Managing the Continuity of Datasets

This guidance relates to:

- **Stage 1:** Plan for action
- **Stage 2:** Define your digital continuity requirements
- **Stage 3:** Assess and manage risks to digital continuity
- **Stage 4:** Maintain digital continuity

This guidance has been produced by the Digital Continuity Project
1. **Introduction**

1.1 **What is digital continuity?**

Digital continuity is the ability to use your information in the way you need, for as long as you need.

Managing digital continuity protects the information you need to do business. This enables you to operate accountably, legally, effectively and efficiently. It helps you to protect your reputation, make informed decisions, avoid and reduce costs, and deliver better public services. If you lose information because you haven't managed your digital continuity properly, the consequences can be as serious as those of any other information loss.

Digital information is more vulnerable than paper. It is underpinned by high volumes of fragile relationships and access to it is mediated by fast-changing, proprietary or bespoke technologies, often only partially understood by a few people in your organisation. If you do not actively work to ensure digital continuity, your information can easily become unusable.

Digital continuity is *not* about digital archiving (although that may be a valid strategy). It is about ensuring an appropriate level of access to your digital information for as long as you need it.

1.2 **What is a dataset?**

A dataset is a structured collection of data created for a specific purpose. Datasets can be stored, managed and published in a variety of formats and technologies. They may reside in a database, spreadsheets, text files, web sites or printed to paper. They can be recorded in any medium capable of recording structured collections of data. Datasets often have value beyond their original purpose, being reused both inside the organisation they originated in, but also wider usage outside.

Although they critically depend on technologies, datasets themselves are not technologies. It is important to understand the technologies which are used to support your dataset - however, it is also essential to define your dataset independently of any particular technology. Technologies to store, manage and migrate structured data are very well developed. While they can require a high degree of skill to use, they are not typically the primary cause of continuity loss.
Examples of datasets:

**Inter Urban Congestion Data** held by the Department of Transport - a dataset providing journey time and traffic flow data for 15-minute periods since October 2009 on the Strategic Road Network in England.

**Vocational Qualifications** held by Department for Business and Skills - providing information on accredited vocational qualifications awarded in the UK. It shows vocational qualification awards by level, gender, age and occupational area.

**Seismic Sections** held by the British Geological Survey - document dataset containing paper copies of the seismic reflection survey sections.

1.3 **What is the purpose of this guidance?**

This guidance is aimed at anyone managing the continuity of datasets. It forms part of a suite that The National Archives has delivered as part of a digital continuity service for government, in consultation with central government departments.¹

There is a four-stage process for managing digital continuity,² this process can be applied either to managing the digital continuity of an entire organisation or of a single information asset (see section 2), such as a dataset. This guidance will follow the four stages and apply them specifically to the management of a dataset:

- **Step 1 – Plan for action**
  - Understand why datasets are at particular risk of losing digital continuity
  - Assign ownership
  - Understand the scope of the dataset
- **Step 2 – Define your digital continuity requirements**
  - Understand and document the business use required from your dataset now and in future
  - Define the technologies that you will need to maintain the required usability
- **Step 3 – Assess and manage risks to digital continuity**
  - Plan and perform a risk assessment

---

¹ For more information and guidance, visit nationalarchives.gov.uk/dc-service
- Risks around change
  - Step 4 – Maintain digital continuity
    - Managing the lifecycle of datasets – versions, publishing, archiving

It is important to undertake this process and consider the likely continuity requirements and risks as near to the point of creation of the dataset as possible. The risk of losing the use of data increases the longer this process is delayed, along with the cost of managing dataset continuity.

However this process can equally be applied to existing datasets, allowing you to understand their digital continuity. If you have lost digital continuity it is vital to try to restore it as soon as possible since your ability to do so will decrease over time through loss of understanding and gradually changing technologies and organisations.
2. Step 1 – Plan for action

The first step of managing your the continuity of your dataset is to make sure all involved parties fully understand the drivers for taking action and the scope of the process.

The drivers and objectives for undertaking this process will vary depending on several factors, for example:

- a new dataset is being commissioned (it cannot be emphasised too strongly that this is the best time to consider likely continuity requirements. See section 3.2.1)
- the dataset may be being transferred to a different owner (either inside the same organisation or external to it)
- the dataset relies on technology which is about to be decommissioned
- there is a wider process of managing digital continuity across the organisation
- the dataset is moving from one phase of its lifecycle to another, for example it is no longer actively collecting data but will be published and archived
- there is a concern the dataset may have already lost its digital continuity and the situation needs to be better understood
- a dataset has been selected for permanent preservation.

By understanding what the drivers are for taking action you can ensure that an appropriate scale of response is made and the correct people are involved.

The key to managing datasets is to understand them as information assets.

**An information asset is** a body of information, defined and managed as a single unit so it can be understood, shared, protected and exploited effectively.

Information assets have recognisable and manageable value, risk, content and lifecycles.

For a dataset, the information asset will contain not only the data itself, but all the metadata and documentation required understand and make ongoing use of the dataset. Managing a dataset as an information asset will enable you to manage it alongside the other information assets your organisation holds, for example, with appropriate risk management and disposal schedules. The
dataset must be included on an Information Asset Register (IAR) to ensure that it is centrally managed.³

### 2.1 Roles and responsibilities

To assure the continuity of a dataset, there must be a clearly identified Information Asset Owner (IAO)⁴ to take responsibility for it. The owner must be the person or role with accountability for the use, maintenance and eventual disposal of the data in the dataset.

The owner of the dataset does not have to be the same as the owner of any technology currently required to access it. For example, if your dataset resides in a database, the database server and the desktop applications required to access it may be managed and owned by your ICT department, but a different business unit will probably own the dataset itself. The owner need not be the creator of the dataset, or its primary user, however they must have sufficient understanding of the dataset and how it is used to make decisions.

The owner will need to work alongside a number of other roles and groups across the organisation to fully manage the digital continuity of the dataset. They will need to consult with the ICT department and any external technology suppliers or service providers. They will also need to liaise with the people who use the dataset, internally and externally, and now and in the future. Depending on the scale of the dataset and the drivers for managing continuity it may be useful to assemble a formal committee with all these groups represented.

### 2.2 Define the scope of the dataset

Next you must ensure that the scope, structure and interpretation of your dataset, or information asset, is fully defined so that you can understand what you need to manage. You will need to define precisely what specific data is in scope for your dataset, and provide comprehensive supporting information and metadata to fully explain it.

The scope of your information asset refers to which data and metadata should be included, and what is considered external. This section will describe some common cases which may influence the scoping of your dataset, and provide simple tests to help you determine whether data should be scoped in or out of your information asset.

³ For more information on information assets and IARs, see our guidance Identify Information Assets and Business Requirements nationalarchives.gov.uk/documents/information-management/identify-information-assets.pdf
The general test to determine whether data is in-scope or not is: ‘is the dataset understandable without it?’ Some data will be undeniably vital to the interpretation of the dataset, and will be obviously in-scope. In practice, there may be degrees of understanding available with or without certain data, so you will often have to make a judgement. It may not be legally, economically or technically possible to maintain all data.

The more widely you define your dataset, the more work is required to keep information on it up to date. However, a dataset which does not include the data required to understand it ceases to be an asset, and becomes a liability.

A single dataset may involve any or all of the following specific areas:

- coded data (section 2.2.1)
- external data (section 2.2.2)
- operational data (section 2.2.3).

### 2.2.1 Coded data

Datasets are often assembled using codes which represent something, but which are not directly meaningful in themselves. Codes may be classified as internal or external to the dataset.

External codes refer to widely available external standards, such as the country codes defined in the international standard ISO 3166-1. In these cases, it is sufficient (and necessary) to document which standard is being referenced by the codes; it is not usually necessary to include the data in the standard as part of your dataset.

By contrast, an internal code is one which has no meaning outside of the dataset in which it is used. For example, the number ‘1’ may mean that a responder to a survey picked the first answer of a set of available options. Without knowing what option the ‘1’ refers to, the dataset is not understandable. Hence, your dataset must be scoped to include the definitions of these internal codes.

Databases often, but not always, provide special lookup tables for these internal codes. Other dataset formats may not provide this essential information. For example, a Comma Separated Value (CSV) text file exported from a database may only contain the raw codes, rather than the final lookup values.
Internal code lookup values must be included in the scope of the dataset. You may decide to replace codes with their look-up values directly, avoiding the issue of maintaining them separately. However, this makes datasets a lot bigger, as the description of a code is usually larger than the code itself. It can also make the dataset more prone to integrity failures if the data will change in future, as many software tools will enforce valid code values defined in lookup tables, but usually not on free-text descriptive fields.

2.2.2 External data

Data frequently contains references to external data, stored or managed in other locations or technologies. For example, a table in a database may reference names of files in an external file store, or list part numbers in a catalogue. You must consider the fundamental test of how well the dataset could be interpreted and used without the externally held data and ask:

1. Is the external data itself in-scope for your dataset?
2. Are the references in-scope for your dataset?

**Example 1:** A dataset holds data on sales of particular items which are identified within the dataset only by part numbers which are then further described in a separate catalogue. In this case the external data would have to be considered within scope of the dataset, because without knowing what the part number refers to the sales information is useless for future planning and analysis.

**Example 2:** If a dataset is constructed to track how often particular files are accessed, then the content of these files is irrelevant to the purpose of the dataset. Hence, you do not need to include the data contained in each file in your dataset, even though the dataset records a reference to this external data. In this case, the file references are definitely in scope for your dataset, but the files themselves are not.

Data which is held in a different IT system, but which is still considered within scope of the dataset is at greater risk of digital continuity loss because of the risk of these references becoming broken. See section 4.2.1 for how to assess and manage this risk.

Where the external data is within scope, but its reference only exists as a retrieval mechanism, the reference is not part of the logical dataset itself. The reference in this case is part of the physical structure of the dataset, required by the specific arrangement of technologies used but not necessarily required if those technologies change.
2.2.3 Operational data

It is common for software to track the state it is in, and to provide audit trails and change history. Frequently, the data used to track these states is recorded using the same systems and technologies as those used to record your dataset data. For example, a database may include a 'last modified date/time' field in a main table, so the software knows whether to update a row or not.

As with all metadata, you must consider whether you will need to maintain this information for the future. For example you may need to be able to show who has accessed particular records, or when a particular version was created otherwise this information may become redundant once the dataset ceases to be actively updated.
3. **Step 2 – Define digital continuity requirements**

To define your digital continuity requirements you must describe how the dataset needs to be used to support your business, and understand what technology is required to ensure that usability. These requirements must be defined for the current usage, but also how those uses might change over time and how different versions of your dataset may be necessary to address the different requirements.

You need to be able to answer the following questions:

- Why was the dataset created, does it have a planned lifecycle of business use?
- What are the usability requirements of the dataset for those business uses?
- What technologies are needed to support those uses, now and in the future?

### 3.1 Business uses

In order to make sensible decisions about the continuity of your dataset, you must understand who needs to use it and for what, including not only your own organisation but other potential audiences. If you do not understand this it will be impossible to make decisions about how the dataset should be managed, you will be at risk of either having insufficient usability from your dataset in the future, or wasting resources maintaining usability which is not required.

You should document the answers to the following questions so that the context and provenance of the dataset’s creation are fully understood:

- Why was it constructed? For example was there a legal mandate or political driver?
- Who was it for?
- What business objective does it help to deliver?

You should also think ahead to potential future audiences for the data. Your dataset may have archival value, or there may be benefits from wider publication or further analysis after its initial use is fulfilled. The government’s public data initiative encourages the publication of as much data as possible. You may want to consider imposing a schedule to review these requirements to ensure that opportunities are not missed in the future.

### 3.2 Usability requirements

From this understanding of the audiences and business objectives, you will be able to define your usage requirements. These requirements will affect your choice of appropriate
technologies and file formats, the degree of documentation you will require, and the cost of maintenance.

In brief, you need to answer the following questions:

1) How will you find the information?
2) Who can open the information, how and when?
3) How do you need to be able to work with the information?
4) What do you need to be able to understand about your information?
5) To what extent do you need to trust your information or demonstrate that it is what it claims to be?

Different audiences may have different usability requirements, which may mean you need to maintain different versions of the dataset for each audience. Similarly these usability requirements may change over time.  

Example: Your organisation is performing a survey, the results of which will be stored in a database. Initially your requirements will focus on being able to put data into the database, however then you may want to make the database read-only and your requirements will be focused on searching the data and generating reports. You may also need to create a separate version of the data which anonymises the results, allowing it to be shared and queried by a third party who should not have access to the original data.

For a dataset, the most complicated usability requirement is usually around the idea of understanding the dataset itself, where the data came from, what it means and how it is structured. If this understanding of the meaning of the data is lost, the dataset can become unusable, even if the data itself can still be read.

In brief, you need to answer the following questions:

- Do you understand where your data came from?
- Do you understand the intended meaning of the data?
- Do you understand how and why the data in your dataset was created?
- Do you know how accurate the data in your dataset is?

---

5 For more information and guidance on understanding these usability requirements, see Identify Information Assets and Business Requirements nationalarchives.gov.uk/documents/information-management/identify-information-assets.pdf
• Do you have enough information about your dataset for others to be able to fully understand it?
• Do you have enough information about your dataset to validate that it is still what it set out to be?

3.2.1 Methodological issues

Understanding how the actual data in the dataset was collected and processed is necessary to understand what reliance can be placed on the dataset. In particular, the methodologies used for capture and validation will influence the quality of the data in the dataset, which in turn will influence how that data may usefully be used in future.

For these reasons, it is important to provide an understanding of how the data in the dataset was collected and validated, allowing users of the dataset to judge whether the data is appropriate for their use. Even if the dataset is intended for only very short-term use, this information on creation is considered a vital element of the dataset which should not be overlooked. Additional uses and applications for the data will often be found even years after its creation, and without the methodological definitions, the data is worthless. It is very easy to lose understanding of the collection and validation methodologies employed. Hence the data creator, ideally at the time of creation, is best placed to describe and document the collection and validation methodology.

Note that low-quality data can still be very useful – particularly when combined with other data sources – as long as the quality level is first understood. Without an understanding of the data quality, undue reliance may be placed on low quality data, or high quality data may go unused. Further note that quality is a subjective measure – for some purposes, a dataset may be considered high quality when in other circumstances it would be low quality.

Examples of issues that need to be addressed in the methodology:

Accuracy: Some data is produced by empirical observation, which will naturally carry some margin of statistical error. The instruments used to perform measurements will likewise have some range of precision, for example measuring distances using a piece of string versus using a laser rangefinder. If the rangefinder was used, what model – and what bias, if any, is introduced by it? It is important to document whether any compensations were applied to the data to correct for these issues.

Question types: Where data has been collected by survey, you should document whether the resulting data is in response to questions asking for facts, opinions or estimates. For example:
• Facts: ‘How tall are you?’
• Opinions: ‘Do you agree that a fortnightly rubbish collection is adequate?’
• Estimates: ‘Roughly how often do you visit a heritage attraction?’

**Missing values:** Where a dataset may have missing values, you should document how these values are dealt with. For example, are survey respondents who fail to supply some answers contacted personally, are the missing values created by statistical means, or does the dataset contain null values in these cases?

**Sampling:** It is often not possible to collect all available data, so some form of sampling is often employed. In order to understand and correct for any bias introduced by samples, it is important to document how samples were selected and if any weighting was introduced for unequal probabilities of selection. Any other statistical corrections to the data should also be documented.

**Source data:** Some datasets may be assembled from large amounts of source, or underlying data, or the finished dataset may only contain summaries of the source data. It is important to consider whether the source data also needs continuity planning, or if it can be safely discarded. It may be very expensive (or impossible) to reassemble the dataset if the original source data is no longer available. In addition, it will be harder for third parties to trust the data in the dataset without being able to verify the methodologies used in its construction.

**Repeatable experiment:** For some data generated by an experiment or a field study, you may be required to make enough supporting information available so the experiment can be repeated. Some scientific datasets are generated by simulation, which in principle could be performed again, as long as the simulation parameters used to generate the data are known, and the cost of doing so is not prohibitive.

**Ongoing data collection:** Over time, datasets may change, and additional versions of a dataset may be produced. It is important to document whether the collection methodology has also altered over time, and if new data was added to long established or historic data, which may have used different collection methodologies.

**Data entry:** It is important to document how data was recorded in the final dataset. For example, was the data in the dataset captured directly in a digital format or transcribed? How was any transcription done, and how were transcription errors identified and corrected? Any further quality assurance steps undertaken must be documented.
An exemplar demonstrating the documentation of the collection and validation process is the work done by the National Digital Archive of Datasets (NDAD) on ‘The Annual Agricultural and Horticultural Census’. Of particular interest may be the sections on:

- nature and content, including
  - scope and content of the dataset
- structure, including
  - how data was originally captured and validated
  - statistical processing and aggregation
  - constraints on the reliability of the data.

3.2.2 Semantic issues

Each attribute, or field of data in a dataset normally carries a name – for example, ‘Creator’ or ‘Size’. However, a name alone does not provide sufficient information to understand what the data in that field actually means. Does ‘Creator’ refer to a person or an organisation? At the very least, a description is required explaining what sort of data should appear in that field and how to understand the possible values in it. Where a number reflects a physical property, then the units and precision of measurement must also be documented.

The values in a field may be constrained in some way. A field may only contain a limited range of defined values, or those values may be dependant on a value which appears in another field. For example, a field called ‘Colour’ may only contain the values ‘Red’, ‘Blue’, and ‘Green’ when the ‘Simple Colours’ field contains ‘True’, and otherwise can contain any colour. All intended constraints on valid field values must be documented.

In some cases, a field may have a widely understood meaning, which is already documented externally. For example, a ‘Creator’ field may have the same meaning as that documented in the ‘Dublin Core’ metadata standard. Where existing metadata standards are equivalent, then the field documentation should reference them, as this will substantially increase the ability to interpret and use the data in that field.

---

6 This reference is to the archived copy of the NDAD web pages in the UK Government Web Archive. The documents that NDAD used as supporting information are available from The National Archives’ DocumentsOnline: [http://webarchive.nationalarchives.gov.uk/20101104101827/](http://webarchive.nationalarchives.gov.uk/20101104101827/) [http://www.ndad.nationalarchives.gov.uk/CRDA/4/detail.html](http://www.ndad.nationalarchives.gov.uk/CRDA/4/detail.html)

7 See [http://dublincore.org](http://dublincore.org)
3.3 Structure

A dataset is, by definition, a collection of structured data. It is essential to document the structure of the data in your dataset, using both logical and physical models:

- A logical model captures what is logically essential in your dataset, regardless of the way it is currently represented using your technologies.
- A physical model captures the particular way your dataset is physically structured using your current technologies.

Both forms of model focus on:

- Types – the different kinds of data you have and their attributes
- Relationships – which types of data can be related to other types of data
- Cardinalities – the number of times data can appear against other data
- Constraints – if there are any constraints on what constitutes valid data

However, a logical model may include fine distinctions informing the meaning of the dataset, which a physical model cannot capture. Conversely, a physical model will show where and how the actual data is stored and managed, which a logical model will not do. The logical model shows you what you need to manage over time and change – the target of your continuity management; the physical model shows you how your data exists at the current time.

Models always require additional explanation to be useful, and the relationship between them must also be made apparent. A full description of a simple dataset is provided in the Appendix – a worked example of logical and physical models.

There is no one right way to produce logical or physical models. The examples given in the Appendix use Unified Modelling Language, Entity Relationship Diagrams and ad-hoc notes. The essential features of models and diagrams are that they should be fairly easy to produce, and, most importantly, that they are comprehensible by other people who need to understand your dataset.

A single model alone will not normally be able to fully describe the structure of your dataset. A logical model will fail to explain how the data is represented with your current technologies. Physical models can have implicit meanings, internal references and extra data which may not actually be required. They may not explicitly describe all the entities in the logical model. Some
of the data in a logical model may be calculated dynamically by software, and may not appear in physical data models.

To achieve continuity, you should have:

- a logical model of your dataset, as it ideally appears
- a physical model of your dataset, describing how it is currently encoded
- up to date documentation mapping the physical structures to the logical structures
- up to date documentation on any dynamically calculated parts of your dataset.

The logical model represents what you need to work with over time, and does not itself change, unless you change the definition of the dataset itself. As technologies change, you will produce different physical models. To understand and maintain your dataset it is necessary to document and map the physical models back to the logical model, ensuring that any changes to the data have been captured in the models and in the documentation. It is all too easy for the data to become desynchronised.

3.4 Technology requirements

It is very common to confuse datasets with the technologies used to encode and manage them. Datasets are not files, databases, software, applications, web pages, encodings, paper, ink or any other technological artefact. Datasets can span several different technologies and a single technology may support multiple datasets. The data in a dataset is all of the structured data required to make use of the dataset (and no more), regardless of how it may be physically split up, encoded, calculated, stored and managed using various technologies.

You must logically separate the collection of data that you need to work with over time and through change from any particular technical representation of it at a moment in time.

Thus far we have been talking exclusively about the datasets themselves, however in order to manage the continuity of the dataset you must also assess and manage the technology environment that is needed to support the dataset’s use. Some technology will always be required, and you may need to transition datasets between technologies. For more on the general process of mapping your data to the technology which supports it, read *Mapping the Technical Dependencies of Information Assets*.8

In brief, you need to answer the following questions:

8 See our guidance *Mapping the Technical Dependencies of Information Assets*

nationalarchives.gov.uk/documents/information-management/mapping-technical-dependencies.pdf
• Do you understand all the technologies which are required to use your dataset?
• Are you using the simplest possible technology to manage your dataset?
• What skills are required to support and use your dataset technologies?
• Can your proposed dataset formats include all of the data scoped to be in your dataset?
• Have you made your data available in a way that allows it to be utilised as fully as possible by potential audiences?

**Example 1:** At its simplest a dataset can be stored in a simple text file, at the other extreme it may require the interaction of complex technological systems automatically grabbing, modifying and exporting the datasets. Complex technologies tend to have a greater risk for long term support and interoperability, meaning that the datasets are at greater risk of losing their digital continuity.

**Example 2:** At the extreme end of complexity, there are applications consuming and transforming data from multiple sources, presenting the combined results back to the user. They may have dedicated client software, and application servers, databases and other storage systems. For these types of dataset, some data may be calculated on-the-fly by the application, or automatically provided via links to other resources on the Internet. These datasets have a very high technological dependency and it may not be straightforward to separate the data from the application. They require highly trained staff to operate, and may involve bespoke customisations or software which are unique to your environment.

**Example 3:** At the other end of complexity, a simple dataset may be stored in a text file, possibly in Comma Separated Value (CSV) format, or a simple spreadsheet. These technologies are widely interoperable, allowing the data to be consumed, re-used and re-purposed. They exhibit low technological dependency, and are easy to use without specialist technical knowledge.

### 3.4.1 Dataset formats

The file format that a dataset is stored in generally depends on, and has an impact upon, the technology that is used to create, modify and access it. It is likely that the format may change over the lifecycle of the dataset, for example it may be exported into different formats for different audiences and uses. Migrating between formats, or creating additional formats poses a
risk to the digital continuity of your dataset, different formats may have very different methods of representing the data, and may not capture all the data in other technologies by default.

For example, there may be lookup values, or audit trails which are not exported along with the underlying data. Additional guidance is also available for evaluating file formats.⁹ You must ensure that the scope of the data in the export format matches the scope of the data in the source format that you need to keep. The Digital Continuity Procurement Framework¹⁰ provides services for data migration which may be of use in these cases.

Most datasets benefit from being able to apply machine-processing to them, to combine the data with other datasets, or to perform analysis over the raw data. In general, you should make your data available in machine-processable formats while ensuring that it is also human readable; to the level where a person has enough information to be able to understand what is contained in the data, or provide both as separate encodings of the same underlying dataset.

---

¹⁰ Find out more about the Framework: nationalarchives.gov.uk/information-management/our-services/dc-technical-tools.htm
4. **Step 3 – Assess and manage risks**

Risk assessment is a key element of managing your digital continuity. You should perform an initial risk assessment as part of this four-step process, including developing and carrying out an action plan to manage the risk. However you must also plan an ongoing schedule of ongoing risk assessments to track your progress and ensure that new risks are fully understood as the dataset moves through its lifecycle or undergoes changes.

4.1 **Risk assessment process**

Before you carry out a risk assessment, you should establish a framework for managing risks to digital continuity. This defines the process you will follow and identifies the outcomes you wish to achieve. It will help to ensure consistency in the way your risks are identified and managed and will enable you evaluate the effectiveness of the actions you take.

To be effective, it is important that your framework is consistent (as far as possible) with the information risk management processes that are already embedded within your organisation. If you have an existing framework for managing information risk, you should extend this to include risks to digital continuity.

Your digital continuity risk management framework should do the following:

- set out **roles and responsibilities** for managing risks to digital continuity
- define **objectives** and success criteria for the process
- define **the scope** of your risk assessments
- describe **the process** of how risks will be identified, analysed, controlled, recorded, monitored and reviewed
- consider how you will provide **assurance** of this process.

For more information on planning and performing a digital continuity risk assessment, see the *Risk Assessment Handbook*¹ which also provides examples of the common risks which threaten the continuity of digital information in general.

---

4.2 Risks during change

Risk assessments should be performed regularly to ensure that the existing risks are being well managed and that new ones are detected early and appropriate action is taken. You should also risk assess as part of any change process, for example if a dataset changes ownership or moves into a different section of its lifecycle (see section 5).

It is also important not to lose sight of the fact that datasets change over time. They can be regularly updated, or constantly overwritten. There can be a core reference dataset, from which manifestations of the data are tailored to different requirements of, for example: licensing, access, usability. When addressing the digital continuity of datasets, it is important to remember that it is not a one-off exercise; the procedures need to be regularly re-visited, and applied as necessary to all versions and manifestations of the data.

Many of the risks to dataset continuity involve not being able to understand the dataset, even if it is possible to read the data, the lack of understanding, or certainty of understanding in where the data came from and how it can be used may lead to the dataset becoming completely unusable. By following the process outlined in section 4.1 and documenting your findings many of these risks will be mitigated, however there will still be areas where continuity could potentially be lost.

The following sections outline two example areas that are at particular risk during times of change, losing understanding of the relationships within the data itself and between the dataset and external data. These are not the only risks, but they are good examples of the complexity that may be involved.

4.2.1 External references

Section 2.2.2 describes how to decide whether external data that is referenced by your dataset should be considered in-scope or not. Any dataset which uses data across different systems is at a greater risk of digital continuity loss because changes could take place to the external data without warning, or even without your knowledge. As part of your risk assessment you must assess the safety of these references against time and change and attempt to put in place appropriate risk mitigation strategies.

There were two types of external data:

- where the external data itself was in-scope of the dataset
where only the references were in-scope of the dataset.

If the reference is part of the dataset, but the data it points to is not, then you are generally safer. This is because, even if the reference becomes broken, the dataset does not lose much, if any, value.

Where the reference is required to retrieve data that is part of the dataset, then you face higher risks to continuity. References tend to be in one of two forms: locations, or identifiers. A location tells you where to find the external data, whereas an identifier only tells you what you need, but not where it is.

As a general rule, location-based references are more susceptible to breaking, as it is more common for things to be moved than for their identifiers to be changed. However, having an identifier-based reference does not mean you are safe, and having a location-based reference does not place you immediately at risk. With identifiers, you have the additional challenges of knowing where to look for something with that identifier, and this information itself can be lost.

Any reference which is used as retrieval mechanism for part of your dataset introduces an immediate risk to the continuity of your dataset. Both the reference and the data it points to must be carefully managed, to avoid breaking the link and thus losing parts of your dataset. You must put in place processes to regularly test that the references are maintained, or establish communications protocols for the external organisation to inform you of any potential issues.

4.2.2 Internal identifiers

Datasets often use internal identifiers for items of data within them, so that relationships between those items can be recorded. This is similar to the coded data example (see section 2.2.1), where an otherwise meaningless value identifies an item of data which another item of data is related to. Internal identifiers can be used to create highly complex relations between items in a dataset, which go beyond the simple look-ups which coded data provides.

In general, internal identifiers themselves are not directly in-scope for a dataset, as the numbers (or other form of identifier) have no meaning outside of that particular encoding of that data. They do not usually carry any guarantee of being unique, or that what they refer to will never change. The same identifiers may be used in different datasets to refer to other things. They may even be re-used within a single dataset to refer to different kinds of thing. For example, primary keys in a database are only stable within a specific instance of a database. As soon as
data is moved between databases, or exported into different formats, care will have to be taken to ensure that these internal identifiers are not confused and the relationships lost.

The relationships encoded by the internal identifiers must be preserved, i.e. they are in-scope for the dataset – but note that the particular method of encoding them is not generally considered in-scope because it can vary according to technology.

Despite the fact that internal identifiers themselves are not directly in-scope, they often become de-facto external identifiers without deliberate planning. It is very common when migrating or exporting datasets into new formats or technologies for the internal identifiers to be directly exposed. For example, if exporting data in a database into a Comma Separated Values (CSV) file, the internal numeric keys used in the database to identify rows may also appear in the CSV file. Another way in which internal identifiers can be exposed is if you have applications which consume data from more than one database, and which may record internal identifiers from one database in another.

Once other systems begin relying on your dataset’s internal identifiers to refer to items of data, then you immediately face some difficult continuity issues. You may be forced to preserve the internal identifiers, if it is important that these external references back into your dataset do not break.

You may even have to use different references to encode the actual relationships as the dataset moves forward, keeping the old ones to assure a certain level of continuity. For example, if you need to merge two datasets, but they both use numbers beginning from ‘1’ as their internal identifiers, then you will get conflicts in the identifiers. You cannot continue to use these internal identifiers to encode your new dataset relationships. Hence, you must create new internal identifiers for these items of data. If external data is using these identifiers (and you cannot afford for these relationships to break), you must now preserve the old ones to prevent a loss of continuity, even though they no longer have any direct meaning within your dataset. Even preserving identifiers in this way may pose some further continuity issues, as external datasets almost certainly will not reference which source dataset its identifiers originally related to.

Key considerations for avoiding and mitigating risk:

- Internal identifiers should be kept as internal identifiers. As such, they are not directly in-scope for your dataset, but the relationships they encode are.
- If you need external data to reference items of data in your dataset, then you should explicitly construct robust external identifiers, which carry guarantees of persistence and uniqueness.
1. Do not rely on internal identifiers provided by some particular technology (e.g. key fields in a database) as a good external identifier.

2. Consider using ‘URI sets’\(^\text{12}\) as external identifiers for your data items. These facilitate re-use and linkage of data across datasets.

- If internal identifiers ever become de-facto external identifiers, then you face difficult continuity issues going forward:
  1. They must be regarded as in-scope for the dataset, even when they no longer fulfil their original purpose of encoding the internal dataset relationships, or:
  2. You accept that these external references can be safely broken.

5. **Step 4 – Maintain digital continuity**

To maintain the continuity of your dataset you must manage it throughout its lifecycle which may include creation, use, change, re-use, publication, archiving and deletion. In this section we will describe several specific use cases and also how to document these.

5.1 **Versions**

It is important to note that datasets may or may not be static over time and you must consider whether you need to keep different versions. A dataset may exist in multiple versions during its lifecycle; each stage of the lifecycle may involve a new version, or there may be a different version for each potential audience/user within a lifecycle stage. The dataset may be:

- produced once – a set of data which is generated/created once and never changed. For example, the results of a scientific experiment
- regularly updated – a dataset which is updated on a regular schedule, for example an annual survey. Each new update could be considered a new, distinct version of the dataset
- live – a dataset that’s continuously updated, overwriting existing data with newer entries, for example a telephone directory. If a dataset is continuously updated, you may decide that particular snapshots of it are required to be maintained for audit, publication or archival purposes.

It is not uncommon for even a dataset ‘produced once’ to have corrections applied to it, separate from a regular or live update process. For data that has been corrected over its lifetime, either by initial checking and validation on creation, or by subsequent correction, you need to decide how much of this process has value to your audiences. Obviously, all businesses take pride in the accuracy of the data they have, for their own use, and for supply to external clients, however, if inaccuracies are found in data, does your business need to capture the following:

- The fact that validation/subsequent correction was carried out?
- The processes used to do this?
- The erroneous data itself?

If so, you will need to ensure that the data is managed appropriately in order to separate ‘good’ data from ‘bad’.
Different versions of a dataset may have different audiences or uses and, therefore have different continuity requirements. If this is the case, it may make sense to manage them as different information assets, and to have separate entries in your Information Asset Register.

Each version of the dataset will have its own usability requirements which will have to be maintained. These requirements will include the fact that users will need to be able to find the correct version, understand how that version is distinct from the others and also trust that they have the right version.

5.2 Publication

Publication is an important stage in the lifecycle of many datasets, allowing others to benefit from the data in it. It facilitates combination with other published data, often producing highly useful results unanticipated by the creators of the original datasets. Where the dataset is fully open, containing no sensitive or personal data, a good method of publication is via the web.

You should consider the following:

- publishing as much of the underlying data as possible
- publishing it on your own website
- publishing data in a machine readable format
- presenting it in a manner that can be directly downloaded via data.gov.uk and harvested into the UK Government Web Archive.\(^{13}\) There should not be any extra input required from the user such as logins, pick lists or tick boxes.

Once you have checked that your dataset can be downloaded from the UK Government Web Archive, you can take your dataset down if it is no longer required to be directly linked on your own site.

Important points to remember:

- You cannot include personal/sensitive/ Freedom of Information (FOI) or Environmental Information Regulation (EIR) exempt data.
- You must adhere to COI guidance on underlying data publication.\(^{14}\)

You should ensure compliance with the Government’s Public Data principles.\(^{15}\)

\(^{13}\) UK Government Web Archive: nationalarchives.gov.uk/webarchive/

\(^{14}\) This has been out to consultation, but the final version not yet published. The interim guidance is available from http://coi.gov.uk/blogs/digigov/wp-content/uploads/2010/07/TG135-Underlying-Data-Publication-v0-4.pdf
• Public data will be available and easy to find through a single easy-to-use online access point ([http://data.gov.uk/](http://data.gov.uk/)). Data published on your website should be linked to, and thus downloadable from your website via data.gov.uk
• Guidelines for publishing metadata through data.gov.uk are available from [http://data.gov.uk/blog/publish-your-data-on-datagovuk-a-simple-how-to-guide](http://data.gov.uk/blog/publish-your-data-on-datagovuk-a-simple-how-to-guide)

5.3 Archiving

There may be a business requirement that once the dataset is no longer being updated, or it is no longer being regularly accessed, it should be archived for permanent preservation. The organisation may have obligations under the Public Records Act to ensure that the dataset is permanently preserved.

If you are a central government department, data published on your department’s website, provided it conforms to the current COI guidance on formats and presentation will be automatically harvested into the UK Government Web Archive. It is therefore important to ensure that your web page carries sufficient supporting information about the purposes, design and methods of the datasets. If your dataset is designated an Official Statistic, this will ensure that it meets the UK Statistics Authority requirement to deposit the data with the relevant national archive.¹⁶

Another place of deposit for datasets is the UK Data Archive. Note that the UKDA is not a comprehensive national archive; it is a Place of Deposit under the Public Records Act for certain open (FOI exempt) social science datasets that have been selected by the National Archives for permanent preservation.

It is important to maintain awareness of the possible existence in your organisation of the dataset at different levels of granularity, sensitivity and disclosure; all of which may yield different answers to the questions in this document; and all of which are subject to change over time, and may therefore need to have their digital continuity requirements periodically re-assessed. For example, government departments should be posting their non-disclosive data on their website, and from here it should be possible to capture it into the UK Government Web Archive; in which case, for the non-disclosive version, digital continuity is taken care of. However, it is often the disclosive version of the data that is of long-term value, which will need

---
to be actively managed to maintain its digital continuity. Remember that the sensitivity will diminish over time, enabling, in the majority of cases, eventual disclosure.
6. Next steps

6.1 Further reading

This guidance relates to the following other publications and digital continuity documents:

**Managing Digital Continuity** - Details of the four-stage managing digital continuity process which can be used for any digital information, including datasets.

[link to Managing Digital Continuity]

**Machinery of Government Changes** - Guidance to help Departments and other public bodies with the task of transferring records, information and knowledge in the event of a Machinery of Government change. Specific guidance on datasets and databases can be found on page 20.

[link to Machinery of Government Changes]

**Public Bodies Reform** - Advice and guidance for public bodies going through reforms, including information on how to transfer information assets.

[link to Public Bodies Reform]

**UK Statistics Authority Code of Practice** - Intended to ensure that the range of official statistics meets the needs of users; that statistics are produced, managed and disseminated to high standards; and that they are well explained.

[link to UK Statistics Authority Code of Practice]

**Public data principles** - A set of principles for the publication and release of public data

[link to Public data principles]

**COI guidance on underlying data** - This guidance is for all public sector communicators, website managers and policy teams. It covers the minimum requirements for releasing data published in documents in a re-usable format as well as the optimum form.

[link to COI guidance on underlying data]
7. Appendix - Worked example of logical and physical models

7.1.1 An example logical model

Below is a logical model of an example file-counting dataset. The model is presented in the form of a Unified Modelling Language (UML) diagram, which is often used to produce logical models.

![Logical Model Diagram]

This model shows the essential aspects of our dataset providing a count of access to files. There are main types of data, which have relationships to other types of data. We can see that Users are employed by a Department, and that a User can either be Internal Staff, or a Contractor. If they are a Contractor, they are provided by an Agent. Finally, a count is provided of Access to a File (which may exist in a parent Folder).

Each of these main types of data can have various attributes – for example, a User has a Name and a Payroll number. Finally, the numbers next to each relationship and type indicate the cardinality of the relationship for that type. For example, the ‘1’ next to Agent indicates that a Contractor can be provided by only one agent. The ‘0..*’ next to the Contractor indicates that an Agent can provide from none to an unlimited number of Contractors.

A logical model describes the types, relationships, cardinalities, and possibly constraints of the data which can appear in a dataset. It does not say how this data is encoded or what sorts of technologies are used to manage it.
7.1.2 An example physical model

Below is a physical model of the same dataset given above. It is provided as an Entity Relationship Diagram (ERD), with a note to cover items which the diagram cannot cover. ERD diagrams are common when producing physical models of data in databases.

The LogFileName gives us the location of a log file which contains file accesses by a user.

Each log file is a text file encoded using UTF-8. The internal structure of the log file is as follows:

1) Each line contains two fields, separated by a tab character (0x09)
2) Each line is terminated by a newline character (0x0A)

<Date Time> (0x09) <FilePath> (0x0A)

The <DateTime> field is written out as an ISO 8601 formatted date. The <FilePath> field contains the full path and name of the file accessed.

Figure 2 - a physical model of a dataset counting access to files

The model in Figure 2 shows two tables – a User and an Agent table. A note is also given describing the structure of a log file. The rows of the diagram describe columns in a database table. The diagram uses various codes and symbols to indicate keys, data types and null status.

The User table has these columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td>An internal reference used by a database as a Primary Key (PK) for each row.</td>
</tr>
<tr>
<td>Username</td>
<td>Maps to the Name attribute of the logical User.</td>
</tr>
<tr>
<td>PayrollNumber</td>
<td>Maps to the Payroll number attribute of the logical User.</td>
</tr>
<tr>
<td>DeptCode</td>
<td>Maps to the Code attribute of the logical Department.</td>
</tr>
<tr>
<td>AgentId</td>
<td>An internal reference used by a database to relate to the primary key of the Agent table. It is a Foreign Key (FK) to the PK of the Agent table.</td>
</tr>
<tr>
<td>LogFileName</td>
<td>The name and path of a log file containing details of file accesses by the user.</td>
</tr>
</tbody>
</table>

The Agent table has these columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentId</td>
<td>An internal reference used by a database as a PK for each row.</td>
</tr>
<tr>
<td>AgentName</td>
<td>Maps to the Name attribute of the logical Agent.</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>Maps to the Phone number attribute of the logical Agent.</td>
</tr>
</tbody>
</table>
7.1.3 Relating the logical to the physical

The first thing to notice about the physical model, is that it is not entirely clear how it encodes all the data in the logical model of the dataset. There are types of data which are missing when compared to the logical model, and there are items of data which do not appear in the logical model at all. Physical models are usually much less clear about the essential underlying structure of the data, as they focus on how particular data has been encoded using particular technologies.

However, both models are in fact mostly equivalent. In order to show this, we need to describe how this physical model encodes the data in the logical model. We will do this by analysing each entity in the logical model and showing how it is represented by the physical model.

i) Department

It is clear that the physical model does not include a Department as a separate entity at all. The closest mapping we can find is a DeptCode attribute on the User entity, but no corresponding name of the department. In this case, the list of names of departments is external data, to which the departmental code is serving as an external reference. Possibly this data is printed on paper, or exists on the organisation’s intranet.

The logical model makes it clear that knowing which department employs a user is part of the dataset. Hence, the dataset must include this ‘external data’, even though it does not appear in the diagram of the physical model. If the physical model were genuinely complete, it should also include this information. However, it is common to find that external data like this is not explicitly included in physical models, as people tend to focus on documenting particular systems and technologies, rather than mapping to logical models of complete datasets. The absence of this information will negatively impact the continuity of the dataset. Special care must be taken to ensure that all data which is logical part of a dataset is captured in the physical models, regardless of which technologies it is managed in.

ii) User

The logical User includes two attributes: Name and Payroll number which are both obviously represented in the User table. However, the physical user table also includes additional attributes whose meaning is not so immediately clear. As described above, the DeptCode attribute is a way to model the relationship between a User and a Department, in this case using a code as a reference to external data.
The UserId column is an internal reference for a User. Since nothing appears to relate to a user by this reference, it does not perform any function in the dataset, and does not seem to have any use in the physical model either. In practice, many databases include a primary key column for each table, even where one is not actually required.

The other attributes appearing in the physical User table relate to other logical entities, and will be described in the sections below.

iii) Contractor and Internal Staff
We do not see any reference to an Internal Staff or Contractor entity in the physical model at all. These entities are not directly modelled, and their existence can only be inferred with knowledge of the logical model the physical model is encoding. However, the physical model is still capable of representing Internal Staff and Contractors.

The AgentId column in the User table is an internal reference pointing to an Agent. We can see that the AgentId column can contain Null values (it does not have an Not Null flag in the physical model). This column is serving two purposes: it is being used to flag whether a User is Internal Staff or a Contractor, and, if the User is a Contractor, to point to their Agent. If the AgentId is Null (has no value), then the User is Internal Staff. If the AgentId contains a value, then the User is a Contractor, and it is an internal reference to their Agent.

It is quite common for data in physical models to do double-duty, and for entities in logical models to be represented implicitly, by the presence or absence of other data in the physical model. This can make it hard to determine exactly what the physical model is modelling, without a description and mapping to its logical model.

iv) Agent
The Agent table represents the logical Agent quite naturally. Both of the main logical attributes are present in the Agent table (Name and Phone number). The only addition in the physical model is the AgentId column, which is the primary key of the table, and which forms the internal reference by which a Contractor is related to an Agent.

v) Access and File
These logical entities are not so neatly represented in the physical model as the other entities. We see that there is a LogFileName attribute on the User table in the physical model. The attribute has an explanatory note below it, which states that it contains the file path of a log file, which contains details of file accesses by a User. Hence, the LogFileName attribute is an internal location-based reference to external data which contains the actual file access
information for that User. The LogFileName attribute is not part of the dataset – it is merely a physical method for encoding a relationship to other data which is part of the dataset.

The note describes the encoding and structure of the log file, which is a text file giving a list of files accessed, and the date/time on which each file was accessed. The log file contains the full path and name of each file accessed by a User. Hence, this allows us to reconstruct the parent folders of each logical File by examining its path.

However, there does not appear to be any count attribute of the files being accessed, or any Access entity in which it belongs. The physical model forces us to calculate the count of the files being accessed, by reading the entire log file and adding up each access listed in it. Therefore, the count attribute is calculated data, and the Access entity only logically exists in software, when someone wants to know how many times each file has been accessed by a User. It has no physical representation in the data model at all, even though it is possibly the most crucial piece of information in the logical dataset, as this is the purpose of the dataset.

Finally, the date/time on which a file access took place is not part of the logical dataset, but is recorded in the physical model. This may simply be an artefact of the technologies used to produce the log file.