

Get Involved: Participate in Improving the LIXI Standard

John Matthews
LIXI

Why Participate?

- The LIXI standard is very comprehensive
- The master schema contains 700 elements and 3000+ attributes
- However, the standard does not have everything
- New schema items are added at the rate of one request per day
- All the discussion for new schema items happens in LIXI Lab
- To be involved in the decision making you need to join LIXI Lab



standards.lixi.org.au

- LIXI Lab is the code repository where the LIXI schemas are kept
- From LIXI Lab you can download the latest schemas
- You can request improvements to the LIXI2 standards
- And participate in LIXI Lab issue ticket discussions

LIXI2 / Transaction Master



Management of the LIXI Master schema from which all LIXI2 Transaction Schemas are derived.







Projects

Your projects Starred projects Explore projects

Filter by name... Last updated

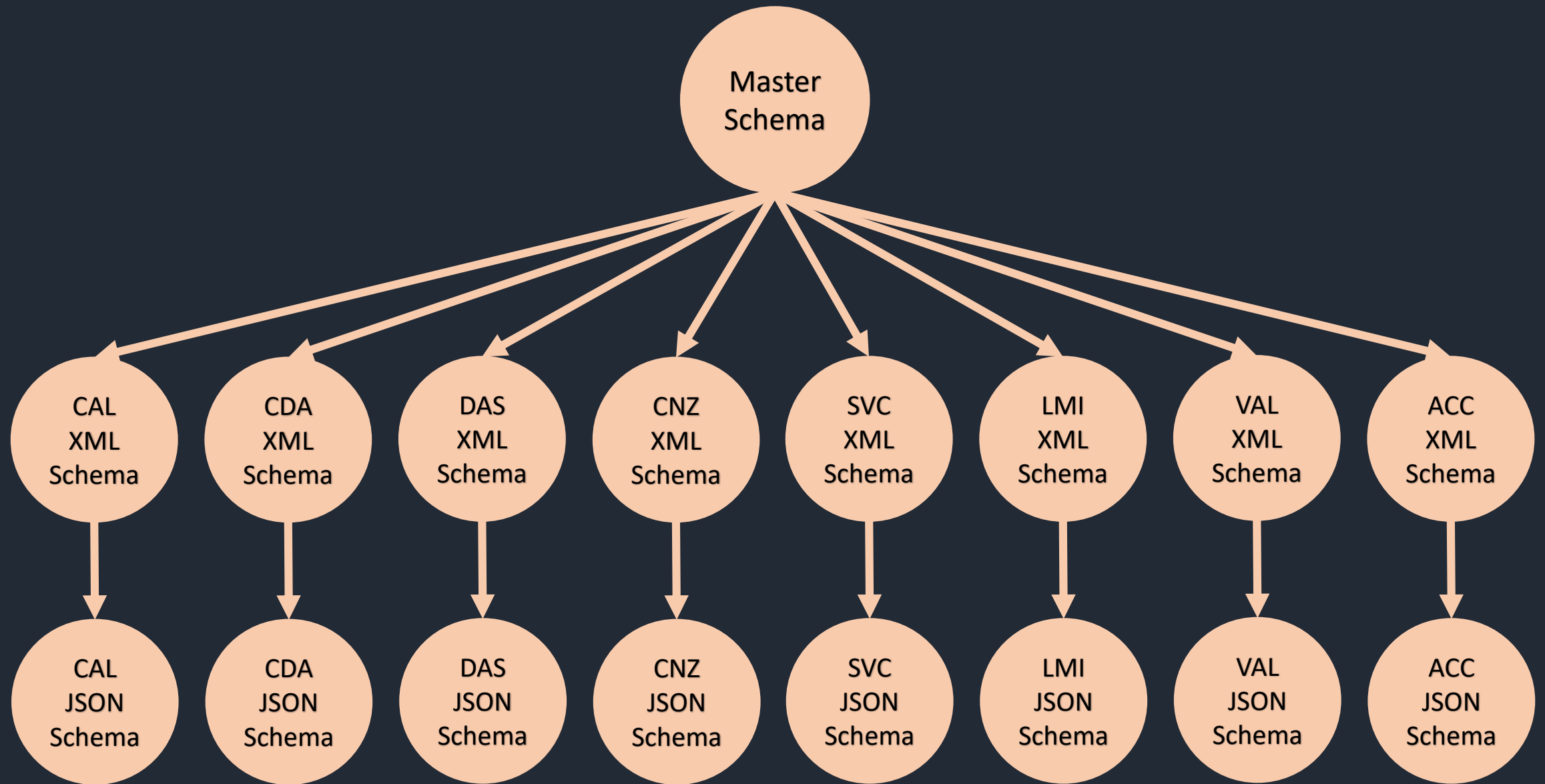
All Personal

-  **LIXI2 / TransactionMaster** Developer
Management of the LIXI Master Schema from which all LIXI 2 Transaction Schema are derived. updated 8 minutes ago
-  **LIXI2 / schematron-demo** Developer
A project to investigate schematron for a base level of business rules. updated 2 hours ago
-  **LIXI2 / API-Template** Developer
A Project containing template files to setup an API that will accept and process LIXI messages using GitLab and AWS updated 1 week ago
-  **LIXI2 / CustomisationByRestriction** Developer
Project to demonstrate Customisation By Restriction updated 1 month ago

The Master Schema



- At the center of all the LIXI processes is the LIXI Master Schema
- The master schema contains all the components of all LIXI the standards
- Changes to the LIXI standard are made to the master schema
- From this single file all other LIXI schemas are automatically derived
- Every time a change is made to the master schema a build is triggered
- This build produces the transaction schemas in both XML and JSON



LIXI Master Schema

```
<xs:element name="PhoneNumber" type="phoneType" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>A phone number for the publisher</xs:documentation>
    <xs:appinfo>
      <lx:path>Package.Publisher.PhoneNumber</lx:path>
      <lx:label>Phone Number</lx:label>
      <li:transactions>CAL,CDA,DAS,CNZ,SVC,LMI,VAL,ACC</li:transactions>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```



An item is included in a transaction schema if it is tagged here in the master schema transactions element

LIXI Transaction Schemas



```
<xs:element name="PhoneNumber" type="phoneType" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>A phone number for the publisher</xs:documentation>
    <xs:appinfo>
      <lx:path>Package.Publisher.PhoneNumber</lx:path>
      <lx:label>Phone Number</lx:label>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

The item is included in the derived transaction schema, identical but without the transaction element

LIXI Transaction JSON Schemas



```
"PhoneNumber": {  
  "$ref": "#/definitions/phoneType",  
  "description": "A phone number for the publisher",  
  "path": "Package.Publisher.PhoneNumber"  
}
```

JSON schemas are derived from each XML transaction schema with the same data structure, names, and descriptions

Why does the Master Schema matter?



When a change is proposed to the LIXI master schema, users of all the LIXI standards can provide comment/feedback on the proposal.

Regardless of whether an organisation has implemented LIXI using XML or JSON, or whether they are using CAL, CDA, DAS, CNZ, SVC, etc.

Organisations from all along the lending chain can join the discussion to ensure that the solution we find is a standardised solution.

Implementing a Change – The Build Process

✓ documentation



Create XML Schemas

✓ in discussion



Create JSON Schemas

✓ schemas (json)



Create HTML Documentation

✓ schemas (xml)



Create Draft Schema of “In Discussion” items

✓ tests (python)



Tests: Structural tests written in Python

✓ tests (schematr...)



Tests: Validate against Schematron rules

✓ tests (tickets)



Tests: Test schema against LIXILab tickets

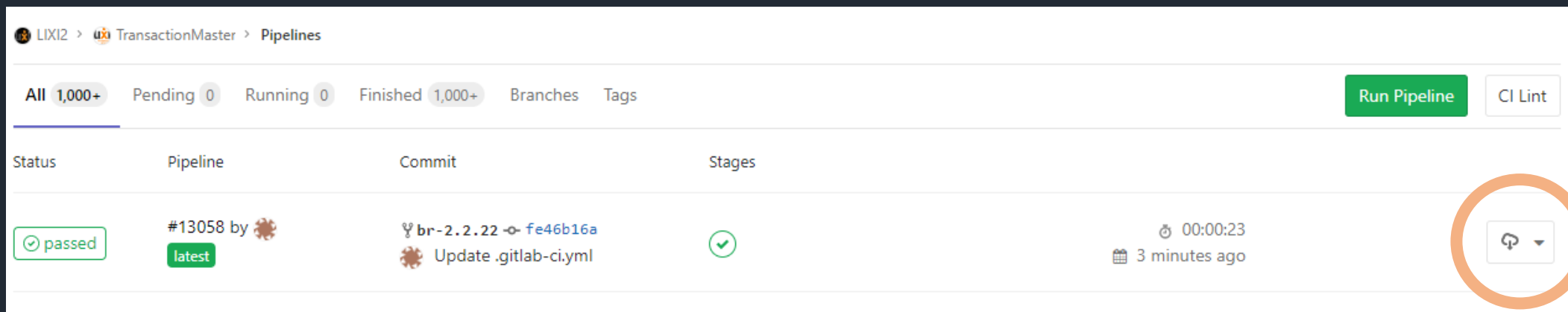
✓ tests (xsd)



Tests: Validate against XSD “master schema schema”

Transaction Master Build Artifacts

- Draft Master Schema
- HTML schema documentation
- XML transaction schemas (annotated & unannotated)
- JSON transaction schemas (annotated & unannotated)



The screenshot shows the GitLab Pipelines interface for the 'TransactionMaster' project. The top navigation bar includes the project name and a 'Pipelines' tab. Below the navigation, there are filters for pipeline status: 'All' (1,000+), 'Pending' (0), 'Running' (0), and 'Finished' (1,000+). There are also buttons for 'Run Pipeline' and 'CI Lint'. The main content area is a table of pipeline runs. The first row shows a pipeline that has passed, with a status of 'passed' in a green box. The pipeline is identified by the number '#13058' and the commit 'br-2.2.22' (commit hash 'fe46b16a'). The pipeline was triggered by the action 'Update .gitlab-ci.yml'. The pipeline duration is '00:00:23' and it was completed '3 minutes ago'. A dropdown menu icon is visible in the bottom right corner of the pipeline row, circled in orange with an arrow pointing to it.

Status	Pipeline	Commit	Stages	Duration	Time
passed	#13058 by latest	br-2.2.22 (fe46b16a) Update .gitlab-ci.yml	✓	00:00:23	3 minutes ago

How to Open a LIXI Lab Ticket



The screenshot shows the LIXI TransactionMaster web interface. The top navigation bar includes 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. A search bar is present with the text 'Search or jump to...'. The left sidebar shows the 'TransactionMaster' project with a menu containing 'Project', 'Repository', 'Issues' (circled in orange with a count of 71), 'List', 'Board', 'Labels', 'Milestones', 'Merge Requests' (0), and 'CI / CD'. The main content area is titled 'LIXI2 > TransactionMaster > Issues'. It features a summary bar with 'Open 71', 'Closed 1,003', and 'All 1,074'. A search bar and a 'Created date' filter are also visible. The 'New issue' button in the top right is circled in orange. Below the search bar, three issue cards are listed:

- Need placeholder for capturing the Net Rental Income for multiple Applicants owning the RealEstateAsset**
#1095 · opened 3 days ago by Harpreet Brar · updated 3 days ago
- Deprecate the 'Child Maintenance' Other Commitments category**
#1094 · opened 1 week ago by LIXI GitLab Site Admin (Shane Rigby) · LIXI Master Schema 2.2.22 · **Deprecated** **Implemented** Annotation CNZ · SVC LMI DAS CDA CAL LivingExpenses · updated 4 days ago
- Create new category of Other Commitments for 'Child and Spousal Maintenance' to replace 'Child Maintenance'**
#1093 · opened 1 week ago by LIXI GitLab Site Admin (Shane Rigby) · LIXI Master Schema 2.2.22 · **Implemented** Enhancement SVC LMI DAS · CNZ CDA CAL LivingExpenses · updated 1 week ago

Example LIXI Lab Ticket

When you first open a ticket in LIXI Lab you are only required to complete the “Problem / Requirement Statement”



'Authoriser' that has authority to authorise the proposed repayment

Problem / Requirement Statement

There is a need to capture whether the applicant has the authority and authorised the repayment from the nominated account.

Solution

Overview

There will be an element called "Authoriser" to capture all owners of an account and whether they authorised the financial institution for repayment.

Detail

Add items:

- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.Authorised
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.AuthorityVerified
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.x_Party

Definitions

Name	Definition
Authoriser	A party that has authorised the proposed repayment.
Authorised	The authoriser has authorised the proposed repayment.

Example LIXI Lab Ticket



After discussion with the LIXI community a solution will be found and we (LIXI staff) will complete the proposal and implement the solution.

LIXI GitLab Site Admin (John Matthews) @lixiapqg03pqq0fXgGPz commented 2 months ago Owner

Hi, @kerrimgroder are these the three requirements?

- is the debit account a sole or joint account
- who are the owners of the debit account
- has each owner approved the direct debit

Kerri Mcgroder @kerrimgroder commented 2 months ago Developer

Hi @johnmatthews, this is correct.

LIXI GitLab Site Admin (John Matthews) @lixiapqg03pqq0fXgGPz changed milestone to LIXI Master Schema 2.2.21 2 months ago

LIXI GitLab Site Admin (John Matthews) @lixiapqg03pqq0fXgGPz added **In Discussion** label 2 months ago

LIXI GitLab Site Admin (John Matthews) @lixiapqg03pqq0fXgGPz commented 2 months ago Owner

@kerrimgroder It is possible to indicate the owners of a sole or joint account that will be used to make loan repayments.

Using the `x_Account` attribute in `ProposedRepayment` to cross reference a `Non Real Estate Asses`, of type `Financial Asset`, we can cross reference the owner using the `x_Party` attribute within the `Percent Owned` element.

```
graph LR; subgraph Loan_Details [Loan Details]; ProposedRepayment; end; ProposedRepayment -- x_Account --> NonRealEstateAsset[NonRealEstateAsset.FinancialAsset]; NonRealEstateAsset -- Owner.x_Party --> PersonApplicant1[Person Applicant 1]; NonRealEstateAsset -- Owner.x_Party --> PersonApplicant2[Person Applicant 2];
```

However, this does not allow us to indicate that an owner has approved a direct debit.

Kerri Mcgroder @kerrimgroder commented 2 months ago Developer

Hi @johnmatthews this is perfect to indicate the owners. However you are correct we still need a flag to indicate that the owner has approved the direct debit.

Example LIXI Lab Ticket

The solution will include:

- A solution overview
- Paths of items being added
- Definitions for each items
- Data dictionary of all details
- XSD snippet of the solution

Solution

Overview

There will be an element called "Authoriser" to capture all owners of an account and whether they authorised the financial institution for the proposed repayment.

Detail

Add items:

- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.Authorised
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.AuthorityVerified
- Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser.x_Party

Definitions

Name	Definition
Authoriser	A party that has authorised the proposed repayment.
Authorised	The authoriser has authorised the proposed repayment.
AuthorityVerified	The authority of the authorising party has been verified.
x_Party	A cross reference to the authorising party.

Data Dictionary

Item Type	Name/Value	minOccurs	maxOccurs	type	use	documentation	path
element	Authoriser	0	unbounded			A party that has authorised the proposed repayment.	Package.Content.Application.LoanDetails.ProposedRepayment.Authoriser

Get Involved: Participate in Improving the LIXI Standard

If you are not registered with a LIXI Lab account, please contact us and we will create an account for you.

john.matthews@lix.org.au

Questions for you...



Question 1

We will be executing a breaking change soon where we will delete items that have been marked for deprecation. Do you see any value in maintaining a version with the deprecated items removed AND maintaining a version with the deprecated items still included?

Questions for you...



Question 2

How frequently should releases occur? Our current release cycle is 4 to 8 weeks depending on urgency of change requests. Would increasing the release frequency to a regular release once a week be useful?

Questions for you...

Question 3

What additional information could be included in our schema documentation? E.g. we are planning to include information about what types of elements can be cross referenced using our x-underscore attributes (x_) what else might be added?