OSCRE Industry Data Model™

Implementation Guidelines
## Contents

1. Introduction and purpose of this document
2. Overview of implementation projects
3. Data exchange implementations
4. Data model implementations
5. Resources and further information
1. Introduction and purpose

These implementation guidelines are intended to provide a reference for business sponsors, project managers and technical managers that identifies the key elements and resources required to successfully implement data standards.

It addresses implementations that use data exchange standards (integrations between separate systems that move data from one to another) and data model standards (where data is being described – typically in an ‘at rest’ state, for example in a database).

Throughout this document, two scenarios will be considered to provide specific examples if the points being discussed. The scenarios are colour-coded and each described below. Whenever you see these colours in this guide, it will be explaining how something applies in the corresponding scenario.

<table>
<thead>
<tr>
<th>Raise Repair – Data Exchange</th>
<th>Customer Data – Enterprise Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>In this scenario, the organization wishes to introduce a standards-based electronic ‘raise repair’ process with key contractors who perform repairs on their housing portfolio. The Raise Repair data exchange will support this activity.</td>
<td>In this scenario, an Enterprise Data Model is being created that includes information about customers, alongside other key organizational entities. The data model standards for Customer Data will be used to support this activity.</td>
</tr>
</tbody>
</table>
In this section, the kind of projects that may involve standards and their constituent stages are discussed and the role of standards set out.

1. Introduction and purpose of this document

2. Overview of implementation projects
   2.1 Project Types
   2.2 Project Stages

3. Data exchange implementations

4. Data model implementations

5. Resources and further information
2.1 Project Types

Data standards can be used to support a wide variety of project types, but the most common are:

1. **Integrating two systems**: this project is where two different systems need to share or exchange data and no existing interface exists (or, if one does, it is not fit for the purpose). The systems may be internal to your organization, or one may be external (i.e. used by one of your partners).

2. **Implementing a standards-based service interface**: in some circumstances you may want to make available a standard interface to one or more of your systems that allows third-parties (e.g. partners, customers, suppliers) to directly interact with your systems – perhaps send you information, or query your system for information they need. This is very similar to the first project type – integrating two systems – but differs primarily in that you might not have a predetermined partner to work with.

3. **Constructing or reviewing an Enterprise Data Model**: one of the strengths of the OSCRE modelling approach is that there is one unified model for both the data at-rest (data model standards) and the data in-motion (data exchange standards). The data model standards can be used to help review or construct an enterprise data model specific for your organization, the scope of which need not be the entire organization or the entire Industry Data Model – you can choose which parts of the Industry Data Model support your particular scope.

4. **Procurement**: a common use case for standards is during vendor selection for a new software platform. Here both data exchange standards and data model standards can be leveraged to explore the extent to which each candidate vendor provides industry standard interfaces and how aligned they are with the industry standard data model.
2.2 Project Stages

Any implementation project is likely to include most or all of these eight stages, and data standards have a role in each:

1. **Identifying the standards opportunity**: it is easy to overlook the contribution standards might make to a project. Data exchange standards can significantly simplify systems integration, and data model standards can dramatically reduce the time taken to architect new systems or system enhancements. It is recommended that consideration of the application of standards be incorporated into your standard project panel/project board meetings when reviewing and approving new projects. It may also be sensible to include these considerations alongside other data-related topics – Data Protection Impact Assessments, GDPR, Privacy by Design etc.

   If the project wasn’t created due to the introduction of the standards, then hopefully the opportunity to use the standards would have been identified in the project proposal, or – if not – during project review as part of an approval checklist (“has the use of standards been considered?”)

   The opportunity to use the standards should have been identified in the project proposal given the nature of the project, but – if not – then it should have been identified as part of an approval checklist (“has the use of standards been considered?”)

2. **Making the business case / securing funding**: the specific business rationale for your organization is of course unique to your particular context and strategic drivers. However, each standard comes with a Model Business Case that can be used to identify candidate benefits or opportunities that you can include in your business case or project justification.

   The Model Business Case for Raise Repair provides both generic and specific possible benefits, for example reduction of void duration, or provide a faster service to customers. Those that apply can be incorporated in the project business case.

   The Model Business Case for Customer Data provides both generic and specific possible benefits, for example support for GDPR, or the value of good customer knowledge in reducing evictions and/or voids. Those that apply can be incorporated in the project business case.
2.2 Project Stages (cont.)

3. **Alignment assessment**: having established the rationale for the use of standards, it is important to investigate the extent to which the standard can apply to your context. Standards assume a degree of alignment to the perceived common practices in the industry, but your organization may differ so much from these (either due to lagging behind best practice or leading innovation) that the standards are not well aligned. Reviewing the standards that might apply in your project and comparing them to your existing or proposed processes and data models will quickly highlight any misalignment. If you suspect misalignment, it may be worth discussing this with HACT, OSCRE and industry peers as this may be an opportunity for process improvement in your organization or enhancement of the standards.

   Reviewing the current process for instructing contractors will reveal the information currently being provided. This should be compared to the Raise Repair standard, and any differences reviewed together with the contractors involved.

   Customer data likely exists in several existing systems. Compare the information held across these systems with the Customer Data standard, and review any differences with key internal stakeholders and, if applicable, the software system vendors.

4. **Partner engagement**: most standards projects involve engaging with business partners, either internally or externally. Data exchange standards explicitly call out two or more stakeholders who wish to share or exchange information, but even data model standards typically require engagement with partners such as technology vendors. Early engagement with these partners, and ensuring they are aware of the standards and your intention to leverage them, is an important step in the project.
2.2 Project Stages (cont.)

5. **Process change**: because data standards remove some of the need for constant checking and rechecking of information, it is likely that the business processes will be impacted and – hopefully – simplified or streamlined. Any data standards implementation should review the processes involved with a view to such simplification.

   Reviewing the current process for instructing contractors will highlight several steps that might no longer be required, or may need adaptation. This is often key to realizing the benefits of standardization – reduced checking, higher quality information, and improved timeliness.

6. **Technical implementation**: over time, the technical implementation step should reduce in scale as the implementations themselves become more standardized. However, in the meantime, it is inevitable that some software changes to the technology systems involved is likely. The following sections cover this area in more detail.

   The potential changes to the way Customer Data is stored (identified in the ‘alignment assessment’ discussed previously) will likely require a review of data capture and processing, especially with respect to when information is available, and any implications of GDPR or other regulation.

   The technical implementation will involve changes to the interfaces and systems in both your organization and the contractors. This may be as simple as applying updates, but most likely will involve detailed collaboration on things like the transport mechanisms and business rules.
7. **Testing and go live**: the more streamlined processes that result from data standards implementation often result in much faster processing. People involved in these kinds of change often have a healthy scepticism with respect to the reliability of the resulting systems and so it is vital that testing is thorough, transparent and fully engages all key stakeholders. Confidence in these solutions is hard won but easily lost.

Testing of data exchanges can be complex as multiple parties are involved. The contractor would be engaged early in developing a shared test plan, and close coordination of resources would be required to ensure there were not delays during test periods or go-live. Good project management is essential.

Testing for changes resulting from standardization of data models are tested in the same way as any other software database change, and standard test regimes can be followed.

8. **On-going maintenance**: the use of industry standards to define the interaction between two parties, or the data model being used within a particular system, means that on-going support, maintenance and changes must be referenced back to these standards. Often the standards will provide good insight into where a problem or opportunity might lie, and failure to build in this standards-referencing to the life-cycle of the solution may lead to gradual deviation from the standards and diminishing benefits.

The degree of automation afforded by the Raise Repairs data exchange standard means that any changes internally or in the contractors’ systems may impact the other parties. This emphasizes the need for effective change control and management, established technical communication and coordination between the parties.

Having aligned the enterprise data model to the standards, processes or checkpoints must be established to ensure subsequent system changes do not deviate, and updates to the standard are considered and incorporated if appropriate.
In this section, five key aspects of implementations of data exchange standards between different systems and/or organizations are discussed.

1. Introduction and purpose of this document
2. Overview of implementation projects
3. Data exchange implementations
   3.1 Selecting the relevant data exchange standards from the library
   3.2 Implementing the extract, load or both
   3.3 Tips for transmitting the data payload
   3.4 Customizing data exchange schemas
   3.5 Support, maintenance and responsibilities
4. Data model implementations
5. Resources and further information

This entire section is applicable to the “Raise Repairs” data exchange standard scenario. The pages have been marked with a green triangular tag on the top right as a reminder of this.
3.1 Selecting the relevant data exchanges

The library of data exchanges can simply be browsed to identify relevant standards. However, the following techniques can also be used to help narrow down the candidate standards that may be relevant to your project:

- Identify key entities in your project domain – for example ‘lease’, or ‘contractor’. Find these entities (or the corresponding entities) in the Data Dictionary and then look at the data exchange standards that use those elements.

- Once some candidate data exchanges have been identified, look at the Model Business Case for each and see if it speaks to benefits you recognize. Also review the process map and see if the data exchange is addressing relevant parts of the business process (this may give you ideas for process improvement too).
3.2 Implementing the extract, load or both

**Extract**

It is generally easiest to implement the extract – i.e. creating the data exchange payload. This process will require mapping your database schema to the data exchange XML schema. The XML schema can be used to validate the resulting XML files – this can be done during development (for testing) and, if required, at runtime to (in order to catch any potential data-related errors before a file gets sent). However, the XML payload being valid is not a sufficient test, and the content must be carefully tested.

The technical implementation is often best achieved as a service or API implementation. That may be published or not, but then can be used both from the user interface (allowing users to ‘export’ a compliant file for semi-manual data exchange), from an ETL package or platform that you provide, or by third parties. You should also consider the process around the data exchange to ensure, for example, any appropriate approvals are obtained.

Many commercial databases provide tools to enable the direct generation of XML content, and this may be a sensible implementation choice and typically include transformation capabilities that are adequate for the mapping to the data exchange format.
3.2 Implementing the extract, load or both

Load

Receiving a data exchange XML payload is considerably more involved than creating and sending one. Several factors must be carefully considered:

- **Mapping** from the data exchange to your database schema;
- **Error handling** both the prospect of a malformed XML payload but also – and more importantly – content that fails to meet your business rules or assumptions made by your system. These latter errors must be handled carefully, as the sender may not be responsible for failing to meet your business rules, so each business rule violation that could arise during an import should be considered and an appropriate process or escalation put in place;
- **Staging** the received data is likely to be required, i.e. pre-processing the data into a holding area. This is useful partly because it makes the handling of the business rules above easier, but also because typically there are decisions (and potentially approvals) required before the onward processing. Obviously, this depends on the nature of the data, and sometimes straight-through processing is possible (and desirable).
3.3 Tips for transmitting the data payload

The standards only describe the format of the payload. No specific mechanism for transferring the data between systems and/or parties is mandated, and therefore must be agreed between the parties involved.

However, the following considerations should help in establishing the best approach:

- **Security** is a primary concern in any movement of data (whether or not the data is particularly sensitive or not). Therefore, any transfer mechanism – even those that happen within a secured network – should ensure that both encryption and authentication (of sender and receiver) are treated robustly.

- **Triggers and Timeliness** will often be key drivers in the transport mechanism. There are three typical scenarios, set out below, and discussed over page:
  
  - The exchange happens on a regular schedule – for example daily. This is often a good choice for slow moving data or data that support management, for example performance reporting or service level monitoring.
  
  - The exchange is triggered by an event in the senders systems. For example a new repair that is raised would be triggered in the Housing Association’s systems and then sent to the selected contractor.
  
  - The exchange is requested by the receiver. Whilst this could be driven by a receiver driven schedule, it could also be driven by events in the receiving organization. For example, a contractor carrying out routine maintenance might request an update of alert information regarding tenants in units that they were about to carry out work in.
3.3 Tips for transmitting the data payload

<table>
<thead>
<tr>
<th>Trigger/Timeliness</th>
<th>Potential Solution Types</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| Schedule           | SFTP (Secure FTP) is often a good choice for scheduled transfers as it is relatively simple to set up and to secure. Often hosting by the sender makes sense as there may be other scheduled data transfers potentially with other parties, but in principle either sender or receiver can host. | Scheduling allows both sender and receiver to know the transfer time in advance. However, care must still be taken for error handling, for example:  
  • What happens if the file is not ready on time/missing?  
  • What happens if the file has not been updated since the last transfer?  
  • What happens if the connection cannot be established? Some form of message queue management is desirable.  
  • What auditing is maintained? |
|                    | ‘Sender event’ and ‘receiver request’ trigger solutions also work for scheduled transfers (depending on whether the sender or receiver ‘owns’ the schedule). | |
| Sender event       | A web services/RESTful API is usually the best option in this case. The sender can implement an event subscription model allowing the receiver to register for the events they are interested in and to specify the delivery resource locator. SFTP techniques from ‘schedule’ triggers can also be used providing that the receiver implements the SFTP server (so the sender can ‘push’ the payload data). | For ‘sender event’ and ‘receiver request’, whatever technology is used, it is important to consider the time taken to create the payload content, and the potential need for caching by the sender to optimize this performance. |
| Receiver request   | Again, web services is likely to be the best choice, allowing the receiver to connect and request information dynamically from the sender. Again, SFTP techniques could be applied, using a simpler web API request from the receiver to trigger the sender into transferring the payload via SFTP to the receiver’s server. | Implementing an event subscription model is a well understood software design pattern, and should certainly include some form of message queue management. |
3.4 Customizing data exchange schemas

Wherever possible, the data exchanges should be used as-is from the standards. However, sometimes specific contextual circumstances require some modification of the standards, or – in cases where an existing standard does not exist – creation of new data exchange definitions based on the Industry Data Model.

HACT and OSCRE can support you in such customizations, that generally fall in to the following categories:

- Changes to cardinality – this is where the data model or specific data exchange standard mandates a particular cardinality that is unsuitable in your specific context. Perhaps the model considers a particular attribute mandatory but that attribute is not available in your context, or the model limits the maximum occurrences in a way that is too constraining for your context.

- There may be additional attributes or even whole entities required by your context that are not (yet) included in the data model. Modification like this should always be considered and – if possible – submitted to HACT and OSCRE for consideration in future versions of the standard as there is a good chance that if they are important to you, they will be important to others too.

- The Industry Data Model may also be used to create completely new data exchange schemas by assembling existing entities in the model.

NOTE: Although customizations are possible, and OSCRE recognize that there are cases when they are necessary, it is very important to understand that customization should nonetheless be a last resort, as it makes it harder to remain in-step with the standards, imposes non-standard constraints on your business partners, and increases the total life-cycle cost of your technology solution.
3.5 Support, maintenance and responsibilities

Once you have a data exchange up and running, it is obviously important to make sure it stays that way. Good preparation for this starts early in the process when engaging with your business partners who will be sending or receiving the information.

Some key points to keep in mind when discussing the roles and responsibilities for supporting and maintaining the data exchange once implemented are:

- Which systems or people are going to notice or be notified when something doesn’t work? What mechanism will be used for these notifications?
- What has to happen in each organization or system to deal with a data exchange failure?
- How are you going to keep each other informed of actions that may impact the other? For fault diagnosis and resolutions? For routine systems maintenance or planned downtime?
- Is the sender going to validate the outgoing file for conformance to the standard? Is the receiver?
- Other than conformance to the schema, what other elements are in the ‘contract’ between the systems? Performance expectations? Availability/up-time? Does it make sense to create a formal Service Level Agreement (SLA) to ensure expectations of both/all parties are aligned?
This sections explores four key aspects of using the data standards for data modelling, and in particular as part of an enterprise data model.

1. Introduction and purpose of this document
2. Overview of implementation projects
3. Data exchange implementations
4. Data model implementations
   4.1 Mining the Industry Data Model
   4.2 Alignment with organizational domain
   4.3 Expressing a partial data model (e.g. for a vendor selection RFP)
   4.4 Maintaining the Enterprise Data Model
5. Resources and further information

This entire section is applicable to the “Customer Data” data model standard scenario. The pages have been marked with an orange triangular tag on the top right as a reminder of this.
4.1 Mining the Industry Data Model

The Industry Data Model produced by HACT and OSCRE for the UK Housing sector is a subset of OSCRE’s larger industry and international standards product.

The model is expressed in several forms, including a tabular Data Dictionary and a Unified Modeling Language (UML) model. Which of these is most useful during your data modelling project depends on your needs.

The Data Dictionary is a very useful tool for searching for specific entities and attributes, finding their definitions, and seeing how they are used in data exchange standards.

The UML model contains almost all of the same information found in the Data Dictionary, but rather than being in a tabular form, the UML notation allows alternative visualizations – and in particular diagrams – to be created. This is useful when understanding the relationships between entities and also for communicating parts of the data model.

The UML model has been used to create a number of diagrams that call out specific aspects of the Industry Data Model – for example the customer data entities and attributes.
4.2 Alignment with organizational domain

Whether it is documented or not, every organization has a tacit data model that is expressed in the vocabulary used when discussing, writing about, or reporting on stakeholders, assets, and activities.

It is unlikely that this tacit data model will exactly match the Industry Data Model, and so – having identified the parts of the Industry Data Model that will be useful – it will be necessary to reconcile any differences.

The differences will typically take one of two forms:

• Terminology – these are where your organizational name for a particular concept differs from the Industry Data Model name, but the definitions are aligned. For example, your organization might call the contractual agreement between a landlord and a tenant a “Tenancy Agreement”, whereas the Industry Data Model calls this a “Lease”. These are conceptually identical but have different names.

• Conceptual – these are where the underlying meaning or intent of a term differs between your organization and the Industry Data Model, whether or not the name is the same. For example, your organization might consider a “Person” to include legal organizational entities (like Limited Companies), whereas the Industry Data Model distinguishes “Person” and “Organization” as distinct entities.
4.2 Alignment with organizational domain

**Implications of terminology or conceptual differences**

Differences in *terminology* are unlikely to be problematic: you can either decide to adopt the Industry Data Model terms (which strategically will bring greater alignment within the sector), or maintain a mapping between your organizations terms and those in the Industry Data Model.

*Conceptual* differences are likely to be more difficult to overcome. You should always begin by questioning the rationale for your organization’s different approach. If there is not a compelling reason for it, then you should consider adopting the Industry Data Model concept. However, if there is a strong case for your organizational perspective, you should approach HACT and OSCRE and explore the merit of modifying or extending the Industry Data Model. If neither of these prove effective in aligning the concepts, then the Industry Data Model may still be of use in other areas, but care must be taken that these conceptual difference do not undermine the related data models.
4.3 Expressing a partial data model

It is almost inevitable that the scope of your enterprise data model will overlap with the Industry Data Model, but highly unlikely that either will be a strict subset of the other. This means in practice that the Industry Data Model will likely express significant parts of your Enterprise Data Model, but there will be entities and attributes in the Industry Data Model that are not in your Enterprise Data Model, and vice-versa.

For these purposes, it is recommended that the Data Dictionary or UML representation of the Industry Data Model is copied, and then the relevant subset retained and all other entities and attributes removed. However, the naming of the entities and attributes that are kept should not be changed – keeping these exactly as they are will make it easier in the future to maintain and update your Enterprise Data Model in response to changes in the Industry Data Model.
4.4 Maintaining the Enterprise Data Model

Once an Enterprise Data Model has been created, the most challenging aspect is to keep it up to date and relevant to your organization.

By retaining the terminology from the Industry Data Model elements that have been re-used in your Enterprise Data Model, it is relatively easy to review any revisions to the Industry Data Model and consider their impact on your Enterprise Data Model.

Indeed, in some cases this modification to the data model may trigger a wider review of associated processes that may also need to be reviewed due to the same underlying industry changes that have driven the updated Industry Data Model.

When conducting any maintenance on your Enterprise Data Model – whether driven by changes to the Industry Data Model or other events in your organization – it is always important to be mindful of the elements of the Industry Data Model that were not used in your Enterprise Data Model: oftentimes changes to your organization may require pulling in more of the Industry Data Model, and if possible this opportunity must not be missed – otherwise you are likely to end up creating your own, divergent terms and definitions.
1. Introduction and purpose of this document
2. Overview of implementation projects
3. Data exchange implementations
4. Data model implementations
5. Resources and further information
All implementations are different; the challenges and opportunities that arise will vary. It is important to ensure that your project team is sufficiently educated on the role and benefit of standards.

Should you require further help, here are some options you may wish to consider:

- The data standards library contains other documentation beyond that referred to in this document that you may find helpful in some cases;

- If you are using consultants or software vendors you should verify their experience with standards and familiarity with the HACT and OSCRE standards;

- HACT and OSCRE can provide help and advice directly, and may be able offer consulting services directly, or recommend specialist third parties with experience in data exchange or data modelling projects.