Automated Conflation Policy:
A Petroleum Industry Dilemma

by

Norman J. Berls

Introduction

For a data process to be valid, all the location information involved must be expressed in the same CRS (Coordinate Reference System). This is to say that the locations must be conflated. This requirement means locations may have to be transformed after they are retrieved from a database but before they are delivered into a data process. The difference between the storage CRS of the data and the operational CRS of the process may involve a difference in geodetic datum. Therein originates a petroleum industry dilemma.

For any given geodetic datum, several different transformation methods and parameter sets may be available. Somebody or something has to choose which one to use. For example, for the Adindan geodetic datum (EPSG CRS 4201), the EPSG lists seven different transformation instances (Codes 1100 - 1106). While a geodesist will know which one is appropriate to use, a computer program will not... unless the geodesist tells it. In this day of very large databases where a project may contain thousands of data parcels, all available for immediate use, the choice of a geodetic datum transform instance must be automated

Conflation Policy

The choice about which geodetic transformation method and parameters to use can be contained in a data bundle called a “conflation policy”. At a minimum, a conflation policy must contain:

1) A “from” datum designation.
2) A “to” datum designation (WGS84 at a minimum)
3) A transformation method and parameters sufficient to transform locations from the “from” datum onto the “to” datum.

The third item in the list guarantees conflatability. The minimum list can be called a “half-transform” in that it provides a translation to/from WGS84. Thus, if such a half-transform conflation policy exists for every other extant datum, conflation can always be accomplished by using WGS84 as an intermediary. This is an important consideration as software users do not want to be constantly badgered with demands for additional information. Indeed, they will not accept and will not purchase a major system that does not include a guarantee of automatic conflation. Finally a sophisticated conflation policy implementation may also include an extended list of alternate “to” datums with transformation methods and parameters. Such transforms may be either

“direct” or “compound” transforms. A “direct” transform translates locations directly from the “from” datum to the “to” datum without using WGS84 as an intermediary. A “compound” transform does the same transformation using several steps instead of just one.

This list, if it exists, carries a higher priority than the half-transform. This is to say that during usage, a conflation policy is first inspected for an extended transform and, only if such is not available, will the half-transform be invoked. In any event and in all cases, the extended transform list can contain only one instance of a particular “to” datum. If it contained more than one, it would create an ambiguous situation for software and that is what a conflation policy is designed to remedy.

An Inconvenient Software Requirement

While there are a great many requirements the petroleum industry has for it’s software, one that impacts conflation policy is the requirement that software be immediately executable following installation. Consumers will not buy software that does not meet this requirement. The toleration for preliminary configuration tasks is very low.

In the case of conflation policy, there is no way software can blindly make a reasonable choice of conflation policies nor, is there any way for a geodesist working for a software company to prescribe conflation policies that will be appropriate for completely arbitrary situations. Only the geodesist managing the project at the client site is in any position to know which transforms might be appropriate. The task for software designers becomes one of configuring the software to pry that information out of the client’s geodesist with minimal perceivable delay.

Possible Implementation Strategies

Any implementation must make some sort of default conflation policy immediately available (no matter how inappropriate) or, it must, in very short order, compel the user to make a choice of conflation policy from a comprehensive list. The extended transform list that was previously mentioned does not fall into this category. Indeed, given the number of possible permutations, a comprehensive list that included extended transforms in all their myriad possible combinations, would run to millions of alternatives (most of them useless). The addition of extended datum transforms can and should be left for users to do at some post-installation date.

One possible alternative implementation can be called an ‘early binding’ system. This implementation requires that a conflation policy be attached to each and every data parcel. This attachment is done at load time by the person loading the data to the database. In practice this action becomes part of the mundane process of attaching a CRS to the data parcel. Most users have grown accustomed to this and accept it as a normal and unavoidable part of the overhead involved in loading data. This implementation requires the geodesist to decide which conflation policies are appropriate and communicate these decisions to the data loader. A really prudent geodesist will need to check periodically to see that the correct conflation policy is being attached to the appropriate data parcel. In one sense this is an imposition on the geodesist’s time. In another sense it means that users are forced to deal with the issue of conflation policy and geodesy very early in a project. As we shall see, this is a good thing because as a project progresses it will

become more and more costly to remedy geodetic errors. Later, when data parcels are fetched into a computational processes, the attached conflation policy is applied on an as-needed basis. The description “early binding” derives from the fact that the choice of which geodetic datum transform and parameters to apply is made early in the life history of a data parcel (i.e. when it is loaded).

Another possible alternative implementation can be called a “late binding” system. This implementation requires no geodetic action at load time other than the association of a CRS with the data parcel. Of course, the CRS contains a geodetic datum reference. Later, when data is fetched into a computational process, a table of designated conflation policies is examined. This table will contain a single conflation policy for each and every possible “from” geodetic datum. These policies are initially set on an arbitrary basis by the software vendor. This guarantees that the system will be executable immediately following installation. However, if the system is to perform conflation in a sane and appropriate manner, a geodesist is going to have to intervene and make sure that relevant conflation policies are reset to an appropriate state prior to the processing of any data. If he does not do this, the system will still run but will produce arbitrary results. The description “late binding” derives from the fact that the choice of which geodetic datum transform and parameters to apply is delayed until the last possible moment when data is fetched into a computational process.

In the end, regardless of implementation, a geodesist has to intervene early in the life cycle of a project to make sure that appropriate conflation policies are defined and implemented. Computers and software designers simply do not have enough information to make these decisions intelligently.

Some Non-Obvious Realities

Modern petroleum industry software works by having the software installed as a single instantiated instance. That instance can then have multiple projects defined under it. Of course there can be multiple installation instances and, this paper refers to these kinds of circumstances as being “multiple databases”. In fact, a “database” may be an installation of a completely different software package.

Under an early binding system, the reservoir of conflation policies is initially created by the software vendor and contains one copy of every known half-transform. If each project were to have its own private reservoir of conflation policies, it would mean a high degree of duplication of these initial loadings. In fact, a single reservoir can serve an indefinite number of projects with no disadvantage to the user. Thus an early binding system needs only a single conflation policy file implemented at the “global” level. This global resource is then shared and used by all projects under the installation.

Under a late binding system, the reservoir of conflation policies is initially created by the software vendor and contains one arbitrarily chosen conflation policy for each “from” geodetic datum. There cannot be more than one because then the software is forced into an ambiguous situation from which it cannot recover. Thus if one project requires a certain conflation policy for DatumA and another project requires a different conflation policy for DatumA, a single reservoir of

conflation policies will not work. A late binding implementation requires that a separate conflation policy file exist for each project. Such files are said to be “local” in character.

Early binding implementations should be implemented globally and, late binding implementations must be implemented locally.

Change Management

If conflation policy must be changed under an early binding system, it can only be accomplished by searching out every data parcel that references the conflation policy and substituting a different reference there. Changes in conflation policy under early binding implementations are very work intensive and verge on the impossible. This must be considered as the primary drawback to early binding systems.

If conflation policy must be changed under a late binding system, all that is required is that the one and only conflation policy instance for a “from” datum be located and the suite of transform methods and parameters recorded under it be changed. This means that a global reservoir of every known transformation method and parameters be kept for users to select from even though this reservoir is never called upon during actual conflation. Of course users must have a tool that allows them to add to this global reservoir. In the end, a change in conflation policy under a late binding system is a very easy thing to do. This must be considered as the primary advantage of late binding systems.

Early Binding Summarized

Early binding implementations of conflation policy have several distinct advantages:

1) They are very stable in the long term.
2) They force users to think about geodesy early and to make decisions early.
3) Personnel changes are not a problem because the system needs little or no modification.

Early binding also suffers from several distinct disadvantages:

1) A geodesist must inform data loaders of conflation policy before any data is loaded.
2) A geodesist should double check that the data loaders are setting conflation policy correctly.
3) A geodesist has only one good chance to get conflation policy right because of the difficulty involved in changing policy.
4) It is possible to tag different parcels, on the same datum, with different policies... leading to location state corruption.

Late Binding Summarized

Late binding implementations have one primary advantage: they allow conflation policy to be changed easily.

Late binding implementations of conflation policy also suffer from several distinct disadvantages:

1) A non-arbitrary initial default state is impossible (viz. “An Inconvenient Software Requirement”)
2) Users may ignore geodesy entirely and the system will still run.
3) Projects can be corrupted by changing conflation policy in mid-life.
4) Data parcel parent-child relationships must be known in exact detail to avoid location corruption following a policy change. A geodesist must know the project history accurately and in detail to safely change conflation policy. This makes personnel changes a problem.

Location Data Corruption

All petroleum industry data is not equal. Some of it is field data and some of it is created during computational processes (i.e. it is *derived* from field data). The process of deriving data may well involve conflation and conflation policies. Any derived data parcel has inherited location information from its parent data and, this inheritance process was potentially governed by a conflation policy.

For example, the creation of a grid may involve sundry input data including:

- Scatter data (e.g. well markers or 2D seismic interpretations).
- Fault data
- Polygon data
- Other grids

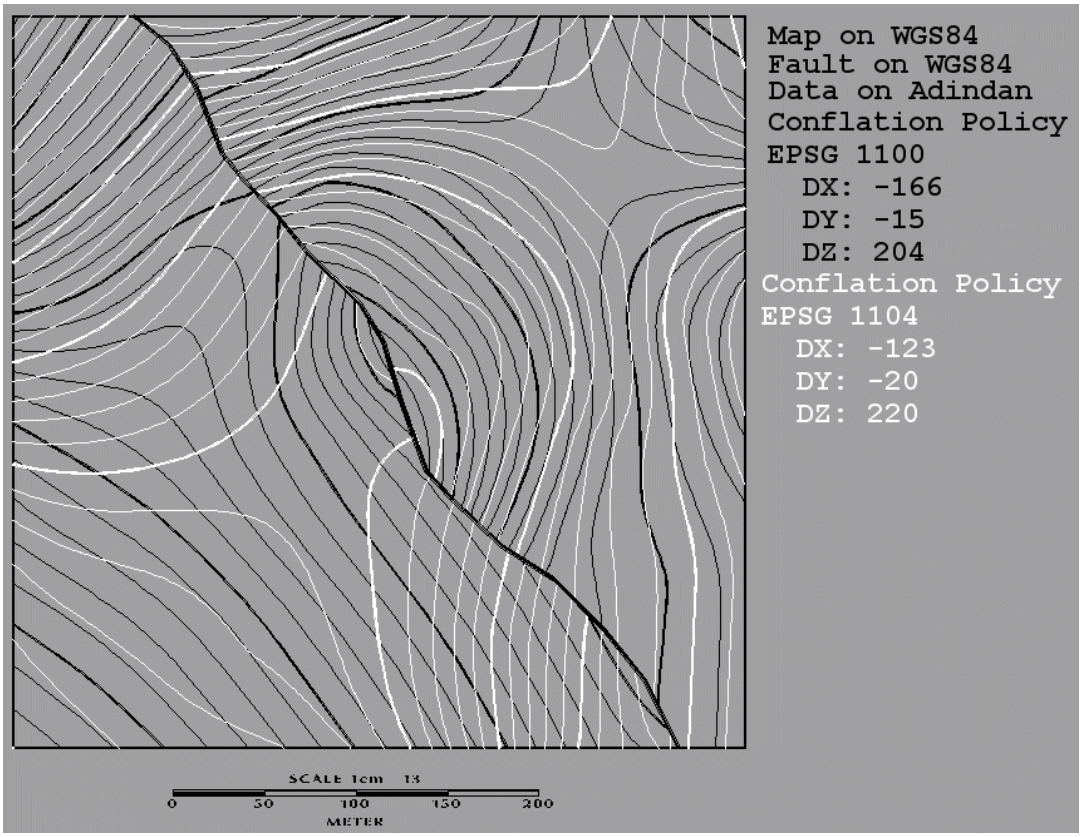
Any one or all of the input data parcels may have required the application of a conflation policy before they could be used to compute the grid. If that conflation policy is changed at a later date, the grid that was a consequence of that conflation policy should be regenerated to maintain the project in a state of spatial consistency.

Let us consider an example. Given:

1) A set of scatter data measured on the Adindan geodetic datum.
2) A fault trace measured on the WGS84 geodetic datum.

It is possible to create and contour a grid based on the WGS84 geodetic datum. In so doing a conflation policy must be invoked for the Adindan data. EPSG mentions two different molodensky transforms identified as 1100 as follows: DX: -166, DY: -15; DZ: 204, and 1104 as follows: DX: -123, DY: -20, DZ: 220.

The resulting grids and contours appear as follows:



When the Adindan data was conflated with transform 1100, the black contours resulted. When it was conflated with transform 1104, the white contours resulted. The difference in conflation policies caused a scatter point near the center of the map to shift from one side of the fault to the other when the data was conflated onto WGS84. When conflation policies are changed and derived data objects are not regenerated, this is the kind of location state corruption that can result.

The exact same argument applies to maps.

The inheritance of location data may not be so simple and direct as in the case of grids and maps. The interpretation of 3D seismic data involves the following inheritance chain:

- Spud location of a well
- Directional Survey

- Computed well trajectory
- Checkshot
- Synthetic Seismogram
- 3D Interpretation

The spud location may have to be conflated into the CRS of the 3D Seismic data for this chain of events to be carried out correctly. If, at a later date, the conflation policy is changed, then the entire sequence of processing steps should be redone if the project is to be maintained in a state of spatial consistency.

The requirement for keeping track of what is generated from what, when and which conflation policies were involved amounts to an extreme intellectual challenge. A geodesist must meet this challenge if he is to safely change a conflation policy. The difficulty involved in doing this makes the process of changing conflation policies in project mid-life a high risk affair. These circumstances mean that even if a software package supports changes in conflation policy, the risks and costs of invoking such a facility reduces the likelihood of it actually being used to a low order. Management is unlikely to allow the undermining of man-years of invested time.

It has been suggested that at least the intellectual challenge can be reduced by the automated keeping of inheritance relationships. In the case of grids, it is easy to see how this could snowball into a very extensive undertaking. Indeed, the growth of the records would become geometric as grids were used to create grids. Then too, it must be conceded, that many data processing decisions take place in the mind of a geophysicist or geologist. Since there is no electronic record of what goes into these decisions, there is no way for a computer to keep fully accurate records of what actually influenced the creation of what. While the keeping of partial records is possible, the keeping of fully accurate and complete inheritance relationships involves so much work of indefinite extent that such an undertaking is impractical.

Implementation Strategy Reconsidered

Thus far we have discussed only the possibility of strictly early binding or strictly late binding systems. There is however, a third possibility: hybrid systems.

A hybrid system would use an early binding strategy as it's primary default and control mechanism for administering conflation policy. It would also use a supplementary system of late binding overrides to the early binding rules. The reservoir of late binding rules would be empty at the beginning of a project. It would only be loaded with rules in the case of dire necessity. The hybrid system would work by examining the early binding rules first. This would guarantee conflatability no matter what. Then the late binding rules would be inspected to see if any available rule applied to the situation. If one was found, it would override the early binding rule.

Thus we must evaluate three possible implementations:

1) Early binding only
2) Late binding only
3) Hybrid

Objectivity

The author has only actually seen early binding systems in operation. The merits or demerits of late binding systems must be estimated from general software experience. This means that some degree of subjectivity is going to end up in this evaluation. It seems appropriate to make at least some formal attempt at imposing as much objectivity as possible. Towards that end, two criteria see common usage among software developers:

1) Usability - How much work does it take to get the software to produce usable output?
2) Risk - What is the probability that users will inadvertently err in using the software?

These criteria are also time dependent. This is to say that different considerations may be involved during the short term vs. the long term but, both must be considered if the evaluation is to be complete.

Short term usability involves questions like:

- How much work must a geodesist do at the outset of a project?
- How much work must geodetic technicians do during data loading?

Long term usability involves questions like:

- How much work is involved if conflation policy must change?
- How much work (learning curve) faces a new geodetic manager?

A numerical rating system can be set up that reflects the answers to these questions:

0 = Unworkable
+1 = Excessive work required
+2 = Significant work required
+3 = Moderate work required
+4 = Minimal work required

Short term risk involves questions like:

- How likely is a geodesist to make an error?
- How likely is a geodetic technician to make an error?

Long term risk involves questions like:

- How much project history must a geodesist know to make a safe change in conflation policy?
- How many interdependencies are likely to develop between projects in time?

A numerical rating system can be set up that reflects the answers to these questions:

0 = Invulnerable
-1 = Unlikely source of trouble
-2 = Possibly a source of trouble
-3 = Probably a source of some trouble
-4 = Very likely a source of trouble

This methodology allows usability and risk to be combined into a single "net" measurement by adding the scores together. The higher the "net" score, the better the implementation will be deemed to be. Thus we are in a position to look at usage cases and try to determine how well each potential implementation solves the problem presented by the usage case.

Usage Case #1

The customer has a single project containing thousands of data parcels. These data parcels have been measured on a variety of different geodetic datums. The customer requires that data be properly conflated before it is used in a computational process.

Problem: How can be support and apply a single conflation policy for a given DatumA –> DatumB transform on thousands of distinct data parcels.

	Short Term Usability	Risk	Long Term Usability	Risk	Net
Early Binding	+2	-2	+2	-1	+1
Late Binding	+4	-2	+2	-4	0
Hybrid	+2	-2	+4	-2	+2

Tandem Usage Rules

In the course of common usage, petroleum industry projects are often not used by themselves. Often two or more projects will be accessed simultaneously by the computational process. This state of affairs is here being called "tandem usage." This complicates the administration of conflation policy and so, some ground rules are necessary to sort matters out. These rules can be called the "Tandem Usage Rules". They are:

When a single earth location is fetched from different data parcels and conflated, the associated XY's must be identical. It should not matter whether the parcels are associated with different projects. It should not matter whether the parcels are associated with different databases (i.e. software instances).

When an implementation involves late binding, either by itself or in hybrid, one project must be declared the "primary" project. All other projects become "secondary" projects. **Late binding policies on the primary project take precedence over late binding policies on the secondary projects.**

Usage Case #2

A customer has an old project. It has been around for 10 years and is used to manage the production from a field. The project involves only two geodetic datums. A marginally adequate Molodensky tranform has been selected to achieve conflation and, management cannot now afford a change in conflation policy.

In the same software instance, the customer wants to start out a new project covering the exact same area as the old project. The goal is to search for new production from deeper targets. Management has been persuaded that a newer Vector Rotation transform would achieve higher quality conflations.

Problem: How can support two different conflation policies, within the same software instance, for a given DatumA –> DatumB transform?

	Short Term Usability	Risk	Long Term Usability	Risk	Net
Early Binding	+2	-2	+2	-1	+1
Late Binding	+4	-2	+2	-4	0
Hybrid	+2	-2	+4	-2	+2

Usage Case #3

The customer has an old project that is 10 years old. It contains only wells. He wants to use it to support some older 3D seismic projects with an older conflation policy that cannot be changed. And he wants to use it to support a new 3D seismic project with a newer, better conflation policy. Both projects exist inside the same software instance.

Problem: How can we support two different conflation policies on a single secondary project that they use in common.

	Short Term Usability	Risk	Long Term Usability	Risk	Net
Early Binding	0	*	*	*	*
Late Binding	+4	-2	+2	-4	0
Hybrid	+2	-2	+4	-2	+2

The early binding policy for the well file can be set to support either the old project or the new project. It cannot support both simultaneously, hence the early binding implementation self destructs.

On the other hand, a late binding system will have one and only one appropriate policy in the primary project. It also may have none, in which case the conflation would have to be refused.

Usage Case #4

This usage case is the same as in #3 but, the projects are in two different software instances.

	Short Term Usability	Risk	Long Term Usability	Risk	Net
Early Binding	0	*	*	*	*
Late Binding	+4	-2	+2	-4	0

Hybrid	+2	-2	+4	-2	+2

The results turn out the same under Case #3.

Technical Conclusions

Conflation policy should be administered under hybrid systems that are combinations of early binding and late binding systems.

Basic default conflation should be administered by means of an early binding system because:

- It forces users to consider geodetic issues early and consult a geodesist early.
- It creates a stable system that requires little latter intervention by a geodesist.

Long term flexibility in conflation policy can best be achieved by a late binding supplementary system because:

- Conflation policy changes are easy to implement for the geodesist.
- Coordination of conflation policy during tandem usage becomes possible for the geodesist.
- Configuration as exceptions to the early binding policies limit content and hence cause less work for the geodesist.

A hybrid system has a susceptibility to location corruption following a conflation policy change like any late binding system. However this susceptibility is limited because the contents of the late binding file are limited. Indeed, the late binding file will be empty immediately after project creation.

The proper usage for a hybrid usage would be to put field data in secondary projects. As much as possible the creation of derived data objects should be restricted to primary projects. Then a change in conflation policy could be managed by the simple expedient of deleting the primary project.

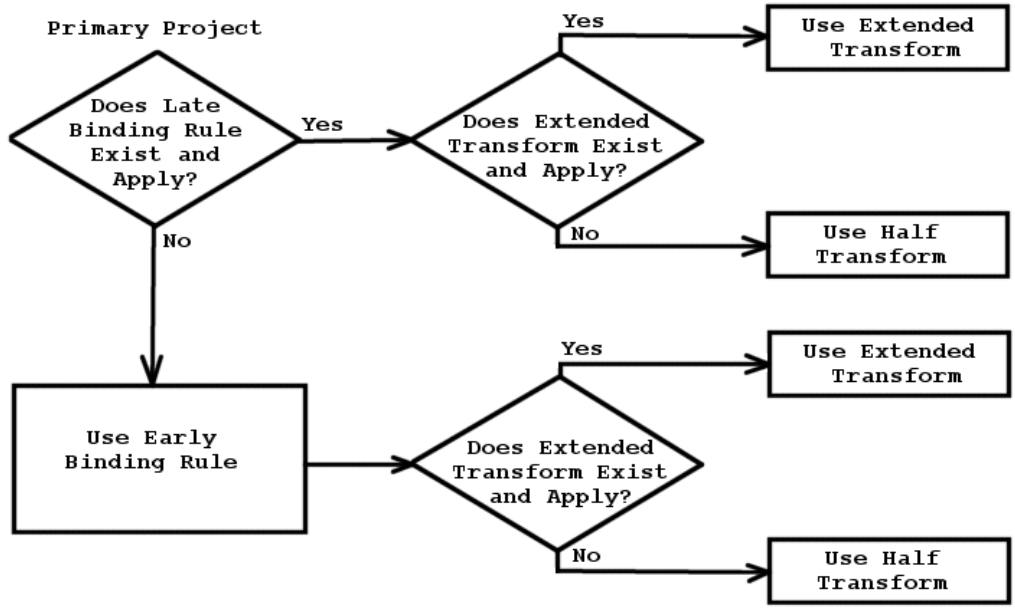
The danger in hybrid systems is that geodesists will eagerly seize upon the late binding file, make it the sole administrative focus for conflation policy and, neglect to regenerate derived data objects following a change in conflation policy.

Economic Conclusions

A “hybrid” system for managing conflation policy will be complex. It will include:

1. A global raw materials file containing all available half-transforms, direct transforms and compound transforms. This file is a resource file from which components are selected for the creation of both early binding and late binding conflation policies.
2. A global policy file with selectable early binding policies. Every data parcel must point at a default early binding policy.
3. A local policy file on each project containing late binding overrides. These files will start empty and then only have content added on an as-needed basis.

A fairly complicated set of rules and policies will be necessary to regulate such a system. Here the primary project may also be the only project.



Building such a system will require a significant programming effort.

The system will be challenging for users to understand and will require a significant education effort.

Submitted to the APSG