

---

Subject: More Xmlize functions

Posted by [koldo](#) on Mon, 22 Feb 2010 08:02:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello all

Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.
- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropDownList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

### File Attachments

1) [Xmlize2.7z](#), downloaded 348 times

---

---

Subject: Re: More Xmlize functions

Posted by [mirek](#) on Mon, 22 Feb 2010 08:16:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Mon, 22 February 2010 03:02Hello all

Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.
- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropDownList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

Are you sure about implementing the time/date this way?

Maybe we should rather use XmlRpc format? (You can see that in XmlRpc package).

Mirek

---

Subject: Re: More Xmlize functions  
Posted by [koldo](#) on Mon, 22 Feb 2010 09:25:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Mon, 22 February 2010 09:16koldo wrote on Mon, 22 February 2010 03:02Hello all  
Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.

- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

Are you sure about implementing the time/date this way?

Maybe we should rather use XmlRpc format? (You can see that in XmlRpc package).

Mirek

Hello Mirek

I thought XmlRpc protocol was not related with serialization.

Anyway if we do not have to use Xmlize it would be better to label it as deprecated. I have spent some hours in it until now.

If it is worthwhile to use it, please indicate me your way to implement time/date.

---

Subject: Re: More Xmlize functions  
Posted by [mirek](#) on Mon, 22 Feb 2010 11:26:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Mon, 22 February 2010 04:25luzr wrote on Mon, 22 February 2010 09:16koldo wrote on Mon, 22 February 2010 03:02Hello all

Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.
- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

Are you sure about implementing the time/date this way?

Maybe we should rather use XmlRpc format? (You can see that in XmlRpc package).

Mirek

Hello Mirek

I thought XmlRpc protocol was not related with serialization.

Anyway if we do not have to use Xmlize it would be better to label it as deprecated. I have spent some hours in it until now.

If it is worthwhile to use it, please indicate me your way to implement time/date.

Ops, nope. What I was saying that perhaps text format used in XmlRpc to represent the date with

single attribute is better than to implement it with several.

See:

<http://en.wikipedia.org/wiki/XML-RPC>

they use

```
<dateTime.iso8601>19980717T14:08:55</dateTime.iso8601>
```

of course, in Xmlize, this might be something like

```
<id value="19980717T14:08:55">
```

Mirek

---

---

Subject: Re: More Xmlize functions

Posted by [koldo](#) on Mon, 22 Feb 2010 11:41:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Mon, 22 February 2010 12:26koldo wrote on Mon, 22 February 2010 04:25luzr wrote on Mon, 22 February 2010 09:16koldo wrote on Mon, 22 February 2010 03:02Hello all

Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.
- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropDownList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

Are you sure about implementing the time/date this way?

Maybe we should rather use XmlRpc format? (You can see that in XmlRpc package).

Mirek

Hello Mirek

I thought XmlRpc protocol was not related with serialization.

Anyway if we do not have to use Xmlize it would be better to label it as deprecated. I have spent some hours in it until now.

If it is worthwhile to use it, please indicate me your way to implement time/date.

Ops, nope. What I was saying that perhaps text format used in XmlRpc to represent the date with single attribute is better than to implement it with several.

See:

<http://en.wikipedia.org/wiki/XML-RPC>

they use

```
<dateTime.iso8601>19980717T14:08:55</dateTime.iso8601>
```

of course, in Xmlize, this might be something like

```
<id value="19980717T14:08:55">
```

Mirek

Sorry for the misunderstanding

It makes sense. I will change it.

---

Subject: Re: More Xmlize functions

Posted by [koldo](#) on Mon, 22 Feb 2010 12:19:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Mon, 22 February 2010 12:41luzr wrote on Mon, 22 February 2010 12:26koldo wrote on Mon, 22 February 2010 04:25luzr wrote on Mon, 22 February 2010 09:16koldo wrote on Mon, 22 February 2010 03:02Hello all

Xmlize functions are a nice way to get serialization using instead of a raw binary file, a XML file.

Advantages are:

- If you change your classes design, .xml data saved is read properly so your old .xml files are useful.
- They integrate very well with code
- Serialization file is a readable .xml file

Now Xmlize.c /.h supports many classes. However some classes and types are not included yet:

- int16
- byte
- Date
- Time
- DropList
- DropTime
- DropDate
- EditDouble
- Option

I have included new files Xmlize2.c / .h to support these types and classes.

If you want to use them just insert Xmlize2.c int your files and #include Xmlize2.h.

Of course they could be integrated in Core Xmlize.c /.h files .

Are you sure about implementing the time/date this way?

Maybe we should rather use XmlRpc format? (You can see that in XmlRpc package).

Mirek

Hello Mirek

I thought XmlRpc protocol was not related with serialization.

Anyway if we do not have to use Xmlize it would be better to label it as deprecated. I have spent some hours in it until now.

If it is worthwhile to use it, please indicate me your way to implement time/date.

Ops, nope. What I was saying that perhaps text format used in XmlRpc to represent the date with single attribute is better than to implement it with several.

See:

<http://en.wikipedia.org/wiki/XML-RPC>

they use

```
<dateTime.iso8601>19980717T14:08:55</dateTime.iso8601>
```

of course, in Xmlize, this might be something like

```
<id value="19980717T14:08:55">
```

Mirek

Sorry for the misunderstanding

It makes sense. I will change it.

Hello all

It is done. New version of Xmlize2.7z is included in previous post.

---

Subject: Re: More Xmlize functions

Posted by [mdelfede](#) on Tue, 23 Feb 2010 10:35:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

It would be interesting to have Value type xmlized.... I've done some small parts on my own because I needed it, but it's only for some basic types, and it's too quick-and-dirty to post here

Btw, my polymorphic xmlizer could be a nice addition too... it's in Bazaar and I think it's quite functional now.

Ciao

Max

---

Subject: Re: More Xmlize functions

Posted by [mirek](#) on Sat, 27 Feb 2010 17:09:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

mdelfede wrote on Tue, 23 February 2010 05:35It would be interesting to have Value type xmlized.... I've done some small parts on my own because I needed it, but it's only for some basic types, and it's too quick-and-dirty to post here

Btw, my polymorphic xmlizer could be a nice addition too... it's in Bazaar and I think it's quite functional now.

Ciao

Max

BTW, why do you create specific Xmlizes for CtrlLib types, IMO it should be possible to have just single generic one? (GetData/SetData).

(Only asking to be sure I have not missed anything

Mirek

---

---

Subject: Re: More Xmlize functions  
Posted by [mdelfede](#) on Sat, 27 Feb 2010 22:51:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Well, mostly because I needed some processing on ctrls loading.  
My xmlizers were mostly aimed to solve a concrete problem, they are not generic ones.  
BTW, I think it would be good to incorporate xmlize into ctrl, and to add OnLoad / OnStore handlers, I guess it could solve most streaming issues.

Ciao

Max

---

---

Subject: Re: More Xmlize functions  
Posted by [mirek](#) on Sun, 28 Feb 2010 08:28:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mdelfede wrote on Sat, 27 February 2010 17:51Well, mostly because I needed some processing on ctrls loading.  
My xmlizers were mostly aimed to solve a concrete problem, they are not generic ones.  
BTW, I think it would be good to incorporate xmlize into ctrl, and to add OnLoad / OnStore handlers, I guess it could solve most streaming issues.

Ciao

Max

That would mean all XML stuff gets always linked in. Something I am not really happy about...

---

---

Subject: Re: More Xmlize functions

Posted by [mdelfede](#) on Mon, 01 Mar 2010 10:17:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Sun, 28 February 2010 09:28mdelfede wrote on Sat, 27 February 2010 17:51Well, mostly because I needed some processing on ctrls loading.

My xmlizers were mostly aimed to solve a concrete problem, they are not generic ones.

BTW, I think it would be good to incorporate xmlize into ctrl, and to add OnLoad / OnStore handlers, I guess it could solve most streaming issues.

Ciao

Max

That would mean all XML stuff gets always linked in. Something I am not really happy about...

Anyways, I don't think a generic ctrl Xmlizer would be very useful, at least, not a simple one. With the OnLoad/OnStore handlers( btw, you'd need a BeforeStore and AfterLoad ones, to be precise...) things would be better, but you'd have to link xml everywhere.

But, with Value xmlized now, to store/load controls becomes trivial.

A generic xmlizer would have problems, for example, with droplists which content depends/links to other controls which are in turn xmlized. On my code, most of the work is to keep in sync all that, not the streaming code itself which is really trivial.

More, In my code I store/internally use values in a single set of units (for example, Newton, mm, N/mm2, and so on) converting on the fly between textual controls contents which is user-customizable. So, when streaming in/out controls I have to perform such conversion, too. That would be hardly achieved by a generic Xmlizer.

Ciao

Max

---