interoperability of Accredited Data Recipients (ADRs), Data Holders (DHs) and the CDR Register. As the Consumer Data Right (CDR) ecosystem grows — soon to accommodate all Accredited Deposit-taking Institutions (ADIs) — the complexity of integrating across a many-to-many system becomes more challenging. Without a well-defined set of standardised error responses, ADRs are forced to build out conditional logic to process each DH differently, possibly inconsistently. This is a non-trivial effort for ADRs that is difficult to scale.

Without standards to provide deterministic error handling, this will lead to inconsistent handling of errors by ADRs and a poorer consumer experience that may not only be confusing but result in interoperability issues that impacts consumer experience.

To provide predictable and scalable error handling across the ecosystem, this decision proposal proposes payload changes to support a standardised error catalogue that participants can reliably and consistently implement to.

This decision document assumes prior knowledge of the previous consultations and decision proposal series.

Consultation conducted

As a problem space, Enhanced Error Handling has been consulted on over a series of Decision Proposals and workshops to agree on the structural requirements. This proposal consolidates technical feedback across the following decision proposals:

- [Decision Proposal 154 – Enhanced Error Handling: Error Code Payload Structure and Transition Arrangements](#)
- [Decision Proposal 155 - Enhanced Error Handling: Error Code Catalogue](#)
- [Decision Proposal 156 - Enhanced Error Handling: Custom Error Code Discovery Service](#)
- [Decision Proposal 119 - Enhanced Error Handling Payload Conventions](#)

- [Decision Proposal 120 - CDR Error Codes for Enhanced Error Handling](#)
- [Decision Proposal 121 - Application of existing HTTP Error Response Codes to Enhanced Error Handling](#)
- [Decision Proposal 122 - Extension of Supported HTTP Response Codes for Enhanced Error Handling](#)
- [Decision Proposal 127 - CX Guidelines for Enhanced Error Handling](#)

Along with feedback provided in consultation workshops. Outputs are available here:

- **Workshop 1 – Payload Data Structure:** https://miro.com/app/board/o9J_kr5jpHQ=
- **Workshop 2 – Error Catalogue:** https://miro.com/app/board/o9J_krfCaQQ=
- **Workshop 3 – Consumer Experience:** https://miro.com/app/board/o9J_knxw6Pc= (input) and https://miro.com/app/board/o9J_khmXlzU= (findings)

Decision To Be Made

- Define the structural data standards required to support error handling and any versioning necessary
- Determine the list of standardised error codes
- Define the transition path towards standardised CDR error codes for the CDR Register, Data Recipients and Data Holders

Feedback Provided

The following feedback is a summary of responses to Decision Proposals 154, 155 and 156.

Error code structure

1. Community feedback proposed a change to the structure to allow custom error codes to be provided as the top-level code string. Where the custom error code extends a standard error code, the standard URN code is to be supplied in the meta object. This allows for no breaking change to existing ADR implementations and lowers the implementation effort for Data Holders. This change is adopted in this document.

Transition arrangements

2. Data Recipients sought an extended transition period to ensure errors could be discovered without adversely impacting ADR software.
3. Data Holders requested that the mandatory transition arrangements commence *after* November 2021 due to the implementation work currently in scope for November 2021

Error codes

4. Some Data Holders requested that error codes not be made mandatory because they are currently intercepted high up their application stack (e.g., WAF or API Gateway level)
5. Feedback indicated that Invalid resource vs Unavailable resource error conditions should be explicitly separate to indicate permanent unavailability of a requested resource vs temporary unavailability of a requested resource
6. Suggestion to optimise the error code for invalid CDR Arrangement ID
7. Feedback supported consolidation of invalid ADR status error codes
8. Feedback supported better alignment of CDR Register error codes to the HTTP status codes used for Data Holder error codes
9. Feedback was received on whether partial successful responses should be supported, however no solutions were proposed. The DSB recommended that the community should raise this as a separate change request for consideration.

10. Feedback was received on how to handle resources that move location (e.g. the ID for a bank account changes due to a change in ID Permanence algorithm), however no solutions were proposed. The DSB recommended that the community should raise this as a separate change request for consideration.
11. Support for 404 Not Found where resources are unavailable, invalid or not found was supported
12. Standard error codes apply to Data Recipients, Data Holders and the CDR Register as applicable.

Versioning

1. Feedback was inconclusive on versioning error codes. No option was supported, and no versioning of error code responses was adopted.
2. Feedback indicated that versioning would result in a higher than necessary implementation cost for data holders.

Discovery service

13. Data Holders were not supportive of formal standards for presenting custom error codes on their Developer Portals
14. Where Data Holders support custom error codes, feedback indicated that Data Holders were supportive of providing those details in a discoverable and developer friendly format based on their individual developer portal solutions and approaches

Implementation considerations

Implementation consideration has been given to how both ADRs and DHs can migrate to the target state.

Primarily, this has involved accounting for four main considerations:

1. Providing time for Data Holders to make changes to their existing implementations
2. A way for ADRs to determine the mapping of old and new error codes for each DH
3. Ensuring there is sufficient time for ADRs to cutover to handling the new error codes
4. A solution that is non-breaking for existing clients during the transition period especially where Data Holders may choose to implement standardised error codes ahead of mandatory obligation dates.

Feedback on implementation timeframes and the amount of change currently proposed for November 2021 was taken into consideration with revised obligation dates recommended as a result of Data Recipient and Data Holder feedback.

Further to this, care has been given to ensure the target state makes it easier for participants to adopt new error codes in future without change to the data model. It is envisaged that a periodic review process would consider adoption of any new standardised error codes identified by the community.

The aim has been to ensure any Data Holders entering the ecosystem can, if desired, adopt the new standardised error codes from July 2021. Where possible this reduces ongoing maintenance to affected DHs. It is recognised that this will not be possible for all current or future DHs and allowing sufficient time for the cutover will be required to ensure DHs can deliver existing implementation obligations whilst providing certainty to ADRs they can continue to handle errors in a reliable way.

Overall, consideration has been given to the extension of the standardised error catalogue where there is benefit in consistency to reduce ADR impact. The solution is industry-agnostic allowing it to apply to the sectors beyond banking.

Data Holders

Data Holders must update their software to correctly respond with standardised error codes within the obligation dates. These changes minimise impact to existing implementations by allowing all affected Data Holders to continue to support their application-specific error codes adjacent to the standardised error code they extend.

Consideration has also been given to the implementation feasibility of each error code because some error codes are more likely to be encountered higher up a Data Holder's application stack that has limited customisability.

Data Recipients

Primary consideration has been given to reducing impact to Data Recipients whilst significantly improving implementation consistency and reliability when connected to many Data Holders. Data Recipients will have no build impact during transition however they will require changes to their internal application logic to correctly handle standardised error codes in the absence of pre-existing application-specific error codes.

Based on community consultation, error codes that are reasonably expected to require consistent error handling have been made mandatory.

Public Clients

Any public clients collecting product reference data will be impacted. Similar to Data Recipients, any public clients must be updated to handle the new standardised error codes. The collection of product reference data by public clients is outside the scope of the decision proposal.

CDR Register

The CDR Register will be impacted when collecting metrics data from Data Holders.

As a single client, with a limited set of errors to handle when collecting metrics data, it is considered there is sufficient time to transition.

The CDR Register also has implementation impacts to the errors it produces to API requests from Data Recipients and Data Holders. This has been factored into the transition periods and commencement of mandatory obligations.

Recommendations

The following changes are recommended based on community feedback

Payload structure

1. Standardised error codes to be defined as a URN structure which denotes the sector, the error category and the error code specific to the application error encountered by the CDR Register, Data Recipients and Data Holders. See Appendix A.
2. Payload response structure is extended to support a conditional URN field.
3. Data Recipients and Data Holders can continue to support application-specific error codes provided the standard URN error code is provided in the conditional URN field, otherwise the standard error code must be provided.

Error code format

The payload structure is extended to support a conditional URN field as follows:

1. Update the description of the error code field to accommodate standard URN error codes and application-specific error codes
2. Extend the error response's "meta" object to include the conditional URN defined as *"The CDR error code URN which the application-specific error code extends. Mandatory if the error `code` is an application-specific error rather than a standardised error code."*

Response Codes

1. Add support for 404 Not Found to the response codes table.

Error Catalogue

1. Add the error catalogue of standard error codes to the standards including the list of mandatory error codes to each end point they apply to. See Appendix B for details.

Response Versioning

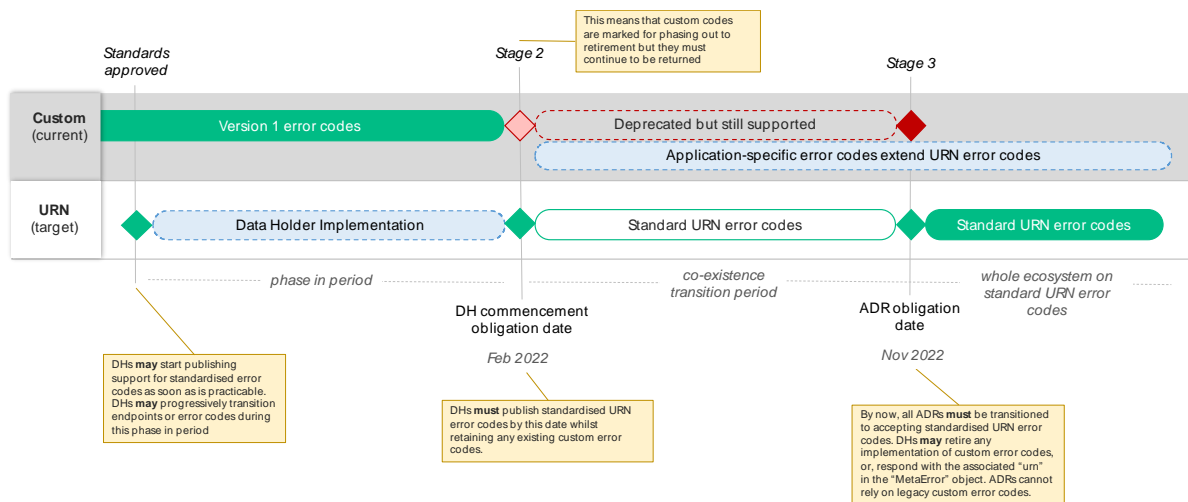
1. No versioning of error codes was supported and no response payload versioning, error code version negotiation or error code versioning changes are recommended.

Discovery Service

1. No standards-based discovery service is recommended. Data Recipients and Data Holders that support application-specific error codes must instead provide a developer friendly list of custom error codes and their mapping to standardised CDR error codes. It is recommended that the format is left to the discretion of the Data Recipient and Data Holder because it is dependent on the developer experience offered by the participant.

Transition arrangements

At a high level, the transition is proposed as follows:



It is recommended that:

1. Data Recipients and Data Holders must support standardised error codes commencing from February 1st 2022.
2. Data Recipients and Data Holders that support application-specific error codes prior to February 2022 must respond with both the standardised error code and the application-specific error code to assist with clients encountering errors to progressively discover and update their client software

3. Data Recipients and Data Holders that support application-specific error codes may retire any custom error codes in favour of standardised error codes from November 1st 2022.
4. Data Recipients and Data Holders may continue to support application-specific error codes provided they continue to respond with the standardised error code in addition to their application-specific error codes
5. Effective as soon as the Data Recipient or Data Holder responds with standardised error codes they must publish a mapping of those codes to the standard CDR error codes in a developer friendly way that is discoverable and freely available.
6. Data Recipients and Data Holders may implement support for standardised error codes at any time prior to February 1st 2022 because the change is non-breaking for existing clients.

Appendix A – Error code URN format

The following standards define the structure and format of the standardised URN error codes:

urn-string = "urn:" **NID** ":" **metatype** ":" **sub-type** ":" **error-category** "/" **error-code**

NID = "au-cds" string.

metatype = "error" string.

sub-type = cds-all / cds-register / cds-banking / cds-energy

cds-all = "cds-all" string. An error code common to all API responses,

cds-register = "cds-register" string. Reserved for CDR Register issued error codes only,

cds-banking = "cds-banking" string. An error code specific to the CDR banking APIs only,

cds-energy = "cds-energy" string. An error code specific to the CDR energy APIs only.

error-category = string. The high-level category code for the error defined in the CDR Error Catalogue

error-code = string. The specific error encountered, defined in the Consumer Data Standards

Appendix B – Error code catalogue

The following list of standardised error codes recommended:

Standard Error Codes

A list of standard error codes to help categorise an error response. The applicable HTTP response code is also given.

General Errors

Error Title	Error Code	HTTP Status Category	Description
Expected Error Encountered	urn:au-cds:error:cds-all:GeneralError/Expected	4xx	<p>An error code that can be used, when an expected error occurs that is otherwise not covered by a more specific error.</p> <p>The error description SHOULD be populated with a meaningful error description, without revealing sensitive information.</p> <p>An application specific error code MAY be provided. The MetaError » urn MUST be populated with the standard CDR error code.</p>
Unexpected Error Encountered	urn:au-cds:error:cds-all:GeneralError/Unexpected	5xx	<p>An error code that can be used, when an unexpected error occurs.</p> <p>The error description SHOULD be populated with a meaningful error description, without revealing sensitive information.</p> <p>An application specific error code MAY be provided. The MetaError » urn MUST be populated with the standard CDR error code.</p>

Error Title	Error Code	HTTP Status Category	Description
Service Unavailable	urn:au-cds:error:cds-all:Service/Unavailable	503 (Service Unavailable)	<p>A request is made but the API unavailable as due to an outage.</p> <p>The error description MAY describe whether the outage is scheduled or unexpected and whether it is fully unavailable or partially unavailable.</p>

400 Bad Request Errors

Error Title	Error Code	Description
Missing Required Field	urn:au-cds:error:cds-all:Field/Missing	<p>The request is missing a mandatory field required for the API. It may be a missing query parameter or missing field in the request payload. This error code can be used, where a more specific validation error is not applicable.</p> <p>The error description SHOULD be the parameter name of the missing field.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>
Missing Required Header	urn:au-cds:error:cds-all:Header/Missing	<p>A required HTTP header has not been provided.</p> <p>The error description SHOULD be the HTTP header name.</p> <p>This error code SHOULD be supported for unauthenticated and authenticated APIs.</p>

Error Title	Error Code	Description
Invalid Field	urn:au-cds:error:cds-all:Field/Invalid	<p>Applies when the value of the URL parameter or request body parameter is an invalid type or the value violates the field's constraints as defined by the interface contract. For example, <code>is-owned</code> is a <code>Boolean</code> but a <code>DateStringValue</code> is provided.</p> <p>The error description SHOULD be the parameter name of the invalid field. The error description MAY include further details explaining the valid format.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>
Invalid Header	urn:au-cds:error:cds-all:Header/Invalid	<p>Applies when a HTTP Header is provided but the value provided is an invalid type or violates the field type constraints as defined in the Consumer Data Standards.</p> <p>The error description SHOULD be the HTTP header name. The error description MAY include further details explaining the valid format.</p> <p>This error code SHOULD be supported for unauthenticated and authenticated APIs.</p>
Invalid Date	urn:au-cds:error:cds-all:Field/InvalidDateTime	<p>An invalid date is provided. For example, a future date value is expected, but a date in past or current date is supplied. Applies to <code>DateTimeString</code>, <code>DateString</code>, and <code>TimeString</code> field types.</p> <p>The error description SHOULD be the parameter name of the invalid date field. The error description MAY include further details explaining the expected date value.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>

Error Title	Error Code	Description
Invalid Page Size	urn:au-cds:error:cds-all:Field/InvalidPageSize	<p>The value provided in the <code>page-size</code> pagination field is greater than the maximum allowed by the Consumer Data Standards (<code>page_size > 1000</code>).</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>
Invalid Version	urn:au-cds:error:cds-all:Header/InvalidVersion	<p>A request is made for a version that is not a <code>PositiveInteger</code>. For example:</p> <ul style="list-style-type: none"> • <code>x-min-v</code>, <code>x-v</code> or <code>x-<HID>-v</code> are not <code>Integers</code>(e.g. <code>x-min-v=foo</code>, <code>x-v=bar</code>, <code>x-ACME-v=cheese</code>) • <code>x-min-v</code>, <code>x-v</code> or <code>x-<HID>-v</code> are not positive-value integers (they are an <code>Integer</code> but <code><= 0</code>) <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p> <p>If the version header is a <code>PositiveInteger</code> but is not a version supported by the Data Holder, the Unsupported Version code applies.</p>

403 (Forbidden) Errors

Error Title	Error Code	Description
ADR Status Is Not Active	urn:au-cds:error:cds-all:Authorisation/AdrStatusNotActive	<p>The ADR or the ADR's software product is not "active".</p> <p>The error description SHOULD contain the current status of the ADR software product.</p>

Error Title	Error Code	Description
Consent Is Revoked	urn:au-cds:error:cds-all:Authorisation/RevokedConsent	<p>The consumer's consent is no longer authorised (for example revoked or expired) and the requested resource will not be provided.</p> <p>This error code SHOULD be supported for authenticated APIs.</p>
Consent Is Invalid	urn:au-cds:error:cds-all:Authorisation/InvalidConsent	<p>The authorised consumer's consent is not associated to the resource requested, is insufficient to execute the resource or is in a status that prevents the resource being executed.</p> <p>For example, if consent is awaiting authorisation of a secondary account holder.</p> <p>The error description SHOULD be a description of the status of consent without revealing sensitive information.</p> <p>This error code SHOULD be supported for authenticated APIs.</p>

404 (Not Found) Errors

Error Title	Error Code	Description
Resource Not Implemented	urn:au-cds:error:cds-all:Resource/NotImplemented	<p>The requested resource URL is a valid API endpoint defined by the Consumer Data Standards, but it is not implemented or not currently supported.</p> <p>This error code SHOULD be supported for unimplemented APIs.</p>

Error Title	Error Code	Description
Resource Not Found	urn:au-cds:error:cds-all:Resource/NotFound	The requested resource URL is not an API endpoint defined by the Consumer Data Standards and it is not a URL recognised by the Data Holder or Data Recipient.
Invalid Resource	urn:au-cds:error:cds-all:Resource/Invalid	<p>The requested resource identifier is permanently unavailable. No subsequent request for the resource will be successful. Applies when the resource ID is provided in the URI.</p> <p>The error description is the resource ID of the resource being requested.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>
Unavailable Resource	urn:au-cds:error:cds-all:Resource/Unavailable	<p>The requested resource identifier is temporarily unavailable. Subsequent requests for the resource may be successful. Applies when the resource ID is provided in the URI.</p> <p>The error description is the resource ID of the resource being requested.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>
Invalid Banking Account	urn:au-cds:error:cds-banking:Authorisation/InvalidBankingAccount	The requested bank account is permanently unavailable. No subsequent request for the account will be successful. Applies when the account ID is provided in the URI.

Error Title	Error Code	Description
		<p>The error description is the account ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>
Unavailable Banking Account	urn:au-cds:error:cds-banking:Authorisation/UnavailableBankingAccount	<p>The requested bank account is temporarily unavailable. Subsequent requests for the account may be successful. Applies when the account ID is provided in the URI.</p> <p>The error description is the account ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>

406 (Not Acceptable) Errors

Error Title	Error Code	Description
Unsupported Version	urn:au-cds:error:cds-all:Header/UnsupportedVersion	<p>A request is made for a version that is lower than the minimum version or greater than maximum version the Data Holder supports for the requested endpoint.</p> <p>The error description MAY include the minimum and maximum versions the Data Holder supports.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>

422 (Unprocessable Entity) Errors

Error Title	Error Code	Description
Invalid Resource	urn:au-cds:error:cds-all:Resource/Invalid	<p>The requested resource identifier is permanently unavailable. No subsequent request for the resource will be successful. Applies when the resource ID is provided in the request body.</p> <p>The error description is the resource ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>
Unavailable Resource	urn:au-cds:error:cds-all:Resource/Unavailable	<p>The requested resource identifier is temporarily unavailable. Subsequent requests for the resource may be successful. Applies when the resource ID is provided in the URI.</p> <p>The error description is the resource ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>
Invalid Banking Account	urn:au-cds:error:cds-banking:Authorisation/InvalidBankingAccount	<p>The requested bank account is permanently unavailable. No subsequent request for the account will be successful. Applies when the account ID is provided in the URI.</p> <p>The error description is the account ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>

Error Title	Error Code	Description
Unavailable Banking Account	urn:au-cds:error:cds-banking:Authorisation/UnavailableBankingAccount	<p>The requested bank account is temporarily unavailable. Subsequent requests for the account may be successful. Applies when the account ID is provided in the URI.</p> <p>The error description is the account ID of the resource being requested.</p> <p>This error code MUST be supported for authenticated APIs.</p>
Invalid Consent Arrangement	urn:au-cds:error:cds-all:Authorisation/InvalidArrangement	<p>The arrangement being executed has previously been revoked and no further action will be taken.</p> <p>The error description is the CDR Arrangement ID of the being executed.</p> <p>This error code MUST be supported for the Data Recipient and Data Holder CDR Arrangement Revocation endpoint.</p>
Invalid Page	urn:au-cds:error:cds-all:Field/InvalidPage	<p>The page being requested it out of of range. For example, the valid pagination range is 5 pages and the client requested page=10).</p> <p>The error description SHOULD be the maximum number of pages that are available.</p> <p>This error code MUST be supported for unauthenticated and authenticated APIs.</p>

CDR Register Errors

The following error codes apply to responses from the CDR Register. Data Recipient and Data Holder clients requesting data from the CDR Register may expect the following standard CDR error codes to be encountered:

Error Title	Error Code	HTTP Status Category	Description
Invalid Brand	urn:au-cds:error:cds-register:Field/InvalidBrand	404 (Not Found)	The brand provided to get the Data Recipient software statement assertion is invalid. Applies to the dataRecipientBrandId path parameter for CDR Register APIs.
Invalid Industry	urn:au-cds:error:cds-register:Field/InvalidIndustry	404 (Not Found)	The industry requested in the path to get Data Recipient or Data Holder metadata is invalid / does not exist and cannot be found. Applies to the industry path parameter for CDR Register APIs.
Invalid Software Product	urn:au-cds:error:cds-register:Field/InvalidSoftwareProduct	404 (Not Found)	The software product requested to get the Data Recipient software statement assertion is invalid or cannot be found. Applies to the softwareProductId path parameter.