Base Language - Feature #1645

implement web services support

10/21/2012 12:29 PM - Greg Shah

Status:	Closed	**Start date:**	02/20/2013
Priority:	Normal	**Due date:**	08/09/2013
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Runtime Support for Server Features		
billable:	No	**version:**	
vendor_id:	GCD		

Description

Subtasks:

Feature # 2021: conversion support for web services	**Closed**
Feature # 2022: runtime support for web services	**Closed**

Related issues:

Related to Base Language - Feature #2208: create a java application running w...	**Closed**
Related to Base Language - Feature #2215: implement the unwrapped mode of pas...	**New**
Related to Base Language - Feature #2225: implement ssl and async realted co...	**New**
Related to Base Language - Feature #2232: finish testing the wrapped mode usi...	**Closed**

History

#1 - 10/21/2012 01:00 PM - Greg Shah

Priority work: implement all usage of the server object handle in regards to connecting to a web service.

Most of the code is common to appserver or sockets. BUT there is this which does need to be added:

```
SET-CALLBACK-PROCEDURE()
```

#2 - 10/31/2012 01:36 PM - Greg Shah

- Target version set to Milestone 7

#3 - 02/23/2013 04:28 AM - Constantin Asofiei

For the Server object handle, all documented attributes/methods are supported except these:

1. REQUEST-INFO, RESPONSE-INFO attribute - the datatype is Progress.Lang.OERequestInfo class
2. RESET method, RESTART-ROWID attribute - following code shows that they do not apply to server object:

```
def var h as handle.
create server h.
h:reset(). /* not queryable attribute for server widget */
h:restart-rowid. /* not queryable attribute for server widget */
```

3. Note that the connect method uses different parameters to connect to a web services, but there is no difference conversion-wise.

Syntax for set-callback-procedure (used with a procedure handle):

```
set-callback-procedure(callback-name,internal-procedure [ , procedure-context ])
```

where callback-name must evaluate to either "REQUEST-HEADER" or "RESPONSE-HEADER". Other methods which are somehow related to callback, but they do not apply to procedures (I think they only work with buffer/dataset/query-related events):

- APPLY-CALLBACK( ) method
- GET-CALLBACK-PROC-CONTEXT( ) method
- GET-CALLBACK-PROC-NAME( ) method

Browsing the OE Dev: Web Services book, I cant find any other methods/attributes related to web services, which we don't currently support.

Looks like I've missed this RUN statement syntax, related to web services:

```
RUN portTypeName [ SET hPortType ] ON [SERVER] hWebService [ NO-ERROR ].
```

If both the SET and the ON service clauses are present, then this needs to convert to a ControlFlowOps.invokeWebService. In the RUN ... ON case, the runtime will disambiguate based on the handle type. The plan for now is to:

1. fix the RUN SET ... ON SERVER statement
2. add SET-CALLBACK-PROCEDURE to PeristentProcedure.setCallbackProcedure

#4 - 02/23/2013 04:51 AM - Constantin Asofiei

> fix the RUN SET ... ON SERVER statement

Look like the ASYNC clause is supported (at least at compile-time) for this case too.
LE: plus the TRANSACTION DISTINCT clause, which is also allowed.

#5 - 02/23/2013 05:28 AM - Constantin Asofiei

Another later edit: it is possible to use the SET hPortType specification even with the TRANSACTION DISTINCT and PERSISTENT SET clauses. I think the best solution is to add another handle parameter (right after the proc name) to all ControlFlowOps.invokeRemote APIs, to not duplicate them all... and if the standalone SET clause is not present, emit null for that parameter.

#6 - 02/23/2013 08:39 AM - Constantin Asofiei

More info about RUN ... SET ... ON SERVER. Something like this works:

```
def var h1 as handle.
def var h2 as handle.

procedure proc0.
    message "bla".
end.

h2 = session.
run proc0 set h1 on server h2. /* runs proc0 */

message h1. /* this is unknown */
```

but the SET clause doesn't change the targeted handle (I guess because we are not running a webservice).
Following is a sample program which connects to a web service and invokes it:

```
DEF VAR i AS INT.
DEF VAR j AS INT.
DEF VAR ch1 AS CHAR.
DEF VAR ch2 AS CHAR.

def var hs as handle.
def var hws as handle.
def var l as log.

create server hs.
l = hs:connect("-WSDL http://www.webservicex.net/convertMetricWeight.asmx?WSDL -Port MetricWeightUnitSoap").

if l = no
then do:
  message "could not connect".
  return.
end.

run MetricWeightUnitSoap set hws on server hs.
if not valid-handle(hws)
then do:
  message "could not establish port".
  return.
end.

MESSAGE hws:TYPE hws:NAME. /* shows "PROCEDURE MetricWeightUnitSoap" */
i = 1000.
ch1 = "microgram".
ch2 = "milligram".

run ChangeMetricWeightUnit in hws (INPUT i, INPUT ch1, INPUT ch2, OUTPUT j).
message ch1 ch2 i j.

delete object hws.
hs:disconnect().
delete object hs.
```

#7 - 02/23/2013 11:26 AM - Constantin Asofiei

- File ca_upd20130223e.zip added

Built on top of #1782 23d.zip update. Fixed support for RUN ... SET hPortType ON hWebServ statement. Added support for SET-CALLBACK-PROCEDURE.

#8 - 02/23/2013 11:38 AM - Greg Shah

In fixups/control_flow.rules refAsync is never assigned, but it is relied upon in the ASYNCH ON case and in the SET hPorttype case. The later case has no null protection, so it should abend. The former case is protected, but will just never trigger.

#9 - 02/23/2013 11:39 AM - Greg Shah

Everything else looks OK with the code.

#10 - 02/23/2013 11:46 AM - Constantin Asofiei

- File ca_upd20130223f.zip added

Attached update fixes the refAsync issue and refSet.move (which should move to copy not refAsync)

#11 - 02/23/2013 11:54 AM - Greg Shah

It looks good. This will go into conversion testing next.

#12 - 02/23/2013 12:21 PM - Greg Shah

This is in testing now.

#13 - 02/23/2013 04:19 PM - Greg Shah

- File ca_upd20130223j.zip added

This update causes a problem with many (all?) RUN statements. Here is a common example diff (this is broken throughout Majic):

```
66c66
<             ControlFlowOps.invoke("xt/quitsym.p");
---
>             ControlFlowOps.invoke("xt/quitsym.p", (handle) null);
```

The new null handle that is passed should not be there. This is a compile problem since there is no signature that matches.

I believe that the problem was in rules/annotations/control_flow.rules where we manufactured a null SET hPortType clause when:

```
type == prog.kw_run and !downPath("KW_SET") and !downPath("KW_IN")
```

But I think that this is only valid for remote calls, which means that there should always be an ON server spec. I changed the test to the following:

```
type == prog.kw_run and downPath("KW_ON") and !downPath("KW_SET") and !downPath("KW_IN")
```

The updated code is attached here. I am putting this back through testing.

#14 - 02/23/2013 05:26 PM - Greg Shah

With this change, testing passed. Checked into bzr as 10193.

#15 - 04/25/2013 10:38 AM - Greg Shah

Comment from CA:

This and #1995 should be worked by the same person, as they are related (I think the web services need the SOAP features internally). And about the X-Document features: serialized XMLs (built using X-Document, then saved to a longchar var) can be passed as input to SOAP service invocation. We should be able to build the SOAP/Web Services without prior X-Document features, but we should test them with the XMLs too.

#16 - 12/15/2013 03:25 PM - Constantin Asofiei

Following are findings about how 4GL implements the web service requests and what tools we need to address this. The most important tool in 4GL is the bprowsdldoc command which receives as argument the URI for a WSDL document and will generate reports on how the services exposed by the WSDL document can be used in 4GL.

About the WSDL document: 4GL will validate the connect options (service, binding and the port type) against the WSDL file. Basically, the WSDL document is an XML file (which we could parse and obtain a DOM Document instance), but this will force us to build tools for accessing the WSDL data in an easier way. The alternative is wsdl4j, which provides a WSDL reader and is realeased under Common Public License - v 1.0. It implements JSR 110 (Java APIs for WSDL) and is found here: http://wsdl4j.sourceforge.net/ ; Also, it is internally used by AXIS2.

Some details about connections: from what I can tell, establishing a web server connection just validates the passed parameters against the WSDL file (and possibly override some options, like -SOAPEndpoint). Once a web server "connection" was established, this means that a port type that can be executed with the RUN port-type SET h-port-type ON SERVER h-web-server statement. At this time, I think once the CONNECT statement establishes a connection, this only determines which port type will be exposed to the RUN statement (this is because a binding is associated with only one port type and, if the service was specified, the port must be specified (if the service has more than one port).

When the RUN statement is used to execute a port type, this only collects the information about the exposed operations (i.e. parameters and return values); the procedure attributes/methods (like get-signature, internal-entries, etc) are a no-op for the "port-type" procedure resource. More, I think no actual network connection to the remote web server is established.

Later on, when an operation is ran on the connected port type using a RUN operation IN h-port-type statement, the operation must be part of the loaded port type. Now, the interesting part. 4GL allows two modes for passing parameters: wrapped and unwrapped. In wrapped mode, the input will be the request message and the output will be the response message, from the SOAP Body node, in serialized XML form; these are passed as char or longchar variables (request as input and response as output). For the unwrapped mode, things start to get... complicated; 4GL will use parameter mode, type and order to map passed parameters to the SOAP response and request message elements; based on the order, it will try to convert from the 4GL data type to the WSDL message data type and viceversa; thus, you can use i.e. a 4GL char variable as output for a decimal value. More, you can use mixed mode: i.e. input parameters can be unwrapped, while the output be wrapped; not sure if mixed mode can be used inside the list of input or output parameters. A question here is how will 4GL handle a case when both the single input and single output are character: will see them as wrapped, unwrapped, combination?

Considering the mixed mode of passing arguments to the SOAP message (and back), I think we can use Apache AXIS2 as a tool to communicate with the web service, as it provides:
- connection pooling (for async requests, but not fully sure I will use this)
- both sync and async requests
- invoke the services java style or low-level, by passing the SOAP message directly
- the downside of AXIS is that it comes together with lots of dependencies (I will need to provide the minimal list of jars)

With AXIS2, at first I wanted to use this approach: provide at build time all the WSDL documents reached by the CONNECT statement and use AXIS to generate java code to access these services; using this generated code (and some other configurations), we could use java reflection to invoke the web service java-style, and not care about how the SOAP message is encoded. But, considering the mixed mode of sending and receiving a SOAP message, this will not save us the effort of checking each passed parameter against the message definition. Thus, as using java style calls does not worth the effort, I think the best solution is to send the SOAP message as wrapped (regardless of how the input parameters are specified) and use the wrapped received SOAP message to set the output variables accordingly.

Finally, web services allow temp-table data to be passed as parameters (both input and output); these are mapped to specially-configured complex data types in the WSDL document. We can add these later, but I will ask Guy for their WSDL documents, to make an idea of how complex are the web services they are using.

#17 - 12/15/2013 07:50 PM - Evgeny Kiselev

This is training server from progress:
http://wstraining.progress.com/index.html

#18 - 12/16/2013 03:28 AM - Constantin Asofiei

I don't want to rely on progress or any other freely available services for the testing. Look at the WSDL structure, make a plan of cases and lets discuss it in #2208.

#19 - 12/17/2013 12:56 PM - Greg Shah

In regard to the CPL 1.0, it is acceptable to use code under that license.

License text:

http://opensource.org/licenses/CPL-1.0

Ultimately, we probably won't include any of these libraries in our "distributions" of P2J. Instead, we will provide a mechanism to easily pull down and patch the versions needed (perhaps through Maven). Customers can pull the specific versions manually too.

But even if we did ship the binaries under the P2J distribution, the CPL does not have the specialized linking behavior that is part of the GPL. This means that having Java code that depends upon the wsdl4j classes/jar does not merge P2J together with wsdl4j as a single work. In other words, P2J is not part of the "Program" nor is it a "Contribution". Note this text in section 1 of the license:

```
Contributions do not include additions to the Program which: (i) are separate modules of software distributed
in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Pr
ogram.
```

Thus, the restrictions you note are only in regard to the wsdl4j code itself, which is not a problem.

#20 - 12/17/2013 02:03 PM - Greg Shah

I have reviewed the above findings. The approach seems reasonable. I'll check the licenses for the AXIS2 dependencies when you post the list.

#21 - 12/20/2013 02:36 PM - Constantin Asofiei

The wrapped mode is working nicely, async mode is almost done too (only remaining feature is canceling the requests), but unwrapped mode I think is going to be a PITA. We will need extensive testing to fully support it properly, as a message can have parts for which their type can be a complex type defined in some custom schema - and we will need to cover all possible schema type definition to be sure we get it right.

What we can do is this: from what I understand, the server project is using only wrapped mode, where each service operation call has at most one input and at most one output parameter, both of type char or longchar (I've double checked this in the 4GL source code and all service operation calls are seem to be invoked in wrapped mode). Thus, we can postpone the unwrapped mode to a later time.

Beside the unwrapped mode, what I need is to figure out how the soap faults are reported by axis2 and add support to create the ERROR-STATUS:ERROR-OBJECT-DETAIL. I will not go into building the SOAP-fault object, I just need to find out how the soap faults are reported by axis2 and invoke a SoapFactory.createSoapFault API which will create the object.

The minimal list of axis2 dependencies is:

```
axiom-api-1.2.13.jar
axiom-impl-1.2.13.jar
axis2-adb-1.6.2.jar
axis2-kernel-1.6.2.jar
axis2-transport-http-1.6.2.jar
axis2-transport-local-1.6.2.jar
commons-httpclient-3.1.jar
commons-logging-1.1.1.jar
httpcore-4.0.jar
mail-1.4.jar
neethi-3.0.2.jar
wsdl4j-1.6.2.jar
XmlSchema-1.4.7.jar
```

from which:

- mail-1.4.jar is released under COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
- wsdl4j-1.6.2.jar is released under Common Public License - v 1.0
- all others are released under Apache License Version 2.0

#22 - 12/20/2013 06:38 PM - Greg Shah

1. Create new redmine tasks for the unwrapped mode and any error/fault processing that is not being implemented. Put enough detail in there for the future readers to understand exactly what needs to be done.

2. I believe that the CDDL 1.0 code can be used. Basically it is a weak copyleft license that does not have the linking requirements of the GPL. In other words, this code can be included on a source or binary basis in other projects (a "Larger Work") which has a different license (including a closed source license) and the source code of the closed license does not need to be released (under the CDDL or another license). Similar to the CPL, the CDDL requires providing the source code and an attribution to the project in our docs. Just like as mentioned above, we won't actually distribute this code with P2J and thus we will avoid even these other requirements.

http://opensource.org/licenses/CDDL-1.0

```
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by th
e terms of this License.
```

It looks to me like all of the additional jars should be OK.

#23 - 12/23/2013 08:13 AM - Constantin Asofiei

- File ca_upd20131223a.zip added

I've cleaned up and documented the code for the web services implementation. What major features are missing:

- finish implementation and testing of the request-header and response-header callbacks (these can be finished when implementing the SOAPHeader features).
- testing of soap faults (this I will finish after a first version of the framework for #2208 is ready, so I can test the soap faults properly).
- add support for the unwrapped mode.

Beside these, there are some low-priority connect options (-ServiceNamespace, -BindingNamespace@, -SOAPEndpointUserid, -SOAPEndpointPassword, -WSDLUserid and -WSDLPassword) which can be postponed and some other TODO's which should not impact how the server project uses the web services. There might be some changes to ControlFlowOps or WebServiceHelper, once I finish testing all the error cases.

#24 - 12/24/2013 07:40 AM - Constantin Asofiei

- File ca_upd20131224a.zip added

This version adds validation for the header callback proc/context and fixes the soap faults (though some more testing is still required). Full SOAP Header and SOAP Fault support can be easily put in place.

#25 - 01/04/2014 04:01 PM - Constantin Asofiei

Evgeny: please help me a little with something related to X-Noderef resources. Is there an existing way to serialize a XNoderef instance to a string? (i.e. return the entire XML sub-tree represented by a node in a string).

#26 - 01/04/2014 04:44 PM - Constantin Asofiei

LE: Nevermind, there is the XmlHelper.write in P2J.

#27 - 01/04/2014 07:03 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> LE: Nevermind, there is the XmlHelper.write in P2J.

I've used for SAX writer:

```
static String serializeNodeToString(Node node)
throws TransformerFactoryConfigurationError,
       TransformerException
{
   StreamResult xmlOutput = new StreamResult(new StringWriter());
   Transformer transformer = TransformerFactory.newInstance().newTransformer();
   transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes");
   transformer.transform(new DOMSource(node), xmlOutput);
```

#28 - 01/06/2014 11:43 AM - Greg Shah

Code Review 1224a

I like the improved abstraction of some parts of the runtime (e.g. ServerHelper). It also seems like you have cleaned up some latent bugs.

Unfortunately, there a huge amount code in WebServiceHelper that is specific to the 4GL SOAP implementation. As such, I can't really evaluate it effectively without going much deeper in that area of the 4GL, which I don't have time to do right now. On the surface it looks OK, but that is not a deep look.

Evgeny: please take a look at this code (realizing that it is an early/unfinished version). In particular, try to focus on the SOAP-specific parts and see if you can find anything of concern.

Other than that, my only question is in ControlFlowOps.initializeProxy(), is there any way the IllegalStateException can be thrown when processed from converted code?

#29 - 01/06/2014 03:16 PM - Constantin Asofiei

Greg Shah wrote:

> Other than that, my only question is in ControlFlowOps.initializeProxy(), is there any way the IllegalStateException can be thrown when processed from converted code?

Looking back at it, it is only called from invokeRemotePersistentSetAsyncWithMode which already is protected against calls using a web service; so, this exception can never be thrown.

#30 - 01/09/2014 03:09 PM - Constantin Asofiei

- File ca_upd20140109d.zip added

Attached update adds:

1. runtime implementation for legacy SOAP Header, Header Entry, Fault and Fault Detail resource. The SOAP Header stuff (creation, deletion, management, etc) is tested and works properly.
2. integration of legacy SOAP Header and SOAP Fault resources with web services. Only basic tested was done.

Please review. I'll perform runtime testing tomorrow.

#31 - 01/09/2014 04:34 PM - Greg Shah

Code Review 0109d

1. SOAPFactory.throwError() needs javadoc.

2. How safe is the use of IllegalArgumentException in SOAPHeaderImpl.getHeaderNode()/setHeaderNode()?

3. WebServiceHelper.msgParams needs javadoc.

4. Several of the files have explicit imports instead of using *.

5. WebServiceHelper.WebServiceConnectOptions has many methods without javadoc.

6. I see that you are using the Java 7 catch (|) syntax. That should be fine, but it is a more explicit dependency on Java 7 than we had before (I think we could still get away with compiling under Java 6 before). It isn't a problem, just a note.

7. I wonder if we have any potential character set conversion issues in our usage of the various InputStream/OutputStream (byte oriented streams versus character oriented reader/writer) constructs? Even if there are some issues there, I would say we create a separate task to consider/solve such issues. But I want us to know if there are any concerns there.

Otherwise, I don't see anything wrong. This version is much more complete, of course. But I still lack some domain understanding to provide a more complete evaluation.

#32 - 01/10/2014 07:41 AM - Constantin Asofiei

- File ca_upd20140110a.zip added

Greg Shah wrote:

> 2. How safe is the use of IllegalArgumentException in SOAPHeaderImpl.getHeaderNode()/setHeaderNode()?

This exceptions should never be thrown; if this code is reached, there is a problem in the SOAP Header resource implementation. It was added as a means to mark SOAP Header related problems, to ease debugging. I've added a comment to the code too.

> 7. I wonder if we have any potential character set conversion issues in our usage of the various InputStream/OutputStream (byte oriented streams versus character oriented reader/writer) constructs? Even if there are some issues there, I would say we create a separate task to consider/solve such issues. But I want us to know if there are any concerns there.

Your concern I think is valid, but I will not be able to test more until we are able to test our own custom web services.

See attached for the modified update.

#33 - 01/10/2014 08:38 AM - Greg Shah

Everything looks good. My only question is related to this (and the related commented-out code):

```
 * The {@link #CONNECT_OPTION_MAX_CONNECTIONS}, {@link #CONNECT_OPTION_CONNECTION_LIFETIME},
 * {@link #CONNECT_OPTION_NO_SESSION_REUSE} and {@link #CONNECT_OPTION_NO_HOST_VERIFY} options are
 * not implemented by {@link #connect}.
```

Does this mean these are **not yet** implemented or that **don't need** to be implemented?

#34 - 01/10/2014 08:45 AM - Constantin Asofiei

Greg, all of these are not used by the server code. Also, the gain vs cost of implementing them I don't think is worth it (this will mean hacking AXIS2 to make connections persistent, hacking the AXIS2's SSL connection code to reuse the SSL sessions and a lot of other ugly changes).

#35 - 01/10/2014 08:47 AM - Greg Shah

I have no problem deferring this work. Please add comments to the javadoc to explain that the work is a TODO and the reason for the deferral.

In addition, add a task to Redmine to describe the work and make it related to this task.

#36 - 01/10/2014 09:13 AM - Constantin Asofiei

- File ca_upd20140110c.zip added

Added the TODO.

Evgeny: when you find some time, please take a look into WebServiceHelper.invoke, the core code for consuming the service (via AXIS2).

#37 - 01/10/2014 09:38 AM - Greg Shah

OK, finish your own testing AND take into account any feedback from Evgeny from his code review of WebServiceHelper.invoke().

Then go ahead with regression testing.

#38 - 01/10/2014 10:32 AM - Constantin Asofiei

Greg, I'm fine with how the runtime works with my testing. I'll put this through regression testing once I'm done with the appserver related testing, but I will not release it until I here from Evgeny.

#39 - 01/16/2014 03:26 AM - Constantin Asofiei

main part of runtime testing has passed. need to double check the CTRL-C part.

Attached version is merged with rev 10438.

#40 - 01/16/2014 03:27 AM - Constantin Asofiei

- File ca_upd20140116a.zip added

and the update...

#41 - 01/16/2014 06:42 AM - Greg Shah

Code Review 0116a

I am fine with the changes. Go ahead and check it in.

Evgeny: I still want you to review this code and post your findings.

Constantin: If there are changes to make later based on the review, you can do so and check in without regression testing IF those changes are in code that cannot be reached by MAJIC.

#42 - 01/16/2014 07:19 AM - Constantin Asofiei

Committed to bzr revision 10439.

#43 - 01/20/2014 02:13 PM - Evgeny Kiselev

Code Review 0116a:
1) com.goldencode.p2j.xml.XNodeRefImpl#getOwnerDocument()
Does RESTRICTED_DOCUMENT can affair to other methods in XNode ? if yes, we need merge this code with isInitialized() method.
2) AsyncRequestImpl
on line 613 is better to use notifyAll() instead of notify(), because it may lead to thread "starvation" problem.

#44 - 01/21/2014 04:19 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> Code Review 0116a:
> 1) com.goldencode.p2j.xml.XNodeRefImpl#getOwnerDocument()
> Does RESTRICTED_DOCUMENT can affair to other methods in XNode ? if yes, we need merge this code with isInitialized() method.

Access is restricted only to OWNER-DOCUMENT, for X-Nodes attached to a SOAP-Header or SOAP-Fault-Detail resource; the X-Nodes are valid, is just that in 4GL access to the SOAP Envelope via DOM is not allowed. But I think I've missed a case: when cloning the DOM Node in SOAPHeaderEntryImpl.setNode, I think RESTRICTED_DOCUMENT info should have been set at the clone's document. I will double check how it behaves and fix this.

> 2) AsyncRequestImpl
> on line 613 is better to use notifyAll() instead of notify(), because it may lead to thread "starvation" problem.

This is safe, as there can be at most one thread waiting for this (when an explicit STOP condition is sent to terminate the request).

#45 - 01/23/2014 04:40 PM - Greg Shah

- Status changed from New to Closed

#46 - 11/16/2016 11:42 AM - Greg Shah

- Target version changed from Milestone 7 to Runtime Support for Server Features

Files

ca_upd20130223e.zip	51.2 KB	02/23/2013	Constantin Asofiei
ca_upd20130223f.zip	51.3 KB	02/23/2013	Constantin Asofiei
ca_upd20130223j.zip	51.4 KB	02/23/2013	Greg Shah
ca_upd20131223a.zip	3.4 MB	12/23/2013	Constantin Asofiei
ca_upd20131224a.zip	3.4 MB	12/24/2013	Constantin Asofiei
ca_upd20140109d.zip	3.43 MB	01/09/2014	Constantin Asofiei
ca_upd20140110a.zip	3.43 MB	01/10/2014	Constantin Asofiei
ca_upd20140110c.zip	3.43 MB	01/10/2014	Constantin Asofiei
ca_upd20140116a.zip	3.43 MB	01/16/2014	Constantin Asofiei