

XML in IDSS

With the release of IDSS for the 2007 reporting year, the Excel data (the original GSUB) format will no longer be used for the submission and storage of HEDIS data. In its place, NCQA will be using an XML-based format. This document is meant to address concerns you may have with this change.

This overview is divided broadly into two sections, each of which answers one of the following questions:

- *What is XML?*
- *Why is NCQA switching to XML over Excel?*

We'll begin by giving a short, very high-level introduction to XML.

What is XML?

XML stands for *eXtensible Markup Language*. Rather than being a specific language, it is more of a framework to make other languages. These “languages” are formats for describing and structuring data. Since HEDIS submissions are really just sets of structured data, they are well suited to being stored as XML documents.

The XML specification sets out some very general rules for what proper XML documents are. They are just normal text documents, so unlike with Microsoft Excel documents, you can actually open an XML file in Notepad (or any text editor) and be able to view and/or modify the information. They consist of the actual data *marked up* (hence the *M* in *XML*) with *XML tags*. These tags are what describe and structure the data. We will see an example of these tags shortly.

So long as one adheres to the basic strictures of XML, anyone can create his or her own XML-based format. That is what we have done with IDSS; we have created a format for storing data that, while remaining compatible with the worldwide standard that is XML, is custom-fit to suit HEDIS's needs specifically. Let's take a look at a brief example of that IDSS XML format now:

```
<?xml version="1.0" ?>
- <submission xmlns="http://www.ncqa.org/ns/2006/idss/hedis" gu
- <data>
  - <component type="GS">
    - <measures>
      - <measure id="cis">
        <reported>false</reported>
        <benefit>>true</benefit>
      - <data-elements>
        - <data-element id="myear">
          <value>2005</value>
        </data-element>
        - <data-element id="collmeth">
          <value>A</value>
        </data-element>
        - <data-element id="eligpop">
          <value>NR</value>
        </data-element>
        - <data-element id="eligdtp">
          <value>NR</value>
        </data-element>
      </data-elements>
    </measure>
  </measures>
</component>
</data>
</submission>
```

This is a short snippet out of an IDSS submission, as it looks when viewed with Internet Explorer. This particular section focuses on the Childhood Immunization Status measure, which is abbreviated “cis” in the XML. Even without knowing anything of XML, it should probably not take a long time to at least get a basic understanding of the structure and meaning of the data.

The XML tags in this sample are the elements that begin with “<” and end with “>”. They are not the data itself; they instead describe the data. For instance, it’s (relatively) clear that this snippet represents a HEDIS measure with an ID of “cis” (which is the ID for Childhood Immunization Status). Inside this measure is a set of data elements. The first of these data elements has an ID of “myear” (this is the Measurement Year) and a value of “2005” (*this* is the actual data). The next has an ID of “collmeth” (which is the Data Collection Methodology) and a value of “A.” Each data element ends at the ending </data-el ement> tag, while the entire measure ends with the ending </measure> tag (which is not visible in the image; it is further down, after all of the data elements of the measure).

This example serves to illustrate some of the properties both of XML in general and of our specific IDSS XML variant. XML requires that all of the tags start and end with angle brackets and that each starting tag (without the “/”) has a matching ending tag (with the “/”). Meanwhile, our own IDSS XML grammar defines exactly *which* tags (in this case, “measure”, “data-element”, etc.) are used and where they go. This means, among other things, that a general XML editor will be able to understand this document well enough to let it be displayed and edited, while we can build more specific tools (like IDSS) that understand our particular “dialect” of XML.

A full HEDIS submission looks very much like the snippet above except with more elements (possibly more than 90 “measure” tags and about 10,000 “data-element” tags).

Why is NCQA switching to XML over Excel?

A full treatment of the advantages of XML would take many pages. A lot of these, while certainly advantages over Excel, would be a little bit esoteric. Instead of a full listing, then, we will list only the most important advantages of XML over Excel, ones that address real problems with the Excel format as it relates to submitting HEDIS data.

1. XML is becoming a widely used format in all industries. It easily and inexpensively enables the transfer and sharing of data across technologies (e.g., operating systems, code languages, etc.) As an example, refer to how Health Level 7 (HL7) is publishing its standards.
2. **The Excel format requires using Microsoft Excel.**

The DST way of doing things is not the most efficient way, by any means. The DST has to be downloaded, filled in, sent to an auditor, sent back, perhaps sent to an auditor and back again, and uploaded back to NCQA. At all steps, the proper version of Microsoft Excel is required.

Because of the switch to XML, we are able to instead have the bulk of this process happen in a Web application. It is even possible to complete a HEDIS submission without ever transferring a file, since all entering and auditing of data *can* be done from within IDSS. But even if you choose to download an import template and upload it back, it is a much less painful process. Excel is not required, the auditing can still be done through the Web application, and initial trials suggest that the import template will be a fraction of the size of the DST.

As we break the dependency on Excel, we alleviate the problems that come from different versions of Excel. There have been difficulties in the past with getting the DST to work in all the versions of Excel used by companies that submit data to NCQA. Those version problems will always exist—even when we fix everything right for the current version of Microsoft Excel, the next version from Microsoft could break it again—unless we move to a non-proprietary format. XML is that format.

3. **Excel uses a coordinate-based system to locate data elements.**

One of the most time-consuming parts of DST maintenance is making the annual changes. The reason is cross-walking: since inserting a new data element means that the row coordinates of all of the data elements below it change, even handling data elements that were not modified in a given year requires a lot of work both for NCQA and for the companies that use the DST.

XML does not use coordinates. Thus, there is no crosswalk. Instead of being defined by their position on the import template, data elements in XML will be defined by their ID's. In the XML snippet that was shown a couple pages back, for example, the Data Collection Methodology element of the Childhood Immunization Status measure can be picked out because it has a data element ID of “collmeth” within the measure with the ID of “cis”. If a hundred new data elements were inserted in the next year of HEDIS, that same data element would *still* be identified by the ID's of “cis” and “collmeth.” That will never change, no matter how many other data elements are inserted or removed. Data-elements and measures use unique ids now. So instead of referring to an Excel coordinate, such as C:2347, one can refer to the id (e.g., measure “cis” data-element “myear.”)

4. **Excel is inflexible.**

Excel is a *binary* format, meaning that it's not readily human-readable. If you opened an Excel spreadsheet in Notepad, you wouldn't recognize much that was in there. Binary formats are particularly inflexible: they require particular applications read or write them, they are not easily convertible into other kinds of files, and difficulty in understanding them means that it's hard to work with them.

XML, on the other hand, is a *text* format. From the example above, it's clear that it's understandable even if you don't know much about XML. XML is widely understandable, in fact: there are literally thousands of editors, software libraries, and other applications that can work with XML. (In fact, IDSS is, at its core, just an XML tool.)

XML files are quite easily convertible to other types. This is what IDSS does; XML data can be converted into Web pages such as in IDSS, where Excel is exceedingly difficult to do this with. But they can also be converted to other types of files, like PDF, plain text, programming language scripts, or even Excel spreadsheets themselves. Thus, even if IDSS doesn't provide you with exactly what you need, a little knowledge of an XML transformation language is all it takes to turn your data into whatever you need.

It's interesting to note that there *is* a way to get some level of flexibility out of Excel, but that's by using the ability to save Excel documents in XML that Microsoft has added in recent versions. Even so, this is a very verbose and general XML dialect that is not well suited for HEDIS data.

5. **Excel is limited.**

Excel is meant to be one thing: a spreadsheet. XML can be many things. IDSS currently uses four different XML dialects internally (note that you will not have to worry about any more than the one being used for your data). Each of them performs a particular job in exactly the way we need it. This is what allows us to make IDSS a Web application. Excel cannot do this.

The extensibility of XML allows us to do better and faster validation, have a cleaner and more intuitive interface, and be able to adapt to changing needs better. At this point, XML isn't being chosen because it's better for this purpose than Excel is; it's being chosen because it's simply not possible to create IDSS with Excel as its data format.

6. **Excel is not the right tool for the job.**

One could drive a screw with a hammer, but it wouldn't work as well as it would with a screwdriver. The same could be said for collecting data with Excel: it can certainly be done, but there are better tools. As was mentioned, Excel is a spreadsheet and is best suited to accounting and the similar. XML is a data structuring format and is best for collecting structured data (e.g., HEDIS data.)

It is XML's suitability for this sort of data over Excel's that makes all of the previous points possible. Once you have the right tool for the job, a whole new set of possibilities opens up. And so it is with IDSS; using the right tool, more things will work and work better than with the Excel-based DST.

Excel certainly works well for many applications, and certainly no one would want to program a spreadsheet package in an XML-based language. They each have their place. XML's place is with data like we will collect with IDSS.

Conclusion

Making a change as large as an entire data storage format is not something that is done lightly. There must be compelling reason to make such a change; just a small benefit is not enough to justify making a concerted effort for something better.

XML provides more than a small benefit. Its ubiquity, its simplicity, and its flexibility will bring huge benefits both to NCQA and to the managed care organizations that submit data to NCQA. These benefits far outweigh the initial effort necessary to support a new data format even now, and perhaps even more in the future. For instance, though these things are not likely to be implemented in the initial version of IDSS, there are a slew of other possibilities for added functionality once everyone adopts the XML format:

- Submission and validation submissions automatically, without using the user interface;
- Reuse of submission data across years;
- Importing or updating one measure at a time, rather than a full submission;
- Use of the same XML format for other products (e.g., Quality Compass) so that the same mapping that MCO's create for use with IDSS can be used with those products;
- Exporting of data in additional formats (PDF, plain text, etc.).

These are just ideas for the future at this point. There may well be dozens of other ideas that will make IDSS even better as it goes into the future, both for NCQA and for MCO's. All of these possibilities are enabled by the switch to XML.