Harmonised Data Model
Policy and Guidelines for Incremental Update

Version: January 2002

Introduction

Incremental update is a component of the model developed for topographic, cadastral, nomenclature and street address data compliant with the Australian Spatial Data Infrastructure (ASDI).

For the purposes of the ICSM sponsored data for the ASDI, incremental update is defined as:

> *A data transfer system that enables suppliers to supply only information that has been affected by change between two versions of a dataset and for users to be able to incorporate those changes into their data.*

This document includes the policy for incremental update to be adopted for ICSM-sponsored datasets, and guidelines as to how the policy might be applied. The approach that has been adopted is one of setting minimal standards because there are several different approaches to incremental update already implemented by different jurisdictions and because data users as well as suppliers also participate in the incremental update process. The users may need to modify their approach to suit their individual circumstances.

Background

Users are increasingly holding topographic and cadastral data online. Frequently they are merging these data with data from other sources or attaching their own attributes. The limited amount of change for these data, the move towards updating the data by themes rather than by tiles, and the relatively large files needed for transfer all point to the need to update the data without re-supplying unchanged features. Incremental update is designed to minimise the amount of data that needs to be supplied to bring users' datasets up-to-date.

ICSM's Topographic and Cadastral Working Groups are building basic requirements of an incremental update system into the design for the ASDI topographic and cadastral data layers. An incremental update system can be seen to have three components: the relevant data fields, the format for data supply and the procedures to incorporate changes into a user's dataset. These guidelines are intended to explain the logic of data fields that have been included and indicate their use.

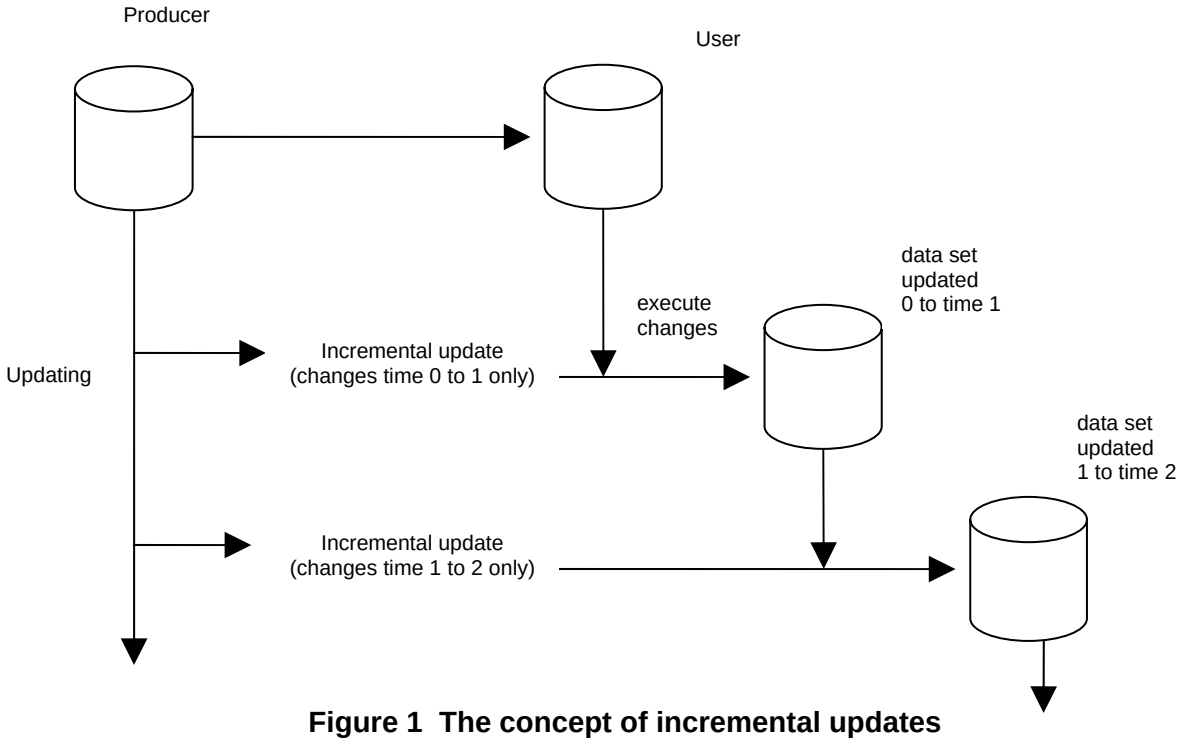

Figure 1 The concept of incremental updates

This document takes the form of guidelines. The actual implementation of incremental update may vary slightly from jurisdiction to jurisdiction depending on the formats each supports. Also the procedures for incorporating the incremental update files will vary depending on the user's implementation.

Policy for Incremental Update

The Working Groups have established a policy for the incremental updates. The policy is:

1. Ability to provide increment updates will be optional until 31 December 2002 for the purposes of ASDI compliance. From 1 January 2003, provision for incremental updates will be mandatory for ASDI compliance.

2. Minimum provision for incremental updates for suppliers will be:

    i. Population of datasets with Persistent Identifier (PID)
    Creation Date
    Retirement Date

    ii. Documentation of method used for supply of incremental updates.

3. The Persistent Identifier will be unique to a real world feature within an ASDI compliant dataset. A real world feature is taken as a feature that can be represented by a single primitive or by a single complex object.

4. Business rules will be established within jurisdictions to document the rules for creation and retirement of Persistent Identifiers. The purpose of such rules will be to preserve the Persistent Identifiers between temporal database representations of the feature.

5. The combination of Persistent Identifier and Creation Date and of Persistent Identifier and Retirement Date will be unique within an ASDI compliant dataset.

6. Jurisdictions will maintain records of retired objects.

7. When a real world feature changes and the Persistent Identifier can be maintained, the object will be retired and a new object with the same Persistent Identifier and a new Creation Date will be created.

8. In instances where a real world feature is stored and current *more than once* in the same dataset for different purposes (for example, a boundary line that is common to both a suburb boundary and a postcode boundary), each occurrence must have a different Persistent Identifiers. That is, each occurrence will be regarded as being a different feature within the database.

Overview of Incremental Updates

The system described here assumes that features will not be deleted from the supplier's database. Instead feature instances will be retired. A feature instance is the representation of a feature in the database relating to the period between its creation date and its retirement date. Note: there may be multiple representations of a feature in the database but only one will be valid at any point in time. All the feature instances relating to a particular feature can be seen as an audit trail of the changes that have occurred to a feature over time.

The data that forms the current representation of reality in a database is referred to here as the *active data*.

Figure 2 shows how an audit trail of changes to a feature is built in the database. The feature is first created in January 1997. A change is made in January 1998. At this time the original feature instance is retired and a new instance is created. A second change is made in January 1999. In 1999 there are three feature instances in the database. Only the instance shown in the bold box is active as it is the instance with no retirement date. Should the feature later be removed from the active data a retirement date would be added to the current active instance and no new instance created.

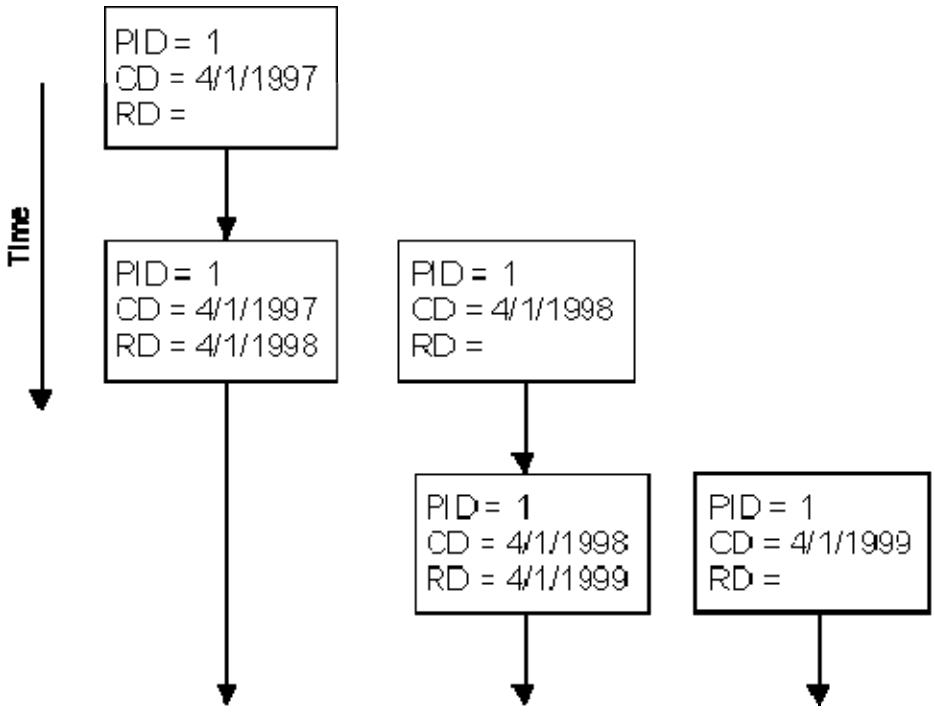

Figure 2, The audit trail of changes to a feature
PID is persistent identifier, CD is creation date and RD is retirement date

Guidelines for Relevant Fields

Persistent Identifier

The Persistent Identifier (PID) serves as an identifier for the feature in both the supplier's and the user's database. To do this adequately, it must be unique within the dataset. In addition, the business rules for the type of changes that apply must be clear. The intention with the PID is that the same string be used to identify a feature as long as possible. For as long as the PID stays the same it is relatively easy to trace the history of a feature and minor changes can be accommodated in users' database with the minimum disruption to any other attributes that may have been attached.

Recommended business rules for the PID are as follows.

- A PID will be attached to either a primitive or complex feature (see figure 3).
- Where a PID is attached to a complex the whole complex will be treated as a single feature in the database. A change to any primitive will result in all primitives that make up the complex feature automatically being also treated as if changed.
- The PID will be maintained when a feature's attributes change.

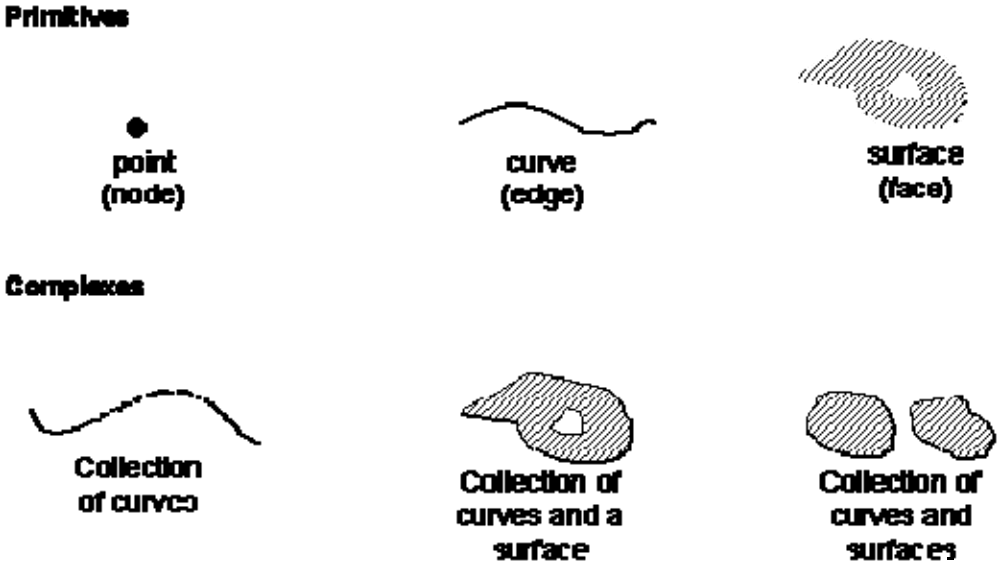

Figure 3 Primitive and complex features

- The PID will only be retired through changes when this is unavoidable. For example, when a linear feature is split into two features or when two features are merged. The PID will be maintained when the spatial representation of the feature changes but logically the feature is the same. For example:
    - for a linear feature if the start node and end node are the same,
    - if the real world feature is unchanged but the database representation changes.

Creation Date
Creation date will be the date the feature instance is created in the database. In general, it will have no relation to the date on which the feature physically came into existence (for example, the date of completion of a building). Typically the creation date will be when the feature instance is captured or when the feature instance is committed to the database. Jurisdictions may set other business rules for allocation of creation dates.

Creation dates must:

- be populated for all features
- accurately reflect the chronology of changes to the database, that is if more than one change is possible within a day time must be included in the date.

Retirement Date
Retirement date will be the date the feature instance is retired. A feature will be retired when it no longer exists in the real world, is being replaced by a new feature instance due to change to the feature, or otherwise is no longer required in the active data. In general, it will have no relation to the date on which the feature was physically destroyed (for example, the date of demolition of a building). Typically the retirement date will be when the feature is marked as retired or when the feature's retirement is committed in the database. Jurisdictions may set other business rules for allocation of retirement dates.

Retirement dates must:

- be populated for all features other than active features
- be null for active features (Note: there must only be one active feature instance in the data at any time for any given feature
- accurately reflect the chronology of changes to the data base, that is if more than one change is possible within a day time must be included in the date

Creation of an Incremental Update File

Guidelines
The supplier should develop and publish business rules to standardise the timing of the creation of files for both full supply and incremental updates. The rules should be designed to prevent changes being lost due to production or timing considerations. For example, committing changes to the supplier's database might occur at close of business and the generation of distribution files be timed to occur after this time.

To produce an incremental update file the supplier will need to know the time span that the file needs to cover. The supplier then needs to extract from the database all feature instances where either the creation date or the retirement date falls within the time span for the incremental update.

One incremental update file will need to be produced for each dataset that has undergone change.

The incremental file will include the PID, creation and retirement dates for all features.

Ideally the file will contain only those feature instances that need to be deleted from the users' datasets and those than need to be included. However, the full sequence of changes may be included. Documenting which of these options is used will be critical. If the full sequence is provided the users will need to build systems that cope with a feature instance or a feature being created and retired within the period covered by the file.

The supplier will determine the format for the incremental update file. It may be a standard transfer format (for example, ARCINFO Export or VPF) or a format specifically developed for incremental updates.

Comments

There are a number of possible variations on the provision of incremental updates depending on the individual jurisdiction's systems. From the ASDI users' perspective it would be desirable that all suppliers adhere to one system. It is possible that this will occur over time as a best practice develops. However, systems are already in place in some jurisdictions and it would require significant investment to change them so variation will be inevitable.

It is critical that jurisdictions adequately document their implementation of incremental up-dates and that the approach is not subject to constant change. This allows the users to develop procedures for importing the data.

At its simplest an incremental update file will include the spatial and attribute objects for features being deleted as well as for those that are being created. The process is more efficient if only the necessary information (PID, creation date and retirement date) is supplied for those features that are to be deleted.

Variations from these guidelines may include:

- Supply of additional information, for example, the PID of the previous feature if a new feature is created by splitting a previous feature
- Supply of a flag for features to be deleted and features to be added in lieu of the creation and retirement dates
- A single file that includes a field which identifies the layer the feature comes from.
- The production of incremental updates to cover fixed time frames. The guidelines assume that incremental updates are custom produced for each user

Incorporation of an Incremental Update File into a User's Database

Guidelines

Users should plan for the incorporation of incremental updates into their data base from the time of original data supply.

It is critical that the users maintain the PID in their dataset and preferable that the creation date is held. If multiple ASDI datasets are stored in the one database then the ANZLIC dataset identifier needs to be stored against features. These guidelines assume that the user only wants to maintain an up-to-date copy of the data so holding the retirement date field is not necessary.

Should users be adding their own attribute data into the database or changing the structure of the data, they need to develop a path to reapply their changes to the data from the incremental update.

The user needs to know the date of the last supply of data. This will be required when ordering a new incremental up date file.

At its simplest applying an incremental update in the user's database would consist of two steps. Firstly deleting from the database the features that have a matched PID and a populated retirement date in the incremental update file. Secondly inserting the spatial and attribute objects for features that have a null retirement date into the database.

A user may need additional processes in the procedure for incorporating incremental updates. This will be dependent upon the changes that the user has incorporated into the database. For example, including data in more than one ANZLIC dataset into one layer would mean that the user would also need to match ANZLIC dataset identifiers as well as PID values.

Storage of the creation date as well as the PID against features allows for the possibility of an audit of the currency of the features.

Comments
Users modifications to the data may take many forms and the data may be stored in a wide variety of systems. So each user will need to determine their own method for implementing incremental updates. This will need to take into account not only their own system but also the actual system being used by the supplier. For incremental updates to work it is critical that the users plan for incremental update at the time they are designing their implementation of the data. It is equally critical that the users have access to good documentation as to what their suppliers' systems are going to be.

References

Department of Primary Industries, Water and Environment, Tasmania, *Business Rules - the LIST, Unique Feature Identifier, Feature Metadata, Creation and Retirement dates*, draft as at 25th August 1999.

ISO/ DIS 19107 *Geographic information – Spatial Schema*, 2001 document N 1096

Appendices

Definitions

ASDI compliant	Data that meets all the requirements for inclusion in the Australian Spatial Data Infrastructure
Attribute object	An object that defines the attributes of a feature
Business rules	A set of rules which standardise and define procedures or outcomes, and are drawn up or adopted by an organisation.
Complex object	A collection of primitive objects that together define a real world feature (see figure 3) (strictly a geometric complex).
Create	To add a new feature to the dataset
Creation date	The date (including time if necessary) that the feature was first added to the dataset
Dataset	An identifiable collection of data (ISO/TC 211)
Delete	To remove an object from the dataset without trace
Incremental update	A data transfer system that enables suppliers to supply only information that has been affected by change between two versions of a dataset and for users to be able to incorporate those changes into their data.
Object	A representation of information in data (strictly 'entity with well defined boundary and identity that encapsulates state and behaviour' ISO TC211)
Persistent identifier	A unique number that identifies a real world feature.
PID	see Persistent identifier
Primitive object	A point, curve or surface (strictly a geometric primitive - 'object representing a single, connected, homogenous element of geometry' ISO TC211)
Real world feature	A feature that exists in reality and can be represented by a single primitive or by a single complex object
Retire	To flag that an object is no longer current in the dataset
Retirement date	The date (including time if necessary) that an object was identified as being no longer current in the dataset.
Spatial object	An object that defines the spatial dimensions of a feature
Suppliers	Organisations responsible for distribution of data either as suppliers and maintainers or as custodians of the data
Users	Organisations and individuals holding and using copies of datasets generated and maintained by others