

# Re: XML parser and writer

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-01/msg00447.html>

---

- *From:* "Jeff Higgins" <[oojhiggins@xxxxxxxxxx](mailto:oojhiggins@xxxxxxxxxx)>
  - *Date:* Thu, 3 Jan 2008 21:31:41 -0500
- 

Brandon McCombs wrote

Jeff Higgins wrote:

Brandon McCombs wrote:

Jeff Higgins wrote:

Jeff Higgins wrote:

Brandon McCombs wrote:

Hey guys,

I'm writing a program to help people track various items and placing them on a calendar. These items are mainly ones that reoccur (monthly bills for example). Each item (or task) has various properties that are set for it and I was thinking of

Re: XML parser and writer

storing all  
the data in  
an XML  
file. I'd  
parse the  
file and  
have the  
data  
available  
throughout  
the  
execution of  
the  
program.  
Updates to  
the file  
would occur  
when  
someone  
edits a task  
to change  
its  
re-occurrence  
frequency,  
etc.  
Therefore I  
will need to  
both easily  
parse and  
write new  
XML files.  
I  
don't have a  
DTD or  
schema  
developed  
and I may  
later so any  
API I use  
will have to  
work w/o  
that for  
now. Can  
anyone  
suggest a  
simple API  
for  
parsing/writing  
my own  
made-up  
XML files?

Re: XML parser and writer

I've ruled  
out SAX  
since it only  
parses and  
was  
thinking of  
using DOM  
but I don't  
know  
if that will  
work  
because I've  
never did  
any  
programming  
with XML  
before.

thanks for  
any pointers

Maybe a Java-XML  
binding framework?  
XStream, Castor, JAXB  
come to mind.

OTOH, why not some database technology?

I thought about an internal (to the app) database however I wanted to strike a balance between beginning and advanced user configurability and the ease with which to carry the user settings from computer to computer. In my mind, an advanced user can edit the XML file directly at the risk of messing stuff up (that's where it would be nice to eventually have a schema defined) instead of using a GUI interface I'll eventually create for beginner users. Simply copying the XML file from 1 computer to another makes it easy for a user to use the application on multiple computers if desired (although it would be up to them to keep them sync'ed up).

Using a database made both of those things more difficult in my mind. An advanced user would have no way of modifying an internal

Re: XML parser and writer

db and an external db would just be overkill and make the installation of the app overly complicated.

Any thoughts to the contrary? Maybe my knowledge is incorrect regarding internal (embedded?) databases.

Well, first, thanks for the invitation to exercise my contrarian tendency. :)  
And second, my extensive knowledge of and real development experience with the above subjects are hereby disclaimed.

I can say though that I have tried unsuccessfully more than once to soften my hard head against the difficulty of using the Document Object Model as a relational database. I think it could make sense to allow your user to import and export his Calendar in XML format for the purpose of transport.  
But as a data model for something even as simple as a single user Calendar/Tasklist the DOM would quickly become unwieldy especially if you have any intention of using any part of the relations in the relational database.

Please don't get all excited about what I have to say. I'm a hobbyist programmer and as such don't have the depth of knowledge or experience that some others here have.

I didn't plan on using a database at all whether for storing user data or user/application settings. If the DOM becomes unwieldy, and SAX can't marshal new XML documents then what should I use if I want to stick with XML? I think I want to stick with XML because I thought it was the modern technology to use for not only storing data but configuration settings.

Spend some time and figure out your use cases and data model and then pick an appropriate technology with which to implement it. Starting out designing a GUI and then fitting a data model to that is counter-productive.

## Re: XML parser and writer

In the past I've used the Java Preferences API and it was great for easily retrieving, manipulating data, and then storing changes but the changes weren't portable nor editable without my custom GUI which provided both edit and import/export features. The Preferences API is just for settings and not for data so I figured I'd have to look to something else this time.

Don't know much about the Preferences API.

The XML file I plan to use is going to store not only the properties of a task (name, description, frequency, etc.) but also all the dates on which the task will occur (based on frequency) which is what I would consider the data in this situation. When the application starts up, and just before the current month is displayed, I build a data structure that contains all the dates for all the tasks on a month by month basis (the biggest timespan displayed by my GUI so far) so every time a month is displayed in the calendar all the tasks and all their occurrences are displayed. I also plan to have a list-like view of a single task, all its occurrences, and the status for which occurrences of that task have been completed.

Good. Data model and use case planning. :)

Spend a bunch of time researching and experimenting with Java-XML technologies. Including how (!) to use xml as a database. This could become a career-long endeavor.

A user may modify the frequency of a task, complete a single or multiple occurrences of a task, or create/delete a task and I was hoping an API was available for me to easily handle data elements, that were sourced from an XML file, to accomplish those user actions.

Good. Building use cases. :)

Java-XML binding frameworks.

That's what I have in my head so far for designing this. Note that this is just a for-fun project I thought I'd try for my own benefit so at this point there is no need for it to be enterprise-ready (i.e. multi-user, etc.).

Don't know about enterprise ready.

Re: XML parser and writer

Re: XML parser and writer

Multi-user is probably not synonymous with enterprise-ready.

At this point the calendar displays better than I imagined and I'm trying to wrestle around how to compute, store and display multiple occurrences of a single task, hence this thread.

Good deal. I saw your earlier thread and it sparked some interest in a Calendar widget.  
See remark above.

You may also consider your Calendar over time, and over multiple users, and over feature creep and over data types. Querying the DOM for all high priority alerts after noon in the second week of January. :(

If this was a professional project I would but I'm not worried about that level of sophistication for something like this although your example of a query probably isn't out of the question but it wouldn't return any more than a handful of results for a user like myself or a family member who might benefit from my program.

I didn't explicate very well there. I guess what I was getting at was that database technology is good for implementating databases and XML technology is good for implementing Extensible Markup Language functionality.

Having that said; If you view your Calendar as a document the the DOM is certainly the way to go.

JH