DELTAXML
EXPERTS IN INFORMATION CHANGE

White Paper

Intelligent Diff for XML

Finding changes is only half the story

"A World of Difference"

As XML emerges as a mission-critical platform, with more business applications and processes reliant on the technology every day, there's a growing need for tools to manage the rapidly expanding volume of XML data and in particular, to manage the constant ebb and flow of change.

A solution is needed which will not only identify changes accurately but represent them in a way which is consistent with an organisation's needs and which will enable those changes to be processed easily.

In this paper we discuss how to achieve this goal, identifying change accurately and representing that change in an XML format that allows onward processing. And we will show how intelligent handling of changes to your XML data can open up new commercial opportunities.

DeltaXML provides the most comprehensive end-to-end system for change control of XML data, uniquely taking in XML, identifying changes, and outputting the delta in XML. Our solutions have redefined the benchmark for XML differencing and have been selected by teams at IBM, Boeing, LexisNexis, Thomson Reuters, Oxford University Press and over 200 other major users and class-leading companies worldwide.

Introduction

XML is now widely used for data exchange in organisations across the world. It's the accepted standard for managing the content of websites, for complex documentation systems, for publishing legal, financial, scientific and academic data, for developing software and in B2B data exchange. But as organisations increasingly rely on XML as a common language – bridging previously incompatible systems in a way which is independent of hardware, software and applications, and enabling levels of data exchange unthinkable two decades ago, so the volume of XML data has ballooned and with it the challenge of change.

A change control solution is needed which is flexible enough to accommodate the diversity of XML control requirements, which is easy to integrate into existing XML infrastructures and which has the performance and scalability to identify change in large amounts of data and growing XML file sizes.

Any solution also has to be open standards compliant to ensure it will be able to handle tomorrow's data as well as today's and it should be capable of representing changes in a way which is easy to process.

This paper presents an analysis of XML change control. Specifically, (i) the need for XML change control, (ii) why XML change control is different, (iii) how to evaluate XML change control tools and (iv) the DeltaXML solution.

Why is XML change control important?

Most corporate and institutional XML data is in constant flux. It may be text on a website, manufacturing data in an XML database or financial data feeds from a third-party supplier. Key to managing all that data is understanding what has changed. Of course, if your data never changes, change control is irrelevant.

- You might want to display changes made to critical data so they can be reviewed or stored as an audit trail.
- You might want to enable editorial processes by allowing editors to check changes made against the original copy. (The "track changes" features in products like Microsoft® Word are no longer available in an XML environment – you need a generic solution for all your corporate data.)
- You might want to manage data translation by identifying sections of text that need to be retranslated within updated versions of the master document.
- You might want to take advantage of change information to reduce the processing overhead. By pre-processing data feeds and exchanging only changed data, database repopulation costs can be reduced significantly.
- Likewise, you might want to cut your bandwidth requirements by transmitting only updates to data files or records. These can be merged with existing data records to keep local copies of the master data up-to-date.
- And, of course, now that you can filter out unchanged information, it's possible to make essential documentation easier to use, ensuring readers don't have to plough through reams of unchanged pages to get to the important new content.

Why is XML change control different?

XML was designed to store, carry and exchange all types of data in a way which is independent of any particular software or hardware. It does this by storing data as plain text in a highly structured, but standardised way.

It seems logical to assume, therefore, that finding the difference between two XML documents would be a straightforward matter of comparison, using one of the many differencing tools that have been available for years. This might be true of traditional, unstructured documents where a standard "diff" algorithm can be applied, but the structured nature of XML mark-up means that those standard solutions are hopelessly inadequate. The rules needed to identify changes within XML files are very different from those used to identify changes in unstructured text files.

There are proprietary differencing tools available which are tailored to particular proprietary document structures but these discard the key advantages of XML – openness, flexibility, standardisation – and can never offer a generic solution.

The following illustrations give some clue to understanding what constitutes a change to an XML document. When applied to XML, traditional string-based change control will identify a huge number of differences that are not significant in XML and should properly be ignored by an XML aware comparison. Identifying change correctly in an XML document or file, is not a trivial exercise.

For example, the following files are identical as far as XML is concerned:

Example 1 – No spaces or namespaces

```
<record xmlns="http://www.myco.com/records"
id="b123">
<name>Michael Brown</name><born>1984-03-
08</born><sex>M</sex>
</record>
```

Example 2 – With spaces and namespaces

```
<staff:record id="b123"
xmlns:staff="http://www.myco.com/records">
  <staff:name>Michael Brown</staff:name>
  <staff:born>1984-03-08</staff:born>
  <staff:sex>M</staff:sex>
</staff:record
```

The second file is different to the first in a number of ways, including the addition of white space, the order of attributes and the declaration of the XML namespace (shown as the default namespace in the first, and using a prefix in the second). None of these changes is significant as far as information in the two XML files is concerned; the two files are "XML-equal".

Other differences in XML are significant, but not as significant as a simple string comparison would suggest. The rules for determining whether these differences constitute a valid change will vary on a case-by-case basis. For example, if the order is important then there is a significant difference between the two files below, but if order is unimportant the only change is the two added elements.

Example 3 – Original records

```
<record id="b123">
<name>Michael Brown</name>
<born>1984-03-08</born>
<sex>M</sex>
</record>
<record id="b124">
<name>Gillian Bryan</name>
<born>1951-03-06</born>
<sex>F</sex>
</record>
```

Example 4 – Updated records

```
<record id="b124">
<employee-no>BR12</employee-no>
<name>Gillian Bryan</name>
<born>1951-03-06</born>
<sex>F</sex>
</record>
<record id="b123">
<employee-no>BR24</employee-no>
<name>Michael Brown</name>
<born>1984-03-08</born>
<sex>M</sex>
</record>
```

When the structure of the XML has changed even slightly, a text-based comparison will often report a huge change to the document. Identifying the minimal set of actual changes that have occurred requires the differencing tool to have an intimate knowledge of XML structures. For example, when a change is found in a sequence of elements, the rest of the data must be searched to identify the re-synchronization point, accounting for possible promotions and demotions within the structure.

While this sort of problem attracts a lot of interest among academics, the reality of XML differencing is that practical commercial solutions are essential. And that is no more true than in the application of an XML differencing solution.

A Holisitic Approach to Change

Most solutions available today focus on identifying optimal change – an abstract notion that lies somewhere between the capability of the tool and an ability to identify all changes. A commercially useful XML change control tool must do more and tackle the problem of change control holistically. It must not only be able to identify the specific level of change required in each application, but it must also make that change information available so that actions can be taken subsequently to use the data.

Most existing tools generate a visual representation that can be viewed in an XML editor. This is fine for users who are familiar with XML and who are comfortable with manipulating the XML itself.

Some other tools provide an annotated version of one of the source documents, with comments identifying the differences. This approach may be sufficient for some requirements, such as viewing changes in the XML, but it is unlikely to work where an operator needs to view changes in context in an XML authoring application and either approve or reject them. Nor is it useful where the delta is to form the

input for a further automated process, because comments are inherently difficult to process.

This approach may also be inefficient when that change information needs to be transmitted, as the whole file must be sent rather than just the change information. Sending delta files rather than retransmitting large amounts of the original data can significantly reduce bandwidth and storage requirements.

A holistic approach to change will also encompass the degree of granularity required in each case. For some applications it is not appropriate to identify changes at too low a level, such as a single field in a record or a single point in a graphical element: rather the system must ensure that the whole record or row is identified as modified, ensuring that a consistent set of related fields is updated at the same time.

For other scenarios, such as the human review of changes to text, the correct granularity will range from the individual characters or words to whole paragraphs. For machine interpretation of the results a minimum delta is most common. A purely academic approach to XML change will be concerned with generating the minimum change for all scenarios, regardless of whether that level of granularity is appropriate for the consumer.

Evaluating XML change control solutions

There are three essential criteria when evaluating an XML change control solution: accuracy, representation and usability.

▲ *Accuracy of the result*

It is imperative that the tool is smart enough to ignore differences that are not significant in XML and flexible enough to include or exclude those which are relevant to each particular application. While changes in the order of attributes, the use of whitespace or of namespace prefixes, for example, are generally not significant, changes to a time-stamp element may well be. Anyone who has spent time wading through false positives in a change report will understand the frustration, not to mention the considerable cost in terms of time wasted.

Other configurable behaviour should be the handling of white space (critical for some users, irrelevant for others), and how the tool handles combinations of ordered and unordered data. To ensure accurate alignment, it should be possible to add keys to the data to identify sections that must be matched, so that (for example) inserting a new data record into a configuration file does not cause misalignment of the comparison. It should be possible to add these keys during processing, stripping them from the output to keep the data pristine.

▲ *Representation of the result*

Change itself is rarely the end goal. In most cases the change information needs to be used either in a form suitable for visual review and approval of changes or for data pipelining or change-triggered processing. So it's important that the solution you use has the flexibility to allow the results to be used in the most efficient way for your organisation.

It's also important that the change results are not presented in a proprietary format, making onward processing difficult or costly. For example, if the representation of the difference is dependent on the name of the element being differenced, it is impossible to create a generic way of handling the differences and every file you evaluate will require a unique utility to interpret the difference information.

If the representation is an "edit script", perhaps based on XPath, then you need the original documents to be accessible in order to make sense of the delta. An XPath solution leads to almost incomprehensible deltas which impede data pipelining and drag down efficiency.

Change information must be generic to allow change processing to be automated. An optimal solution would keep the structure of the delta very similar to that of the original. As an added bonus, it could offer both "changes-only" and "changes-plus-original" formats, to make in-context change identification a breeze. Only the delta file would be needed for a comprehensive change report.

▲ *Usability of the solution*

Finally, it is important that the solution is both efficient and usable. Efficiency will determine its ability to cope with rapidly changing XML data or large XML data sources. To be usable the solution must be easily accessible to developers for integration into automated processes, end user tools or other applications; it must be fully standards compliant and wrap internal complexities behind a clear API. And, of course, to reduce integration costs to a minimum, good documentation and responsive technical support are essential.

Why choose DeltaXML?

We stated at the beginning of this paper that if any change control solution is to be capable of managing the huge volumes of rapidly changing XML data that now pervade organisations across the world, it must meet some key criteria. It must be compliant to open standards, it must be flexible and easy to integrate and it must have the performance and scalability to identify change in large amounts of data and growing XML file sizes.

At DeltaXML we have been working on XML differencing since it was first endorsed by the World Wide Web Consortium (W3C) in 1998 and launched DeltaXML Core three years later. Our investment since then in R&D and the wealth of practical application experience we have gained working with some of the world's leading XML users, has led to the development of a family of XML change control products that are well ahead of contemporary offerings and provide a future-proof choice for XML change control.

DeltaXML provides a complete and flexible solution which produces an optimal or near-optimal delta in every situation. It is efficient, allowing processing of large files at high speed. The delta is represented in XML with a structure close to the original so that it can be used for human consumption or as a trigger or input for further processing. For integration, DeltaXML has a carefully designed and well documented Java and .NET API based on industry standards, which

enables developers to rapidly integrate DeltaXML into effective business critical solutions. Graphical User Interface and command line access is also provided. Our documentation, configuration options and technical support are designed to get you up to speed as fast as possible.

DeltaXML provides…

- Accurate results in every XML change scenario – with provision for the user, or developer, to define how particular differences should be handled.
- Patented difference format and highly-tuned algorithms – providing a uniquely complete solution that generates the optimum change for ongoing processing.
- Close to minimal representation of change – independent tests show DeltaXML ahead of other products.
- Representation of deltas in XML – allowing consumption by humans and by automata. For example, you can generate a visual display of the delta in HTML either on its own or in the context of the original document, formatted exactly as required. Alternatively, for machine processing you can isolate the delta with a change set close to the minimum and with a notation that is accurate and allows rollback of the change at a later time.
- Representation of deltas separate to the source documents – reducing the bandwidth required to transmit the change and removing the requirement to store multiple copies of the same document
- Extensive configuration options, designed for pipeline architectures.
- Scalability: able to compare very large source files.
- Java and .NET API – fully standards compliant, well documented and designed for deep integration.

Making Commercial Sense of Change Management

So, what does all this mean in real terms? Aside from the obvious financial advantages to be gained from reducing corporate data storage and bandwidth requirements, DeltaXML can also deliver some real business advantages too.

By managing change entirely within an XML environment, DeltaXML will speed up processing and reduce errors in documents and data. But it can also add real value to content by enabling publishers to offer, not just the changed information, but information about those changes as well.

- This could be in the form of important change notices or summaries to technical documentation to speed up distribution and assimilation of critical data – aircraft manufacturers use DeltaXML to do just this for large and complex operating manuals.
- It could be to offer change information by date – allowing online users to see what changes were made and when. State legislators in the US are using DeltaXML to enable citizens to do just this.
- Or it could be to publish redlined versions of frequently accessed reference data so that changes can be easily identified. Standards organisations are already using DeltaXML to do this.

DeltaXML offers the only truly comprehensive solution for managing change in XML environments, empowering businesses and organisations not just to record change but to use change as a powerful and valuable asset.

Head Office (Sales and Support)

DeltaXML Ltd
Malvern Hills Science Park
Malvern, Worcestershire
WR14 3SZ UK

deltaxml.com
info@deltaxml.com
+44 1684 532 130

DELTAXML EXPERTS IN INFORMATION CHANGE