

User Interface - Feature #3287

local web client installation

04/27/2017 02:48 PM - Greg Shah

Status:	New	Start date:	04/27/2017
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to User Interface - Feature #3288: support > 1 browser tabs for a sin...		New	
Related to User Interface - Feature #2683: reverse proxy implementation		Closed	

History

#1 - 09/28/2017 01:39 PM - Sergey Ivanovskiy

Greg, please clarify the web client installation task. Does it include the spawn installation? What is an environment? I guess that the web client must be exposed via the Apache reverse proxy. Correct?

#2 - 09/28/2017 03:02 PM - Greg Shah

Every 4GL GUI application we have ever seen contains some number of native platform dependencies. Since these applications only work on Windows, this translates into a variety of Windows dependencies. OCX usage, DLL usage (WIN32 or 3rd party), COM Automation, Windows registry access, running Windows-specific child processes are examples.

These very same customers are urgently trying to shift to the web. No one wants a Windows application in today's world (2017). In fact, no one has wanted a Windows application for some time (before 2010 most likely). The FWD web client is a really impressive solution for this pervasive problem.

It is highly likely that customers will not be able to remove all of the local dependencies from their application before needing to put their converted application into production. In some cases, this may be due to the amount of work needed to rewrite something that uses Excel or other native programs. In other cases this may be due to needing a local device (e.g. a label printer on the manufacturing line) or some locally installed software (a CAD program). I know that many client requirements can be moved to the cloud as long as they remove the Windows-specific nature and make it non-interactive. But these dependencies are not just Windows-specific, but possibly are inherently local to the customer's machine/site.

We know that customers can run the swing client locally and we could support these use cases as is. Since the Swing client runs on the desktop system for each user, it is running on the same system on which the old 4GL Windows GUI application was previously running. This means that all of the native Windows dependencies can be directly accessed. DLLs can be called, COM Automation objects can be invoked and so forth. This is an option for customers so that they don't have to remove all the Windows dependencies in advance.

On the other hand, customers want all users to use the web interface, since getting away from the old Windows GUI is a key objective. This becomes a problem, for example, when new screens are written in html5/javascript which cannot be accessed in the Swing client. The problem is that the web client runs remotely, on a server. Although it is possible to run this on a Windows server somewhere. such an approach is expensive AND it can only be used with Windows platform dependencies that are NOT interactive. If the 4GL application launches a separate .EXE that displays a user interface, this cannot be done on a remote Windows server.

The idea of this task is to make changes to our client to allow the FWD client process to run locally (on the end-user's system), but where the FWD web client is executed instead of the FWD Swing client. This could easily be made to work for virtual desktop mode, but that wouldn't have any manually written web screens either. Embedded mode will be the key here. The question is: can we run a locally installed web client but still use embedded mode? In this case the embedded mode server is still installed remotely, not installed locally. How would our FWD web client process need to change to allow this to work?

Option 1: We could use a remote spawner for this. But in that case we would have to have a remote spawner + web client instance on every local user's system.

Option 2: We allow the web client to be started and to connect in unauthenticated mode. It just sits and waits until the embedded mode app connects and wants to attach to the web client. Then we "upgrade" to an authenticated connection and allow the load of the web client into the iframe to work normally from there.

Option 3: We write a thin, non-interactive version of our client platform delegate that runs like a daemon process (kind of like an appserver agent), connecting to the server and waiting for "jobs". Then we redirect the local client processing to this daemon. This one seems hard because the application code might need to be reworked to make this reasonable.

My preference is option 2. What options can you think of that I have not considered? What problems are there with the ideas above?

#3 - 09/28/2017 03:02 PM - Greg Shah

From Constantin:

I think the 2nd option is the way to go; we will need a way to mark certain users that they will use an explicit client and also ensure the client where we redirect this request is authorized to handle it. Another issue will be to keep this client alive, i.e. when the user terminates its web client the P2J client will fallback in the 'listening' mode and not actually terminate.

Also, this will allow only one web client for this mode to be connected, it will not be possible to have multiple clients connected (unless more than one client is started)... although we could make this client to act as a remote spawner, too, so if it is busy it will start another one.

#4 - 09/28/2017 03:03 PM - Greg Shah

From Sergey:

Did you consider a use case to update the P2J client from a server with new version? Please explain more thoroughly the client use case. I didn't understand these client's requirements.

#5 - 09/28/2017 03:05 PM - Greg Shah

Did you consider a use case to update the P2J client from a server with new version?

No, I didn't consider this. I assumed that they would track and apply the updates to the p2j.jar, libp2j.dll and other dependencies for the client, whenever the server side changed. We might consider some scripts for this, but it is a deployment concern only.

Please explain more thoroughly the client use case. I didn't understand these client's requirements.

This is a software developer that writes an ABL application and licenses it to many different companies. These different companies currently have local installs of a multi-user Progress database and each end-user has a Windows system and runs the actual application. They must deploy Progress clients, their "compiled" application and all the native dependencies/configuration/data files/programs... to each of these Windows systems. They directly use Excel in thousands of places in the application, via COM automation. They make many Windows DLL calls throughout the application. They run specialized 3rd party .exe files on the client systems to drive local label printers. Many of these things can't be eliminated.

They have too many Windows dependencies to eliminate all of them before going into production with FWD. They want to move to a web based UI with all of their ADM and ADM2 GUI screens in embedded mode. Often, the server will be hosted in the cloud and will be a Linux VM with Postgres and FWD. But they have users that need to execute these local things that would require a local FWD client JVM.

I am trying to see how we can still give them a web client but have it be local instead of a JVM spawned on the remote server VM.

#6 - 09/28/2017 03:07 PM - Greg Shah

I think the 2nd option is the way to go; we will need a way to mark certain users that they will use an explicit client and also ensure the client where we redirect this request is authorized to handle it. Another issue will be to keep this client alive, i.e. when the user terminates its web client the P2J client will fallback in the 'listening' mode and not actually terminate.

The "only one client" limitation is a good point. With that restriction, it might be necessary to go with a remote spawner so that multiple client processes are possible. On the other hand, if we can handle the multiple screen case below with a single client, then it would be much better.

Also, this will allow only one web client for this mode to be connected, it will not be possible to have multiple clients connected (unless more than one client is started)... although we could make this client to act as a remote spawner, too, so if it is busy it will start another one.

Hmmm, interesting point. We already know they will have the requirement to allow multiple "tabs" so that different modules can be run simultaneously. The set of related tabs in one window of an ADM/AMD2 application would be considered one module and those application "tabs" would be on the same browser tab.

I guess we can provide a virtual tab approach such that the top (z-order) tab can surface the iframe. However, this is a bit of a user problem which seems like a poor design. I think customers want might need a kind of "split screen" mode where two legacy windows (thus two or more iframes) are active at once.

Can we support this idea from a single client process?

- in the same browser tab?
- in different browser instances?

#7 - 09/28/2017 03:08 PM - Greg Shah

From Constantin:

The first thing that comes to mind is how we will handle the business logic: if a single client process is used, and we want two or more iframes to be used, what happens if the business logic is at different points in the code: i.e. an iframe is doing server-side work and the other wants to do other server-side work? Unless we allow only one iframe to be active at a time (i.e. no concurrency), I don't see a simple solution. A mode where one or more windows are active and each one assigned a dedicated iframe could work, but adding concurrency so that each window works on its own will be tricky.

So, I don't think is a matter of where the iframe resides (same browser or not), the hard problem is how we manage these concurrently.

#8 - 09/28/2017 03:08 PM - Greg Shah

A mode where one or more windows are active and each one assigned a dedicated iframe could work, but adding concurrency so that each window works on its own will be tricky.

I don't think we need real concurrency. I think this should work exactly like when our current (virtual desktop) client has multiple windows. Only one can have focus at a time, but both can be displayed and focusable. There would still only be a "single thread" and event queue on the Java side. This really is designed to be the multi-window case.

The problem that I see is that our web client is not designed to have multiple https sessions/web sockets. If we have multiple iframes in the same browser session, perhaps we can have one that is the real session and the others redirect their web socket usage through cross-document messages?

Or can we more directly handle this because we have the same origin?

The multiple browser case seems harder.

#9 - 09/28/2017 03:09 PM - Greg Shah

From Constantin:

By 'same origin' you mean same host and port, right? This could help so we will have a direct JS access between the iframes.

But why not add support for multiple web sockets? The driver will just send all commands to the web socket associated with the active iframe, and when iframes are switched, it will notify the driver which web socket to use.

If we do this in the JS code via a single web socket, we will need something similar, so that the web socket message is redirected to the active iframe.

But what if an iframe receives a command for a window which is located in another iframe? This implies that a web socket message will use the active iframe only when creating a window, all other messages will have to determine the iframe to which the window belongs and redirect the message there, as the window's iframe might not be the active iframe.

#10 - 09/28/2017 03:09 PM - Greg Shah

By 'same origin' you mean same host and port, right? This could help so we will have a direct JS access between the iframes.

Yes, exactly. Without the same-origin, you must use something like cross-document messaging to communicate.

But why not add support for multiple web sockets? The driver will just send all commands to the web socket associated with the active iframe, and when iframes are switched, it will notify the driver which web socket to use.

I like this, if we can make it work. This would enable the multiple browser and multiple (real) browser tab cases too, right? That could be very useful.

We will have to respond to focus/activation changes which will be coming from the JS side.

Also, how do we handle up calls? If we have multiple real browser tabs or multiple browsers, then each one would need to make up-calls to support the embedded app's needs.

But what if an iframe receives a command for a window which is located in another iframe? This implies that a web socket message will use the active iframe only when creating a window, all other messages will have to determine the iframe to which the window belongs and redirect the message there, as the window's iframe might not be the active iframe.

Yes.

#11 - 09/28/2017 03:10 PM - Greg Shah

Yes, multiple web sockets can allow multiple browser/tabs. Another issue to think about: what happens when a tab (same browser or not) is closed - how do we cleanup the windows associated with those iframe(s), from the server-side and client-side?

Also, how do we handle up calls? If we have multiple real browser tabs or multiple browsers, then each one would need to make up-calls to support the embedded app's needs.

Well, the embedded app will need to know which iframe to use for a certain call, and direct the call to that iframe. I don't see an issue here, as long as the correct web socket is used, the response (and the business logic work) will use that web socket. Assuming we are correctly deactivating the other iframes so that the user can't focus it accidentally and switch the web socket, or another way to force that web socket to be used by the client. But we can send a message to all the other web sockets to inform them 'I'm doing dedicated work for a certain iframe, you can't handle any work until

I'm done', and this will handle multiple tabs.

#12 - 09/28/2017 03:11 PM - Greg Shah

This task is designed for implementing the locally installed "listening client" mode (this must work for multiple users, each user will have their own listening client process running on their dedicated system).

The [#3288](#) task is for adding the multiple iframe/websocket support.

#13 - 09/28/2017 03:11 PM - Greg Shah

- Related to Feature #3288: support > 1 browser tabs for a single FWD java client instance added

#14 - 10/10/2017 01:22 PM - Sergey Ivanovskiy

Does it make sense to add a support for this parameter client:cmd-line-option:startup-procedure within this task? For an example, this get request <https://localhost:7443/gui?startup%2Dprocedure=hello%2Ep> will be used to invoke hello.p program.

#15 - 10/12/2017 03:53 AM - Sergey Ivanovskiy

Created new task branch 3287a.

#16 - 10/12/2017 04:56 AM - Sergey Ivanovskiy

We already implemented the proxy server support when client requests to <https://proxyserver> are redirected to <https://proxyserver/client.i>. Is it possible to use this functionality if the proxyserver is localhost here? It needs to install proxy server and now we can only use the Apache web server. I just need an understanding what should be done here.

#17 - 10/12/2017 06:38 AM - Sergey Ivanovskiy

Greg, please clarify the user environment. We have one server on which P2J server and Embedded Web application are executed and a client on which P2J client must be executed and this P2J client must be accessed via a web interface using this localhost as a server, correct?

#18 - 10/12/2017 07:23 AM - Sergey Ivanovskiy

Greg Shah wrote:

We know that customers can run the swing client locally and we could support these use cases as is. Since the Swing client runs on the desktop system for each user, it is running on the same system on which the old 4GL Windows GUI application was previously running. This means that all of the native Windows dependencies can be directly accessed. DLLs can be called, COM Automation objects can be invoked and so forth. This is an option for customers so that they don't have to remove all the Windows dependencies in advance.

The idea of this task is to make changes to our client to allow the FWD client process to run locally (on the end-user's system), but where the FWD web client is executed instead of the FWD Swing client. This could easily be made to work for virtual desktop mode, but that wouldn't have any manually written web screens either.

I don't understand how it can work for virtual desktop mode. Greg, could you clarify your vision.

#19 - 10/12/2017 08:17 AM - Sergey Ivanovskiy

Did you mean to configure a broker client ([Remote Launchers](#)) on each user's host? I didn't test this kind of setup and it is not clear for me now how to use it.

#20 - 10/12/2017 08:19 AM - Greg Shah

Sergey Ivanovskiy wrote:

Does it make sense to add a support for this parameter `client:cmd-line-option:startup-procedure` within this task? For an example, this get request `https://localhost:7443/gui?startup%2Dprocedure=hello%2Ep` will be used to invoke `hello.p` program.

I don't want to take too much time with this, but if it can be done quickly then I guess it is OK. I'm worried about our deadline. This feature is useful but not required.

#21 - 10/12/2017 08:23 AM - Greg Shah

Sergey Ivanovskiy wrote:

We already implemented the proxy server support when client requests to <https://proxyserver> are redirected to https://proxyserver/client_i. Is it possible to use this functionality if the proxyserver is localhost here? It needs to install proxy server and now we can only use the Apache web server. I just need an understanding what should be done here.

No, I don't think the proxy will work. The idea is that we actually must spawn the FWD Client JVM on the local system where the user is "sitting". This way it can launch child programs, call COM automation objects, call DLLs/shared libraries and all the other client-platform functions that must run there. For example, perhaps they have a device like a printer that must be used for receipts or labels. That cannot be run in the cloud. It is a local device and the FWD Client is perfectly capable of using these devices or other local features, as long as it is running on the system on which the local features exist.

No proxy is needed or wanted here.

#22 - 10/12/2017 08:23 AM - Greg Shah

Sergey Ivanovskiy wrote:

Greg, please clarify the user environment. We have one server on which P2J server and Embedded Web application are executed and a client on which P2J client must be executed and this P2J client must be accessed via a web interface using this localhost as a server, correct?

Correct.

#23 - 10/12/2017 08:30 AM - Greg Shah

I don't understand how it can work for virtual desktop mode. Greg, could you clarify your vision.

Whether it is embedded mode or virtual desktop mode should not matter. Both cases could have their FWD Client JVM running locally with a remote FWD server (and in the embedded mode case, the remote embedded web application).

Please explain which part of this you think is not possible and why.

#24 - 10/12/2017 08:36 AM - Greg Shah

Sergey Ivanovskiy wrote:

Did you mean to configure a broker client ([Remote Launchers](#)) on each user's host? I didn't test this kind of setup and it is not clear for me now how to use it.

That is what I called "Option 1" in [#3287-2](#). In that case we have a remote spawner that connects to the FWD server and waits for the FWD server to tell it to launch a client.

I fact this is probably the easiest thing to implement since we already have the infrastructure there.

We also have the idea of Option 2, which we discussed above. This is intended to be a special kind of "client that is waiting to authenticate". It would require more extensive changes but would have the advantage of being only a single process to have running on the remote system. Although it has that advantage, it also brings more risk because it involves changes in the authentication process itself, which are quite tricky.

For this reason, I think we should focus on making the remote spawner work well for multiple clients. One part of this will be making sure that the user's login maps to the right client system. I seem to recall that our remote spawner implementation might also be limited to a single remote system right now and that would need to be resolved.

Constantin: can you list the things you know that need to be done for the remote spawner approach?

#25 - 10/12/2017 09:23 AM - Sergey Ivanovskiy

Greg Shah wrote:

I don't understand how it can work for virtual desktop mode. Greg, could you clarify your vision.

Whether it is embedded mode or virtual desktop mode should not matter. Both cases could have their FWD Client JVM running locally with a remote FWD server (and in the embedded mode case, the remote embedded web application).

Please explain which part of this you think is not possible and why.

Let us consider the case of the embedded web application that uses an iframe element to include a js embedded web client that accesses a remote P2J web client by url. This remote P2J web client is executed as a separate process (spawned process) on the P2J server host, correct?

#26 - 10/12/2017 09:26 AM - Greg Shah

This remote P2J web client is executed as a separate process (spawned process) on the P2J server host, correct?

Today when we run the web client, this is the most common case. But it is not required. The spawner could be remote on some other system.

In a "local web client" installation, the separate spawned process would definitely be remote, and it would be on the user's system instead of a shared server.

#27 - 10/12/2017 09:34 AM - Sergey Ivanovskiy

I don't understand how to launch the web client on the different host. How to setup this configuration? For an example, P2J server is started on the host1, P2J web client must be started on the host2 not on the same host, host1. The user uses this url <https://host1:7443/gui> to access to the P2J web client that must be spawned on the host2? How to setup this configuration?

#28 - 10/12/2017 09:44 AM - Sergey Ivanovskiy

I would like to test what are remote launchers and what are brokers or application agents in order to evaluate working functionality. Constantin, Greg, could you help me with starting configurations?

#29 - 10/12/2017 10:08 AM - Greg Shah

I don't have one that is already created. Perhaps Constantin has something.

If not, you should use [Spawner](#) and [Remote Launchers](#) to create a working installation.

#30 - 10/12/2017 10:22 AM - Sergey Ivanovskiy

Thank you, planning to use this manual.

#31 - 10/12/2017 04:03 PM - Sergey Ivanovskiy

We should check the following scenario for the embedded web application. A user loads an embedded js application client. The P2J server on this request creates target remote spawned web clients on the user's host and on web page load event this embedded js application client loads target js embedded P2J clients by iframe using localhost. Does the browser security policy permit this scenario and execute js scripts from the different domains: the server domain and localhost domains (one domain per a P2J web client)?

#32 - 10/13/2017 02:19 AM - Sergey Ivanovskiy

But this schema doesn't work at all now because the embedded web driver uses WebClientLauncher (WebHandler) to launch P2J web client. WebHandler needs a broker support somehow to launch P2J web client via brokers. I don't know at the moment how to implement it fitting the existing code.

#33 - 10/13/2017 03:31 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

But this schema doesn't work at all now because the embedded web driver uses WebClientLauncher (WebHandler) to launch P2J web client. WebHandler needs a broker support somehow to launch P2J web client via brokers. I don't know at the moment how to implement it fitting the existing code.

No, it is not correct. It seems that ClientSpawner has a broker support such that a user name is mapped to its broker.

#34 - 10/13/2017 03:40 AM - Sergey Ivanovskiy

OK, according to [Remote Launchers](#) there are limitations for remote launches: working directory and ip addresses must be assessed from the user browser. It is an interesting case if the proxy server exists and the user is trying to reach the owned host remotely. The proxy server has this mapping an external client name into the owned backend host.

#35 - 10/13/2017 03:43 AM - Sergey Ivanovskiy

It seems that we don't need to develop code there except the broker client installation schema and the server host configuration with proxy server and without proxy server if it was used inside the customer's Intranet.

#36 - 10/13/2017 04:10 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

It seems that we don't need to develop code there except the broker client installation schema and the server host configuration with proxy server and without proxy server if it was used inside the customer's Intranet.

I would like to recall that we implemented only the localhost launch with its ports range. Thus we need to extend it to any host to allocate web client on the external host. It seems I understand what we need to develop. The code for WebClientManager must be changed in order to support the schema with remotely spawned P2J web clients. I think we don't need a multiple web socket support at this moment because it is a completely different idea to use a lightweight web client associated with its web socket. I think it will require at least one working month only to develop this schema.

#37 - 10/13/2017 08:01 AM - Sergey Ivanovskiy

Greg, do you agree with this schema that launches many P2j web clients on the remote host and doesn't use multiple web sockets for one P2J web client?

#38 - 10/13/2017 08:10 AM - Greg Shah

Sergey Ivanovskiy wrote:

Greg, do you agree with this schema that launches many P2j web clients on the remote host and doesn't use multiple web sockets for one P2J web client?

For the purpose of this task, yes. We will work on [#3288](#) separately.

#39 - 10/17/2017 03:52 AM - Sergey Ivanovskiy

I tried to prepare configuration or the broker process but didn't succeed. The P2J server logged

```
58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Received certificate of broker1 account
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Certificate has alias <broker1>
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Associated process accounts: 1
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Associated user accounts: 0
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Associated process account is <
broker1_process>
[10/17/2017 10:37:58 MSK] (SecurityManager:FINE) {00000000:00000011:standard} hook class is com.goldencode.p2j
.security.GuestAccess
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000009:00000011:broker1_process} Search open: resId 0, in
stance logon, mode 0, handle 72
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000009:00000011:broker1_process} Search next: handle 72,
rights {true}
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000009:00000011:broker1_process} Search done: handle 72,
decision true, cache false
[10/17/2017 10:37:58 MSK] (SecurityManager:FINER) {00000000:00000011:standard} Sending AUTHRESULT0 (action = 0
)
[10/17/2017 10:37:58 MSK] (SecurityManager:SEVERE) {00000000:00000011:standard} Error sending AUTHRESULT java.
net.SocketException: Broken pipe (Write failed)
```

but the broker client logged that

```
Oct 17, 2017 10:36:37 AM BrokerCore.start()
INFO: {main} Connecting (9) ...
Oct 17, 2017 10:36:38 AM BrokerCore.connect()
SEVERE: {main} Connection Exception.
java.lang.Exception: Authentication failed
    at com.goldencode.p2j.net.SessionManager.createQueue(SessionManager.java:1130)
    at com.goldencode.p2j.net.LeafSessionManager.connectDirect(LeafSessionManager.java:249)
    at com.goldencode.p2j.main.BrokerCore.connect(BrokerCore.java:432)
    at com.goldencode.p2j.main.BrokerCore.start(BrokerCore.java:189)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:246)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
```

until reconnect tries were run out. It seems that there must be a user account that can run this broker process on the server side but documentation said that the user must be OS user for the external system to be able to launch remote spawner process on the server broker peer side. The next issue I encountered is that com.goldencode.p2j.security.SSLCertGenUtil can generate passwords (to store private keys and certificates for standard server and broker1 process) such that they can contain xml entities that must be escaped, & -> \$amp; > -> > < -> <. Thus they can't be included in

client.xml file without escaping them(to start this broker client).

#40 - 10/17/2017 04:16 AM - Sergey Ivanovskiy

Greg, Constantin, it seems that security model is not clear for me now. Could you help to understand how to associate user account with this broker process?

The standard server has these settings under /acl/accounts

```
<node class="container" name="processes">
  <node class="process" name="standard">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="P2J server"/>
    <node-attribute name="master" value="TRUE"/>
    <node-attribute name="server" value="TRUE"/>
    <node-attribute name="alias" value="standard"/>
  </node>
  <node class="process" name="broker1_process">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="broker1_process"/>
    <node-attribute name="server" value="FALSE"/>
    <node-attribute name="alias" value="broker1"/>
    <node-attribute name="broker" value="broker1"/>
  </node>
</node>
```

and these ones under /server/default

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="bogus"/>
    <node-attribute name="account" value="admin"/>
    <node-attribute name="account" value="sbi"/>
  </node>
</node>
```

#41 - 10/17/2017 04:21 AM - Sergey Ivanovskiy

The client.xml has these settings

```
<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="true" />
    <truststore filename="srv-certs.store" />
    <truststore alias="p2j-root" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjy1lQtPs_EgvIKZ!" />
    <password keystore="XL57uun0Aod4(7xniwAg6e`1a0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNuRfPJ6swd_2&lt;JEGQX6"/>
    <password masterkeyentry="5xYW$9>9@bvqiD9Fh)nWilXWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
  </remote>
</node>
```

and server.xml is

```
<node type="server">
  <net>
    <router threads="2"/>
  </net>

  <security>
    <server id="standard"/>
    <keystore filename="standard-private-key.store"/>
    <keystore alias="standard"/>
  </security>

  <directory>
    <backend type="xml"/>
    <xml filename="directory.xml"/>
  </directory>

  <access>
    <password masterkeyentry="5xYW$9&gt;9@bvqiD9Fh)nWilXWgMDE939Q"/>
    <password keystore="&QF4g&TSaA6AHBAo1dMw7%rEvI7=332dsGJE"/>
    <password keyentry="5seEse61i7HI1P3?pSAQHdaU4=2XaY`LbW^s"/>
  </access>
</node>
```

I did this

```
<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="false" />
    <truststore filename="srv-certs.store" />
    <truststore alias="standard" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjy1lQtPs_EgvIKZ!" />
    <password keystore="XL57uunOAod4(7xnIwAg6e`la0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNUrFpJ6swd_2&lt;JEgQX6"/>
    <password masterkeyentry="5xYW$9>9@bvgiD9Fh)nWilXWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
  </remote>
</node>
```

and changed this acl/system to permit only to this process to access accounts resources

```
<node class="container" name="000500">
  <node class="strings" name="subjects">
    <node-attribute name="values" value="broker1_process"/>
  </node>
  <node class="systemRights" name="rights">
    <node-attribute name="check" value="true"/>
  </node>
  <node class="resource" name="resource-instance">
    <node-attribute name="reference" value="accounts"/>
    <node-attribute name="reftype" value="TRUE"/>
  </node>
</node>
```

and now the server logged

```
[10/17/2017 11:38:44 MSK] (SessionManager.listen():WARNING) {00000000:00000001:standard} INSECURE sockets in use!
[10/17/2017 11:38:44 MSK] (SessionManager.listen():INFO) {00000000:00000001:standard} Server ready
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Received certificate of broker1 account
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Certificate has alias <broker1>
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Associated process accounts: 1
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Associated user accounts: 0
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Associated process account is <broker1_process>
[10/17/2017 11:39:02 MSK] (SecurityManager:FINE) {00000000:00000009:standard} hook class is com.goldencode.p2j.security.GuestAccess
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000001:00000009:broker1_process} Search open: resId 0, in stance logon, mode 0, handle 64
```

```
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000001:00000009:broker1_process} Search next: handle 64,
rights {true}
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000001:00000009:broker1_process} Search done: handle 64,
decision true, cache false
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Sending AUTHRESULT0 (action = 0
)
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Sent AUTHRESULT
[10/17/2017 11:39:02 MSK] (SecurityManager:FINER) {00000000:00000009:standard} Sending LOCALUSERID: null
```

and it seems the broker client is connected

```
Oct 17, 2017 11:39:01 AM BrokerCore.start()
INFO: {main} Connecting (9) ...
Oct 17, 2017 11:39:12 AM BrokerCore.connect()
INFO: {main} Broker has been registered uid=ed4fb5e1-f0ab-4f04-911d-18399102b8c4
```

but the directory settings are not clear for me yet

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="broker1_process"/>
  </node>
</node>
```

Is it correct? Or Should it be a OS account here?

#43 - 10/17/2017 05:07 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

```
I did this
[...]
and changed this acl/system to permit only to this process to access accounts resources
[...]
and now the server logged
[...]
and it seems the broker client is connected
[...]
but the directory settings are not clear for me yet
[...]
Is it correct? Or Should it be a OS account here?
```

No, the os account must be here

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="sbi"/> <!-- The OS account -->
  </node>
</node>
```

and the client looks like this one

```
<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="true" />
    <truststore filename="srv-certs.store" />
    <truststore alias="standard" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjy1lQtPs_EgvIKZ!" />
    <password keystore="XL57uunOAod4(7xniwAg6e`1a0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNuRfPJ6swd_2&lt;JEgQX6"/>
    <password masterkeyentry="5xYW$9>9@bvgid9Fh)nWilXWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
  </remote>
</node>
```


#44 - 10/17/2017 05:09 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

No, the os account must be here

Yes, OS account or FWD account; the FWD will redirect any spawn request for a OS account or FWD account to a broker which has a matching user OR has no users (i.e. 'all users are spawned by this broker').

#45 - 10/17/2017 06:04 AM - Sergey Ivanovskiy

OK, understand, thank you. I have a new issue.

The P2J web client spawned by the broker client is failed with web socket error for unknown reasons. If the broker was started from IDE in the debug mode, then it uses this mode to start spawn process incorrectly, because it tried to use the debug port opened by the broker client. After fixing this issue (temporarily removed JVM parameters propagation) the spawn process is started but the web socket is failed. Investigating the root causes. It seems `-Djava.library.path=/home/sbi/projects/3287a/build/lib/` was not set up during this launch due to this exception

```
Oct 17, 2017 1:45:32 PM GenericWebServer.startup
INFO: {main} Server URL: https://localhost:7450/
java.lang.InterruptedExce...tion: sleep interrupted
    at java.lang.Thread.sleep(Native Method)
    at com.goldencode.p2j.ui.client.driver.web.WatchdogTimer.run(WatchdogTimer.java:128)
java.lang.UnsatisfiedLinkError: no p2j in java.library.path
    at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1867)
    at java.lang.Runtime.loadLibrary0(Runtime.java:870)
    at java.lang.System.loadLibrary(System.java:1122)
    at com.goldencode.p2j.main.ClientCore.loadNativeLibrary(ClientCore.java:432)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:199)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:158)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
java.lang.InterruptedExce...tion
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.waitForMessages(PushMessagesWorker.java:187)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.run(PushMessagesWorker.java:144)
java.lang.InterruptedExce...tion
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.WebTaskWorker.run(WebTaskWorker.java:110)
```

#46 - 10/17/2017 07:24 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

OK, understand, thank you. I have a new issue.

The P2J web client spawned by the broker client is failed with web socket error for unknown reasons. If the broker was started from IDE in the debug mode, then it uses this mode to start spawn process incorrectly, because it tried to use the debug port opened by the broker client. After fixing this issue (temporarily removed JVM parameters propagation) the spawn process is started but the web socket is failed. Investigating the root causes. It seems `-Djava.library.path=/home/sbi/projects/3287a/build/lib/` was not set up during this launch due to this exception [...]

This looks like a path issue, assuming FWD has the `libp2j.so` library compiled (the native task). Have you set the `clientConfig/libPath` in the directory, in `server/default` section?

#47 - 10/17/2017 01:58 PM - Sergey Ivanovskiy

Yes, I set it, the `BrokerCore` uses its own parameter when it starts spawn process. In general I checked that the schema is workable, but it needs to fix `BrokerCore`, may be it is possible to reuse `ClientBuilder`, but now I didn't decide how it should be done properly. I tested P2J standard server and its remote broker client that spawns the local clients successfully after the command line fix. The spawn was initiated from the P2J server on web requests to the target resource.

#48 - 10/17/2017 02:04 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Yes, I set it, the `BrokerCore` uses its own parameter when it starts spawn process. In general I checked that the schema is workable, but it needs to fix `BrokerCore`, may be it is possible to reuse `ClientBuilder`, but now I didn't decide how it should be done properly. I tested P2J standard server and its remote broker client that spawns the local clients successfully after the command line fix. The spawn was initiated from the P2J server on web requests to the target resource.

OK, I think `BrokerCore` is missing a library path (i.e `-Djava.library.path`).

The approach is to do something similar to how we pass the classpath; see these classes:

1. `BrokerCore.initialize` - save the broker's library path into a `BrokerCore.libPath` instance field
2. `BrokerCore.spawn` - it needs to pass the lib path via a `args.getEnvironment().put(BROKER_LIBPATH, libPath)`; call
3. `NativeSecureConnection.remote` - this needs to set `-Djava.library.path` to the path received via a i.e. `BrokerCore.BROKER_LIBPATH` argument.

#49 - 10/17/2017 03:25 PM - Sergey Ivanovskiy

Yes, understand.

I need to check another case when the web portal is used and the embedded clients are used. The embedded clients must be created on the host that runs the broker client. It looks like that it can be done only if the web portal logged user is associated with the broker process and the peer broker client is running on the remote host, then

the embedded clients can be spawned on the target remote host associated with the logged user. Is this connection schema required by the customer?

#50 - 10/17/2017 03:33 PM - Greg Shah

The customer is definitely going to use an embedded mode web client with multiple users that each run a separate local web client.

The embedded web application will need to associate users with the remote broker. I don't think we have figured out how that will work yet, but we need to come up with an approach.

#51 - 10/18/2017 11:14 AM - Sergey Ivanovskiy

Please evaluate this schema. Brokers are listed in the directory with associated users

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="user1"/> <!-- The OS account -->
  </node>
  <node class="broker" name="broker2">
    <node-attribute name="account" value="user2"/> <!-- The OS account -->
  </node>
  <node class="broker" name="broker3">
    <node-attribute name="account" value="user3"/> <!-- The OS account -->
  </node>
</node>
```

and associated processes

```
<node class="container" name="processes">
  <node class="process" name="broker1_process">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="broker1_process"/>
    <node-attribute name="server" value="FALSE"/>
    <node-attribute name="alias" value="broker1"/>
    <node-attribute name="broker" value="broker1"/>
  </node>
  <node class="process" name="broker2_process">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="broker2_process"/>
    <node-attribute name="server" value="FALSE"/>
    <node-attribute name="alias" value="broker2"/>
    <node-attribute name="broker" value="broker2"/>
  </node>
  <node class="process" name="broker3_process">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="broker3_process"/>
    <node-attribute name="server" value="FALSE"/>
    <node-attribute name="alias" value="broker2"/>
    <node-attribute name="broker" value="broker2"/>
  </node>
  .....
</node>
```

If a remote broker peer client is registered, then its remote address can be available for the server side. Thus we have a dynamic map of a user to its remote broker peer client.

It seems that BrokerClientServices can be extended to expose new getRemoteHost() that should return IP address of the host or the domain name of

the host if dns service is used inside the customer's network. And new static method `Map<String, String> getRemotePeerMap()` of `BrokerManager` can return a user to its registered remote broker client address map. The web clients will be launched by these registered broker clients and managed with help of `WebClientsManager` that will use this remote peer map to assign the host and port from the available ports assigned for usage on this host. The port range policy are set by the P2J server and propagated to all these remote broker peer hosts. The Apache reverse proxy can be used without changes with this client to backend map

```
client11  host1:7449
client21  host1:7450
client31  host1:7451
```

```
client12  host2:7449
client22  host2:7450
client32  host2:7451
```

```
client13  host3:7449
client23  host3:7450
client33  host3:7451
```

where host1 is associated with user1 and its available ports are from this range 7449...7451. Names client11, ...client33 identify the js web clients.

#52 - 10/18/2017 01:05 PM - Sergey Ivanovskiy

I found that it can be simplified since `client:web:host` can be returned by the web remote client executing `processTemporaryClient(String uuid, BootstrapConfig config)` with

```
TemporaryClient tempClient = spawner.getTemporaryClient(uuid);
driver = tempClient.doWork(spawner, uuid, config);
```

Thus it seems that this map from a user to host is not required to support by `WebClientsManager`.

#53 - 10/18/2017 01:08 PM - Sergey Ivanovskiy

I meant `client:web:host`.

#54 - 10/18/2017 01:43 PM - Sergey Ivanovskiy

There is one issue with this schema above

```
@Override
public void terminate(Session session)
{
```

```
if (!portsRestricted)
{
    return;
}
int peerNode = session.getRemoteAddress();
.....
```

Greg, could you help here that I don't know if it is possible to get the host from this `com.goldencode.p2j.net.Session` object or not?

#55 - 10/18/2017 01:51 PM - Sergey Ivanovskiy

Don't mind it doesn't need since I will use pair (`session.getNodeAddress`, `session.getRemoteAddress`).

#56 - 10/18/2017 03:25 PM - Sergey Ivanovskiy

Don't mind it doesn't need since I will use pair (`session.getNodeAddress`, `session.getRemoteAddress`).

It seems that `session.getRemoteAddress` is not unique and there can be two different remote clients with the same `session.getRemoteAddress`? Is it a true statement?
Greg, could you help?

#57 - 10/18/2017 03:34 PM - Greg Shah

It seems that `session.getRemoteAddress` is not unique and there can be two different remote clients with the same `session.getRemoteAddress`? Is it a true statement?

No. Our FWD client is implemented using the "direct" session in our networking model. This means that there is one and only one Queue instance for that client and the queue is dedicated to that client. This means that the remote node address represented by that queue is unique. It is an integer value that is assigned at the time that the queue is created and it does not change for the lifetime of that queue, which is also the lifetime of the client's session.

#58 - 10/18/2017 03:42 PM - Sergey Ivanovskiy

Is it possible that for these two sessions `server(host0)<-session1->client(host1)` and `server(host0)<-session2->client(host2)` with 3 different hosts the

following statement will be true?

```
session1.getRemoteAddress() == session2.getRemoteAddress()
```

#59 - 10/18/2017 03:45 PM - Sergey Ivanovskiy

Greg, thank you. It follows from your explanation that the answer is no.

#60 - 10/18/2017 04:12 PM - Greg Shah

Correct.

#61 - 10/18/2017 06:39 PM - Sergey Ivanovskiy

Committed revision 11181 changed to allocate new system resources in order to run new web client on the target host. The rest tasks are to fix private String getPortName(String host, int port) in order to be able to generate the configuration file for the Apache Web server as a reverse proxy and to set client:web:host to the client host address for spawned web client. This revision is working if run broker similarly to this command `java -cp ./home/sbi/projects/3287auid/lib/p2j.jar -Djava.library.path=/home/sbi/projects/3287a/build/lib com.goldencode.p2j.main.ClientDriver client.xml` with

```
<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="true" />
    <truststore filename="srv-certs.store" />
    <truststore alias="standard" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjy11QtPs_EgvIKZ!" />
    <password keystore="XL57uunOAod4(7xnivAg6e`1a0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNuRfPJ6swd_2&lt;JEgQX6"/>
    <password masterkeyentry="5xYW$9>9@bvgiD9Fh)nW11XWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
  </remote>
</node>
```

#62 - 10/19/2017 03:35 AM - Sergey Ivanovskiy

We can consider one broker process that serves many users. In this case we need to select broker client that is associated with its user to run a web client. It seems that we need to collect this mapping on the server side in the directory.

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="user1"/> <!-- The OS account -->
    <node-attribute name="account" value="user2"/> <!-- The OS account -->
    <node-attribute name="account" value="user3"/> <!-- The OS account -->
    <!-- users to hosts map ->
    <node-attribute name="user1" value="host1"/>
    <node-attribute name="user2" value="host2"/>
    <node-attribute name="user3" value="host3"/>
    <!-- ----- ->
  </node>
</node>
```

and use this map to assign spawned task to the particular broker remote client if this client is registered within the server. Greg, what do you think? Now we need one broker process per a user in order to start web client on the dedicated host.

#63 - 10/19/2017 12:19 PM - Sergey Ivanovskiy

Greg, please review 3287a (the current revision 11183 after rebasing). Dedicated hosts are given by this settings

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="user1"/>
    <node-attribute name="account" value="user2"/>
  </node>
<node class="container" name="hosts">
  <node class="string" name="user1">
    <node-attribute name="value" value="host1"/>
  </node>
  <node class="string" name="user1">
    <node-attribute name="value" value="host2"/>
  </node>
  <node class="string" name="user2">
    <node-attribute name="value" value="host1"/>
  </node>
</node>
</node>
```

and one process is associated with this broker - broker1

```
<node class="container" name="processes">
.....
  <node class="process" name="broker1_process">
    <node-attribute name="enabled" value="TRUE"/>
    <node-attribute name="description" value="broker1_process"/>
  </node>
</node>
```

```

    <node-attribute name="server" value="FALSE"/>
    <node-attribute name="alias" value="broker1"/>
    <node-attribute name="broker" value="broker1"/>
  </node>

```

```

</node>

```

The broker client.xml looks like this with new parameters to be able to launch in debug mode from IDE and with assigned web host

```

<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="true" />
    <truststore filename="srv-certs.store" />
    <truststore alias="standard" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjyl1QtPs_EgvIKZ!" />
    <password keystore="XL57uunOAod4(7xniwAg6e`1a0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNuRfPJ6swd_2&lt;JEgQX6"/>
    <password masterkeyentry="5xYW$9>9@bvgid9Fh)nWilXWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
    <spawner host="192.168.1.37"/>
    <java args="-Xmx1G -Djava.awt.headless=true -XX:MaxPermSize=64m -Djava.library.path=/home/sbi/projects/
3287a/build/lib/ -Dfile.encoding=UTF-8" />
    <java classpath="/home/sbi/projects/3287a/build/lib/p2j.jar" />
  </remote>
</node>

```


#64 - 10/20/2017 08:00 AM - Constantin Asofiei

Review for 3287a rev 11183:

1. LogicalTerminal is missing history entry
2. I would prefer to have a java library path defaulted to the broker's library path (same as we do with the classpath), as this would minimize the configuration. But this would require the OS user used to spawn the process to have access to the broker's paths.
3. can you explain why is remote:spawner:file required? I assume in trunk the broker was able to spawn without this explicit setting... the idea is I want to minimize the configuration required for the broker, to have good defaults.
4. what is the relation between remote:spawner:host and directory's brokers/hosts/user1/host1 setting? Why can't we just use the directory configuration and pass it to the broker? For me it looks better to have a centralized configuration then client-local configurations...

#65 - 10/20/2017 08:16 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Review for 3287a rev 11183:

1. LogicalTerminal is missing history entry

OK.

1. I would prefer to have a java library path defaulted to the broker's library path (same as we do with the classpath), as this would minimize the configuration. But this would require the OS user used to spawn the process to have access to the broker's paths.

If client.xml didn't provide these settings, then the broker library path is used. It helps to debug the broker client and the spawned process from IDE, otherwise the spawned process gets the broker settings and is failed at the start since the debug port is used by the broker client.

1. can you explain why is remote:spawner:file required? I assume in trunk the broker was able to spawn without this explicit setting... the idea is I want to minimize the configuration required for the broker, to have good defaults.

Don't know. I didn't change these settings. I will investigate if it is possible to run spawned process without this setting.

1. what is the relation between remote:spawner:host and directory's brokers/hosts/user1/host1 setting? Why can't we just use the directory configuration and pass it to the broker? For me it looks better to have a centralized configuration then client-local configurations...

It was done because we have a one broker process and a broker instance associated with this process and many broker clients, when the broker client is registered it gets its server uuid. Theoretically it can get its host but we have no low level API here, because Queue has IP address of communication channel but doesn't expose this to external usages. Directory settings can hold domain addresses and it will be required to resolve and to identify the registering broker client to assign its host from this list. Thus I decided to use assignment from the client side. It simplifies the server side and absent of the API makes this task difficult to implement.

Sergey Ivanovskiy wrote:

Constantin Asofiei wrote:

1. I would prefer to have a java library path defaulted to the broker's library path (same as we do with the classpath), as this would minimize the configuration. But this would require the OS user used to spawn the process to have access to the broker's paths.

If client.xml didn't provide these settings, then the broker library path is used. It helps to debug the broker client and the spawned process from IDE, otherwise the spawned process gets the broker settings and is failed at the start since the debug port is used by the broker client.

Ah, sorry, I understand now, library path is part of the broker's JVM arguments.

1. can you explain why is remote:spawner:file required? I assume in trunk the broker was able to spawn without this explicit setting... the idea is I want to minimize the configuration required for the broker, to have good defaults.

Don't know. I didn't change these settings. I will investigate if it is possible to run spawned process without this setting.

You're right, you didn't add this, I mistook something. Leave it as is for now.

1. what is the relation between remote:spawner:host and directory's brokers/hosts/user1/host1 setting? Why can't we just use the directory configuration and pass it to the broker? For me it looks better to have a centralized configuration then client-local configurations...

It was done because we have a one broker process and a broker instance associated with this process and many broker clients, when the broker client is registered it gets its server uuid. Theoretically it can get its host but we have no low level API here, because Queue has IP address of communication channel but doesn't expose this to external usages. Directory settings can hold domain address and it will be required to resolve and to identify the registering broker client to assign its host from this list. Thus I decided to use assignment from the client side. It simplifies the server side and absent of the API makes this task difficult to implement.

I assume the client's remote:spawner:host is used for the web client to know where to redirect? Or what is the scenario where this setting is needed?

I don't have a full picture on how these two settings, client cfg and directory, work in the spawning process.

Constantin Asofiei wrote:

1. what is the relation between remote:spawner:host and directory's brokers/hosts/user1/host1 setting? Why can't we just use the directory configuration and pass it to the broker? For me it looks better to have a centralized configuration then client-local configurations...

It was done because we have a one broker process and a broker instance associated with this process and many broker clients, when the broker client is registered it gets its server uuid. Theoretically it can get its host but we have no low level API here, because Queue has IP address of communication channel but doesn't expose this to external usages. Directory settings can hold domain address and it will be required to resolve and to identify the registering broker client to assign its host from this list. Thus I decided to use assignment from the client side. It simplifies the server side and absent of the API makes this task difficult to implement.

I assume the client's remote:spawner:host is used for the web client to know where to redirect? Or what is the scenario where this setting is needed?

I don't have a full picture on how these two settings, client cfg and directory, work in the spawning process.

It is important that this setting has been set, because it is used to assign client:web:host and this configuration settings are used in WebClientSpawner.TemporaryClientTask.doWork(Spawner spawner, String uuid, BootstrapConfig config) to allocate server resources for this host and port because we can use port ranges and proxy server to make possible to connect to this host host1 from the external network and from internal network too.

```
.....
    int    port = config.getInt("client", "web", "port", 0);
    String host = config.getString("client", "web", "host", null);

WebClientConfig webClientConfig = null;

    Object clientResources;
    try
    {
        clientResources = spawner.allocateClient(uuid, host);
    }
    catch (OutOfResources e)
    {
        throw new RuntimeException("All client sessions are allocated.", e);
    }
.....
```

If user1 has dedicated host host1, then the usage scenario without embedded server can be this

- 1) client --- request page -> server P2J(host0)
- 2) server P2J(host0) -- login page --> client
- 3) client --- provide user1 and password -> server P2J(host0)
- 4) server P2J(host0) --- assign task to broker client on the host1 --> Broker P2J client (host1) -- create new spawned web client --> P2J web client (host1)
- 5) Broker P2J client (host1) -- return the url to redirect ---> server P2J(host0)
- 6) server P2J(host0) -- return redirect to given url to host1 --> client
- 7) client -- request redirected page from new spawned web client --> P2J web client (host1)
- 8) P2J web client (host1) -- return target resources to --> client

#68 - 10/20/2017 08:44 AM - Constantin Asofiei

Sergey, do you have a scenario where remote:spawner:host and the host configured in the directory's broker for the OS user have different values?

#69 - 10/20/2017 08:48 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey, do you have a scenario where remote:spawner:host and the host configured in the directory's broker for the OS user have different values?

In this case user1 can't run spawned web client on this host because it is not assigned to this user. If this user has no assigned hosts, then BrokerManager runs web client on any available broker client or locally on the P2J host. I checked this scenario first.

#70 - 10/20/2017 09:13 AM - Greg Shah

It is important that when a spawner is dedicate to a specific user, that no other users can spawn FWD client instances using that spawner. Do we have sufficient protections in to assure this case?

#71 - 10/20/2017 09:22 AM - Sergey Ivanovskiy

Greg Shah wrote:

It is important that when a spawner is dedicate to a specific user, that no other users can spawn FWD client instances using that spawner. Do we have sufficient protections in to assure this case?

If the same host is not assigned to a different user, then BrokerManager.schedule can't assign this host to run on this host new web client. Now in the code of BrokerManager if user has no any assigned host, then BrokerManager schedules this task to spawn new web client on any available host on which the broker client is running now.

Greg, please clarify what are correct requirements for this task to run web client remotely. Two different user can spawn web clients on the same host if this host is assigned to these users.

#72 - 10/20/2017 10:26 AM - Greg Shah

I see 3 cases:

- **dedicated agent** is a dedicated single-user agent in which executing the spawner should be only allowed for the user on that same system, even if multiple FWD clients are spawned at the same time, they can only be used by that user
- **user-limited shared agent** is an agent that runs the spawner on a system that may run multiple simultaneous FWD clients with different users, so long as all of these users are explicitly associated with that spawner
- **unlimited shared agent** is an agent that runs the spawner on a system that may run multiple simultaneous FWD clients with different users, with no limits on which users can spawn clients

The dedicated agent case is the one we are working on for this task.

Some other thoughts:

- Can we reduce or eliminate the directory configuration by doing some kind of registration when a dedicated agent is launched? If we launch the agent in a special "dedicated mode" and pass the user account name that is to be mapped to that agent, then it seems like we could dynamically create the right mapping without directory configuration.
- Any session for that user account that comes from the same host on which exists a dedicated agent would only be allowed to spawn using that host's dedicate agent.
- No other user accounts are ever allowed on a dedicated agent, except for the specific account which is registered (at launch, hopefully).
- It should be possible to have multiple dedicated agents for the same user account at the same time. For example, the user has access to multiple systems, each with a dedicated agent.
- If a user with a dedicated agent registered connects from a host that does not have a dedicated agent, then normal non-dedicated agent rules apply (they would spawn using a non-dedicated agent). This means spawning on a user-limited shared agent that has that account listed or spawning on an unlimited shared agent).

#73 - 10/20/2017 10:38 AM - Sergey Ivanovskiy

Please review this branch 3287a 11184, it fixed getPortName (WebClientsManager) to create well-defined map from hosts and their ports to web clients identifiers. It fills a gap when Apache proxy server is used and assigned hosts are defined in the directory. Let me recall that the proxy server configuration requires to support a map of clients to backends.

For my configuration environment `~/projects/testcases/simple/server2$ java -cp ../../uast/p2j/build/lib/p2j.jar com.goldencode.p2j.main.ClientsToPortsGenerator ./directory.xml` generates this configuration map

```
client01    localhost:7449
client02    localhost:7450
client11    127.0.0.1:7449
client12    127.0.0.1:7450
client31    192.168.1.37:7449
client32    192.168.1.37:7450
```

#74 - 10/20/2017 10:50 AM - Sergey Ivanovskiy

Can we reduce or eliminate the directory configuration by doing some kind of registration when a dedicated agent is launched? If we launch the agent in a special "dedicated mode" and pass the user account name that is to be mapped to that agent, then it seems like we could dynamically create the right mapping without directory configuration.

I think it is difficult to implement now because for the Apache reverse proxy we need to support well-defined map from hosts to numbers. We can try to use ip numbers and resolve host names to IP4 addresses and then enumerates IP4 addresses. Do you know some well-defined enumeration function for IP4 addresses from customers network?

#75 - 10/20/2017 11:07 AM - Sergey Ivanovskiy

Greg, do these requirements force to create new directory entity as host. Now we have these entities from accounts: user, group, processes, certificates. Does it make sense to add host entity?

#76 - 10/20/2017 11:18 AM - Sergey Ivanovskiy

Can we reduce or eliminate the directory configuration by doing some kind of registration when a dedicated agent is launched? If we launch the agent in a special "dedicated mode" and pass the user account name that is to be mapped to that agent, then it seems like we could dynamically create the right mapping without directory configuration.

Does it mean that the broker client can add this "dedicated user to its host" entry to this "users to hosts" map? The problem is that we need to enumerate IP4 addresses from the customer's network. But the network can be complicated. It is possible to use this enumeration function from IP4 addresses to numbers:

a1.a2.a3.a4 -> a1a2a3a4 but it opens network topology to outside world.

#77 - 10/20/2017 11:26 AM - Greg Shah

I think we don't care about the external IP address or hostname because we are not really using this for connectivity. We are just trying to match the user's initial web connection to the FWD server with the dedicated agent that is on the same system.

For this reason, it seems like we can make some kind of unique string using a combination of the user's account name and the local hostname (or IP address if there is no hostname). For example, user1_somebox.internal.acme.com might be sufficiently unique.

That user would not be able to have more than 1 dedicated agents started on systems that have the same hostname or IP address. But that seems like a reasonable restriction.

#78 - 10/20/2017 11:27 AM - Greg Shah

The idea is that if the browser can forward enough information to calculate this value, then it can be used to lookup any existing dedicated agent.

#79 - 10/20/2017 11:54 AM - Sergey Ivanovskiy

Should these dedicated and user-limited shared agents save new added users and hosts in the directory?

#80 - 10/20/2017 12:07 PM - Greg Shah

Should these dedicated and user-limited shared agents save new added users and hosts in the directory?

The user-limited shared agents are already present in the directory. It is OK to leave them that way.

I think the dedicate agent is probably different. It don't see a reason that we need to store this in the directory. When the agent disconnects it should be removed from the list of available dedicated agents.

#81 - 10/20/2017 02:29 PM - Sergey Ivanovskiy

It seems that I don't understand this conception. Please explain what mean that user-limited shared agents are already present in the directory.

#82 - 10/20/2017 02:31 PM - Greg Shah

The current approach of allowing specific users to be associated with an agent is the implementation of this concept. It is not based on dynamic registration but is configured in the directory, right?

#83 - 10/20/2017 02:40 PM - Sergey Ivanovskiy

Yes. BrokerManager collects brokers for a given user if there are no such brokers, then it collects brokers without accounts and Brokers.isGeneric() returns true in this case.

Then BrokerManager selects broker agent to launch a client from this prepared list of broker agents for this user, Brokers. I understand your conception in this way that

BrokerManager first should try to spawn a client on dedicated agents, and if it is not successful, then it tries user-limited shared agents if there are such agents for this user, and then all other agents if it is not succeeded to launch user-limited shared agents, correct?

#84 - 10/20/2017 02:48 PM - Sergey Ivanovskiy

If my understanding is correct, then we can generate clients to backends map for proxy server dynamically. If new dedicated agent is registered, then new "clients to backends" file will be generated. Then we don't need to store this node broker/hosts in the directory. Correct?

#85 - 10/20/2017 02:49 PM - Greg Shah

Exactly right.

#86 - 10/20/2017 03:47 PM - Sergey Ivanovskiy

Let us consider this scenario. The generic broker without associated users registers its broker agent that proclaims that i am the dedicated agent for a given user. Next another broker agent tries to connect to the same broker and it proclaims that i am the dedicated agent for the same user. What to do in this case? What broker agent will launch new web client?

#87 - 10/20/2017 03:56 PM - Sergey Ivanovskiy

Greg, please explain more thoroughly. This rule

If a user with a dedicated agent registered connects from a host that does not have a dedicated agent, then normal non-dedicated agent rules apply (they would spawn using a non-dedicated agent). This means spawning on a user-limited shared agent that has that account listed or spawning on an unlimited shared agent).

is not clear for me now. What means "If a user with a dedicated agent registered connects from a host that does not have a dedicated agent".

#88 - 10/21/2017 01:31 AM - Sergey Ivanovskiy

Is it OK if the dedicated broker client will start web clients accessed only by the user from the same host on which dedicated client is running? This functionality could be implemented by starting web client with web:client:host=localhost in this case this user can't access this web client from the different host because the web client doesn't provide web service for network outside its host. What logic should be implemented if this user has its account in the different broker on which another user-defined broker client is registered and the user tries to access the target resource from the different host that doesn't run its own dedicated broker client? These questions block my development on this task.

#89 - 10/21/2017 02:59 AM - Sergey Ivanovskiy

I think that the embedded server use case can work only in this case. We have one user 'fwd_user' and from the local host a user starts a "dedicated" broker client. The scenario is this

- 1) user -- request embedded page resource --> embedded server (host0) --- connect to P2J server and starts procedure startup-procedure.p --> P2J server(host1)
- 2) P2J server(host1) -- must check the host from which the page is requested if this host runs dedicated broker agent, then it should launch web client on this host --> return url -> embedded server(host0)
- 3) embedded server (host0) -- return the started page with iframe that displays login page --> user
- 4) user -- enters login and password -->

Now the embedded server doesn't provide a host parameter if it launches procedure emain.p. Only the embedded server knows from what host this target resource is requested, so it must provide the requesting host to the P2J server with help of WebClientLauncher.

#90 - 10/21/2017 10:36 AM - Constantin Asofiei

Sergey, for embedded mode, there is no real knowledge of the authenticated FWD or OS user - the credentials you enter at this time are legacy credentials, from the 4GL DB or something else.

Greg, I think we need a way to distinguish this client, so that it ensures the FWD client is spawned by the local broker; can you see another way other than entering the OS user name somewhere? Consider that the same business logic username might be 'shared' among users... maybe let the embedded app determine the host who is making the request and use the broker associated with that host/IP?

#91 - 10/21/2017 01:02 PM - Sergey Ivanovskiy

I agree with Constantin that it seems there no ways to implement these requirements to start local clients in the case of the embedded server usage.

#92 - 10/23/2017 04:58 AM - Sergey Ivanovskiy

3287a is updated over 11181, current rev. 11186. Planning to fix a dynamic registration for dedicated broker agents and to generate client to hosts table for The Apache reverse proxy configuration, and to implement an identification of requesting clients in order to delegate a launch remote client to the correct dedicated broker agent.

#93 - 10/23/2017 07:53 AM - Greg Shah

Sergey Ivanovskiy wrote:

Let us consider this scenario. The generic broker without associated users registers its broker agent that proclaims that i am the dedicated agent for a given user. Next another broker agent tries to connect to the same broker and it proclaims that i am the dedicated agent for the same user. What to do in this case? What broker agent will launch new web client?

A dedicated agent can only be accessed from a browser session that is running on the same system. It is DEDICATED to the user on that system.

For this reason, the answer to your question depends on which system the browser is being used:

- Browser is on same system as a dedicated client 1, dedicated client 1 is used.
- Browser is on same system as a dedicated client 2, dedicated client 2 is used.
- Browser is on any other system, a dedicated client CANNOT be used and normal rules apply.

#94 - 10/23/2017 07:54 AM - Greg Shah

Is it OK if the dedicated broker client will start web clients accessed only by the user from the same host on which dedicated client is running?

It is REQUIRED to work this way.

#95 - 10/23/2017 07:58 AM - Greg Shah

This functionality could be implemented by starting web client with `web:client:host=localhost` in this case this user can't access this web client from the different host because the web client doesn't provide web service for network outside its host.

Of course it is fine to exclude outside network access.

#96 - 10/23/2017 08:27 AM - Greg Shah

I think we need a way to distinguish this client, so that it ensures the FWD client is spawned by the local broker; can you see another way other than entering the OS user name somewhere? Consider that the same business logic username might be 'shared' among users... maybe let the embedded app determine the host who is making the request and use the broker associated with that host/IP?

The embedded application will have to provide enough information to uniquely identify the user + broker/host combination. It will automatically know the IP address/hostname of the connecting browser, so it seems it would only need to know the user name.

Let's back up a bit here. The core idea of the task is to make it possible (and easy) for a user to run an embedded FWD web client on the same system on which their browser process exists. That browser process will have a session with the remote embedded web application which would often be on the same system as the FWD server.

We don't have to use a broker model here. If it is a clean and secure implementation to use the broker approach, do it.

But if this is not working, then we can take an alternate approach. For example, we can allow the user to launch the FWD embedded web client directly. This has multiple advantages:

1. There is no need for an OS authentication because we don't have to spawn a process. The process naturally runs inside of an existing OS user's account and context.
2. The FWD client would be launched in dedicated mode. The dedicated FWD client could create a unique authentication token that can be passed to the FWD server AND can be used to identify the user's browser session to the embedded web application. In other words, they could use this automatically generated token as a kind of one-time use value that connects the client to the embedded web application user.

3. The FWD client could even launch the user's browser, redirecting it to the embedded web application URL and the authentication token can be a parameter in the URL.

4. To ensure security, the FWD client must be authenticated with the FWD server in some way. I can see these valid approaches:

- The FWD client is configured with a certificate that allows non-interactive authentication with the server. This adds a burden of extra configuration and management, so it is not the best option. Any breach of the security of the client system may also breach the FWD application environment.
- The FWD client requires its own interactive password login. This is less configuration/management, but it requires multiple logins for the user which is not a great experience.
- The embedded web application handles authentication of the user and tells the FWD server to allow access for the client that passes the authentication token AND to associate it with a specific FWD account. This is similar to the approach that we are already using for the embedded mode. It requires very little configuration and only needs a mapping of the embedded web application user to the FWD account. This can be considered a variant of the trusted spawning approach we already use.

#97 - 10/23/2017 05:52 PM - Sergey Ivanovskiy

Please review the committed revision 11187 that used brokers to start web clients.

ADDED LATER

The dedicated broker client has this similar configuration

```
<node type="client">
  <client>
    <mode broker="true" />
  </client>

  <net>
    <server host="192.168.1.37" />
    <server port="3333" />
    <server secure_port="3334" />
    <connection secure="true" />
  </net>

  <security>
    <certificate validate="true" />
    <truststore filename="srv-certs.store" />
    <truststore alias="standard" />
    <keystore filename="broker1-private-key.store" />
    <keystore processalias="broker1" />
    <authentication type="program"/>
  </security>

  <access>
    <password truststore="d61pi96tP8MS!T^87Efcjy1lQtPs_EgvIKZ!" />
    <password keystore="XL57uun0Aod4(7xniwAg6e`1a0Su~)g8DRmX" />
    <password keyentry="g8j4=mxVd*063QskHNuRfPJ6swd_2&lt;JEGQX6"/>
    <password masterkeyentry="5xYW$9>9@bvgid9Fh)nWilXWgMDE939Q" />
  </access>

  <remote>
    <retry count="10" />
    <retry seconds="10" />
    <spawner file="/home/sbi/opt/spawner/spawn" />
    <agent host="192.168.1.37"/>
    <agent user="user1"/>
    <agent dedicatedMode="true"/>
    <java args="-Xmx512m -Djava.awt.headless=true -XX:MaxPermSize=64m -Djava.library.path=/home/sbi/projects/3287a/build/lib/ -Dfile.encoding=UTF-8" />
    <java classpath="/home/sbi/projects/3287a/build/lib/p2j.jar" />
  </remote>
</node>
```

For the embedded server use case the user should be the FWD user used for the trusted access. It can be OS or P2J user. If broker1 has an account user1 and dedicatedMode=false, then new spawned web clients can be accessed by this user user1 from the network. If dedicatedMode=true, then all spawned web clients are only local accessible from this agent host.

#98 - 10/24/2017 01:35 AM - Sergey Ivanovskiy

11187 covers these cases

dedicated agent is a dedicated single-user agent in which executing the spawner should be only allowed for the user on that same system, even if multiple FWD clients are spawned at the same time, they can only be used by that user

user-limited shared agent is an agent that runs the spawner on a system that may run multiple simultaneous FWD clients with different users, so long as all of these users are explicitly associated with that spawner

unlimited shared agent is an agent that runs the spawner on a system that may run multiple simultaneous FWD clients with different users, with no limits on which users can spawn clients

Now working on the reverse proxy use case. Planning to implement this schema. Now broker agents are registered dynamically. It means that the new registered agent can update the "clients to backends" map. If new agent is registered this file hosts.txt will be updated, it should contain the host IP address and its registered order number.

```
127.0.0.1 1
192.168.1.37 2
.....
```

This file is built at the application run-time. ClientsToPortsGenerator should use the path to this file in order to generate correct "clients to backends" map that reflects the current changes. The Apache server should be restarted in order new changes become effective. This schema has limitations but I think it is workable because it covers the case when a user tries to access to its own broker agent from outside of its own private network. So these prerequisites should be satisfied. Its own broker agent should be already registered on the running P2J server and the Apache configuration should be prepared.

#99 - 10/24/2017 02:07 AM - Sergey Ivanovskiy

Added rev 11187 explanations to help reviewers. Please look at [#3287-97](#).

#100 - 10/24/2017 04:38 AM - Sergey Ivanovskiy

Committed revision 11188 implemented the hosts file. Please review. The next file for planning changes is only ClientsToPortsGenerator. The regression testing for 3287a can be done in parallel if these changes, rev. 11188 and 11187, will be approved.

#101 - 10/24/2017 09:48 AM - Greg Shah

Constantin will review the code and provide feedback.

One concern I have: the need to restart the proxy server seems to be a problem. Dedicated clients will be coming and going often. The user often has the ability to reboot their workstation at any time. If this requires the proxy server to be recycled each time, that will not be workable in a production environment. What can we do to eliminate this?

#102 - 10/24/2017 10:14 AM - Sergey Ivanovskiy

Greg Shah wrote:

One concern I have: the need to restart the proxy server seems to be a problem. Dedicated clients will be coming and going often. The user often has the ability to reboot their workstation at any time. If this requires the proxy server to be recycled each time, that will not be workable in a production environment. What can we do to eliminate this?

We can persist hosts mapping or prepare it before when the environment will be setup. The hosts file is a sequence of IP4 addresses

```
127.0.0.1 1
.....
```

This file can be prepared according to the network environment.

#103 - 10/24/2017 02:52 PM - Sergey Ivanovskiy

The 3287a is updated over rev 11182. The current revision 11190 fixed ClientsToPortsGenerator. I think that this branch must be in the trunk asap after the review and the regression testing.

#104 - 10/24/2017 03:05 PM - Sergey Ivanovskiy

Can I commit server2/ and broker/ configurations under testcases/simple with generated certificates and their log file?

#105 - 10/24/2017 03:05 PM - Greg Shah

Sergey Ivanovskiy wrote:

Can I commit server2/ and broker/ configurations under testcases/simple with generated certificates and their log file?

Yes.

#106 - 10/24/2017 03:18 PM - Constantin Asofiei

Review for 3287a rev 11190:

1. BrokerManager.removeUserHostWithCheck - shouldn't this have some synchronization? Same for getHostsForUser, getOrderedNumber

#107 - 10/24/2017 03:55 PM - Sergey Ivanovskiy

Yes, thank you for findings.

BrokerManager.removeUserHostWithCheck is used only from void deregisterBroker(BrokerParameters brokerAgent) and the last one is used from Broker.removeBroker(Session), and removeBroker is used from the listener that has a synchronization over activeBrokers. Thus BrokerManager.removeUserHostWithCheck is thread safe.

getHostsForUser is used from private static BrokerParameters schedule(final List<BrokerParameters> brokers, String dedicatedUser, String clientAddress) and schedule is not synchronized over activeBrokers. I will fix it now.

It seems that getOrderNumber uses ConcurrentHashMap and is used from WebClientManager.allocateClient initiated from the spawned web client. Thus it is a safe usage because this order map has a target entry already since the broker agent is registered first.

#108 - 10/24/2017 04:03 PM - Sergey Ivanovskiy

Committed revision 1677 in testcase project added configurations for the standard server with broker and for its broker client. The personal settings in the directory must be changed but certificates can be used as they are committed.

#109 - 10/25/2017 03:05 AM - Sergey Ivanovskiy

Committed 11194 fixed getHostsForUser to be a thread safe. Planning to setup broker for the embedded web application. It seems that the existing certificates and the root certificate can be reused by com.goldencode.p2j.security.SSLCertGenUtil and new certificate and private key for the broker process can be generated?

#110 - 10/25/2017 05:08 AM - Sergey Ivanovskiy

The broker has been setup successfully with help of com.goldencode.p2j.security.SSLCertGenUtil and certificates have been created for the all aliases and new alias for the broker process has been added. The CA private key and certificate has been reused. The dedicated broker client can be used and it can spawn web clients successfully. The login page for poc was displayed, then providing credentials produced ./usermenu.p was not found exception

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/sbi/projects/3287a/build/lib/slf4j-jdk14-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/sbi/projects/3287a/build/lib/slf4j-simple-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Oct 25, 2017 11:54:27 AM org.eclipse.jetty.util.log.Log initialized
INFO: Logging initialized @756ms
Oct 25, 2017 11:54:27 AM WebPageHandler
INFO: {main} target=/index.html page=/com/goldencode/p2j/ui/client/driver/web/index.html
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.Server doStart
INFO: jetty-9.3.z-SNAPSHOT
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.handler.ContextHandler doStart
INFO: Started o.e.j.s.h.ContextHandler@12028586{/ ,null,AVAILABLE}
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.handler.ContextHandler doStart
INFO: Started o.e.j.s.h.ContextHandler@2d127a61{/client, jar:file:/home/sbi/projects/3287a/build/lib/p2j.jar!/com/goldencode/p2j/ui/client/gui/driver/web/res,AVAILABLE}
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.handler.ContextHandler doStart
INFO: Started o.e.j.s.h.ContextHandler@2bbaf4f0{/common, jar:file:/home/sbi/projects/3287a/build/lib/p2j.jar!/com/goldencode/p2j/ui/client/driver/web/res,AVAILABLE}
```

```

Oct 25, 2017 11:54:28 AM org.eclipse.jetty.util.ssl.SslContextFactory doStart
INFO: x509=X509@120d6fe6(standard,h=[localhost],w=[]) for SslContextFactory@4ba2ca36(null,null)
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.AbstractConnector doStart
INFO: Started ServerConnector@7364985f{SSL,[ssl, http/1.1]}{localhost:7449}
Oct 25, 2017 11:54:28 AM org.eclipse.jetty.server.Server doStart
INFO: Started @992ms
Oct 25, 2017 11:54:28 AM GenericWebServer.startup
INFO: {main} Server URL: https://localhost:7449/
java.lang.InterruptedExcepion: sleep interrupted
    at java.lang.Thread.sleep(Native Method)
    at com.goldencode.p2j.ui.client.driver.web.WatchdogTimer.run(WatchdogTimer.java:128)
UI Theme successfully changed to 'windows10'
Oct 25, 2017 11:55:01 AM ErrorManager
SEVERE: {main} ** "./usermenu" was not found. (293)
descriptor: BitmapRGB[ICOEntry: width: 32, height: 32, colors: 0, planes: 1, bit count: 32, size: 4264, offset
: 29290, BitmapInfoHeader: size: 40 bytes, width: 32, height: 64, planes: 1, bit count: 32, compression: 0, im
age size: 4224, X pixels per m: 2835, Y pixels per m: 2835, colors used: 1 (unknown), colors important: 1 (all
)]
descriptor: BitmapRGB[ICOEntry: width: 32, height: 23, colors: 0, planes: 1, bit count: 32, size: 3076, offset
: 22, BitmapInfoHeader: size: 40 bytes, width: 32, height: 46, planes: 1, bit count: 32, compression: 0, image
size: 5888 (calculated), X pixels per m: 2835, Y pixels per m: 2835, colors used: 1 (unknown), colors importa
nt: 1 (all)]
Oct 25, 2017 11:55:22 AM ErrorManager
SEVERE: {main} ** "?" was not found. (293)
java.lang.InterruptedExcepion
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.waitForMessages(PushMessagesWorker.java:187)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.run(PushMessagesWorker.java:144)
java.lang.InterruptedExcepion
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.WebTaskWorker.run(WebTaskWorker.java:110)
Oct 25, 2017 11:55:24 AM org.eclipse.jetty.server.AbstractConnector doStop
INFO: Stopped ServerConnector@7364985f{SSL,[ssl, http/1.1]}{localhost:7449}
Oct 25, 2017 11:55:24 AM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped o.e.j.s.h.ContextHandler@2bbaf4f0{/common, jar:file:/home/sbi/projects/3287a/build/lib/p2j.jar!/c
om/goldencode/p2j/ui/client/driver/web/res,UNAVAILABLE}
Oct 25, 2017 11:55:24 AM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped o.e.j.s.h.ContextHandler@2d127a61{/client, jar:file:/home/sbi/projects/3287a/build/lib/p2j.jar!/c
om/goldencode/p2j/ui/client/gui/driver/web/res,UNAVAILABLE}
Oct 25, 2017 11:55:24 AM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped o.e.j.s.h.ContextHandler@12028586{/,null,UNAVAILABLE}

```

and the dedicated broker client produced

```

Oct 25, 2017 11:53:06 AM BrokerCore.start()
INFO: {main} Connecting (9) ...
Oct 25, 2017 11:53:06 AM BrokerCore.connect()
INFO: {main} Broker has been registered uid=3bb52929-08dd-41e1-be5e-5fd4bb04eb9b
Oct 25, 2017 11:54:27 AM BrokerCore.spawn()
INFO: {main} Command line arguments: [/home/sbi/opt/spawner/spawn, 1, sbi, /home/sbi/projects/hotel_gui/deploy
/client, java, -Xmx1G, -Djava.awt.headless=true, -XX:MaxPermSize=64m, -Djava.library.path=/home/sbi/projects/3
287a/build/lib/, -Dfile.encoding=UTF-8, -classpath, /home/sbi/projects/3287a/build/lib/p2j.jar, com.goldencode
.p2j.main.ClientDriver, server:spawner:uid=6afd6949-bee1-45e6-9efa-50bb8d56635f, net:server:host=192.168.1.37
, net:connection:secure=true, net:server:secure_port=3333, client:driver:type=gui_web, client:gui:desktopHeigh
t=768, client:gui:desktopWidth=1024, client:gui:taskbar=true, client:gui:graphicsCached=true, client:web:socket
Timeout=-1, client:web:watchdogTimeout=120000, client:web:maxBinaryMessage=32894, client:web:maxTextMessage=4
096, client:web:maxIdleTime=90000, client:web:embedded=false, client:web:port=0, web:referrer:url=https://192.
168.1.37:7443/gui, client:cmd-line-option:debugalert=true, client:web:host=192.168.1.37, client:web:dedicatedM
ode=true]
Oct 25, 2017 11:54:27 AM BrokerCore.spawn()
INFO: {main} Spawn exitValue=0
Oct 25, 2017 11:56:09 AM BrokerCore.spawn()
INFO: {main} Command line arguments: [/home/sbi/opt/spawner/spawn, 1, sbi, /home/sbi/projects/hotel_gui/deploy
/client, java, -Xmx1G, -Djava.awt.headless=true, -XX:MaxPermSize=64m, -Djava.library.path=/home/sbi/projects/3
287a/build/lib/, -Dfile.encoding=UTF-8, -classpath, /home/sbi/projects/3287a/build/lib/p2j.jar, com.goldencode
.p2j.main.ClientDriver, server:spawner:uid=66f7f83d-c0a8-470a-bce2-0fedf44ae60e, net:server:host=192.168.1.37
, net:connection:secure=true, net:server:secure_port=3333, client:driver:type=gui_web, client:gui:desktopHeigh
t=768, client:gui:desktopWidth=1024, client:gui:taskbar=true, client:gui:graphicsCached=true, client:web:socket
Timeout=-1, client:web:watchdogTimeout=120000, client:web:maxBinaryMessage=32894, client:web:maxTextMessage=4
096, client:web:maxIdleTime=90000, client:web:embedded=false, client:web:port=0, web:referrer:url=https://192.
168.1.37:7443/gui, client:cmd-line-option:debugalert=true, client:web:host=192.168.1.37, client:web:dedicatedM

```

```
ode=true]
Oct 25, 2017 11:56:09 AM BrokerCore.spawn()
INFO: {main} Spawn exitValue=0
```

#111 - 10/25/2017 05:12 AM - Sergey Ivanovskiy

I encountered this menu exception before, but its root cause was not fixed by me so I didn't remember what was the case. Constantin, please could you help if it is in your memory now.

#112 - 10/25/2017 07:52 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

I encountered this menu exception before, but its root cause was not fixed by me so I didn't remember what was the case. Constantin, please could you help if it is in your memory now.

Don't mind it can be resolved by adding usermenu folder under the working directory. What files are important for menu

```
total 92
drwxrwx--- 3 sbi sbi 4096 Oct 25 14:48 ./
drwxrwx--- 7 sbi sbi 4096 Oct 25 11:25 ../
-rw-rw---- 1 sbi sbi 3519 Oct 25 11:18 broker1-private-key.store
-rw-rw---- 1 sbi sbi 12803 Oct 25 11:50 broker_sbi_1508921330362.log
-rw-rw---- 1 sbi sbi 2447 Oct 25 11:56 broker_sbi_1508921586324.log
-rw-rw---- 1 sbi sbi 1328 Oct 25 14:48 broker_sbi_1508932070029.log
-rw-rw---- 1 sbi sbi 2835 Oct 25 14:48 client_sbi_1508932089386.log
-rw-rw---- 1 sbi sbi 1156 Oct 25 11:53 client.xml
-rw-rw---- 1 sbi sbi 9129 Jan 31 2017 menu.cor
-rw-rw---- 1 sbi sbi 9126 Jan 31 2017 menu.fol
-rw-rw---- 1 sbi sbi 9173 Jan 31 2017 menu.lst
-rw-rw---- 1 sbi sbi 228 Jan 31 2017 popup.lst
-rw-rw---- 1 sbi sbi 2240 Oct 25 11:20 srv-certs.store
drwxrwx--- 3 sbi sbi 4096 Jan 31 2017 usermenu/
```

#113 - 10/25/2017 08:51 AM - Sergey Ivanovskiy

Constantin, do you know how these parameters are used by the spawned web client in the embedded web application?

```
client:cmd-line-option:param=USERID=<CUSTOMER_INFORMATION_REDACTED>|PASSWORD=, client:cmd-line-option:startup-  
procedure=emain.p
```

#114 - 10/25/2017 09:17 AM - Sergey Ivanovskiy

There are some issues with js embedded part, the spawned web client response looks correct but the process is not going father this login point and iframe doesn't load application.

```
var xmlHttp = new XMLHttpRequest();  
xmlHttp.onreadystatechange = function()  
{  
  if (xmlHttp.readyState == 4 && xmlHttp.status == 200)  
  {  
    var fwd_url = xmlHttp.responseText;  
  
    if (fwd_url.startsWith("null"))  
    {  
      showError({ "error": "Could not connect to the FWD server!" });  
    }  
    else  
    {  
      ifr.src = fwd_url;  
..... passed up to this point correctly
```

Investigating reasons.

#115 - 10/25/2017 09:26 AM - Sergey Ivanovskiy

One of the reason is this parameter must be set TRUE.

```
<node class="container" name="webClient">  
  <node class="boolean" name="embedded">  
    <node-attribute name="value" value="TRUE"/>  
  </node>  
.....
```

The second reason is early initialization of network object. Working on the fix now. Planning to use proxy object instead of WebClientsManager.

#116 - 10/25/2017 10:12 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

One of the reason is this parameter must be set TRUE.

[...]

The second reason is early initialization of network object. Working on the fix now. Planning to use proxy object instead of WebClientManager.

embedded=true is set only for ehotel user; you are enabling it globally, and Web virtual desktop client will no longer work properly. Why do you need it changed there?

Also, you can make changes to the embedded/src/ Java sources as needed.

#117 - 10/25/2017 10:13 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

There are some issues with js embedded part, the spawned web client response looks correct but the process is not going father this login point and iframe doesn't load application.

[...]

Investigating reasons.

Please make sure the deploy/server/hotel-root.der is imported in your browser as a root CA. Trunk works with the current settings, so check for any JS errors...

#118 - 10/25/2017 10:17 AM - Constantin Asofiei

Sergey, I'm OK with changes in 3287a rev 11194. I assume all sync issues are solved?

Also, you mention to release 3287a - is there a reason not to wait to see if the POC embedded mode works properly?

#119 - 10/25/2017 12:02 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

One of the reason is this parameter must be set TRUE.

[...]

The second reason is early initialization of network object. Working on the fix now. Planning to use proxy object instead of WebClientManager.

embeded=true is set only for ehotel user; you are enabling it globally, and Web virtual desktop client will no longer work properly. Why do you need it changed there?

WebClientBuilder.addClientOptions adds new configuration options to command line and if an option is given by the directory settings, then it can be repeated in the command line with the different values. Can we define any configuration option by this pattern key=value? If it is true, then we should override values of these keys and write command line without repeated keys.

#120 - 10/25/2017 02:14 PM - Sergey Ivanovskiy

Yes, you are correct. New hotel-root.der should be generated and imported into the browser. It seems that this issue appears for me regularly.

#121 - 10/26/2017 01:43 PM - Sergey Ivanovskiy

In two runs of the main-regression task there were failed tests in the first run

```
gso156_session1, gso156_session2, gso_430, tc_job_002, tc_job_clock_001, tc_job_clock_002, tc_job_clock_003, tc_job_clock_004, tc_job_clock_005, tc_job_clock_006, tc_dc_slot_023
```

and there were failed in the second run

```
gso_190, tc_job_002, tc_job_clock_002
```

Thus, tc_job_002 is suspected. Can it be run as a standalone test? Can I merge to the trunc if the regression tests are succeeded?

#122 - 10/26/2017 01:44 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Thus, tc_job_002 is suspected.

This is the single test which still fails in FWD, so you can ignore it.

#123 - 10/26/2017 01:47 PM - Sergey Ivanovskiy

Yes, you are correct. I mean tc_job_clock_002.

#124 - 10/26/2017 01:54 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Yes, you are correct. I mean tc_job_clock_002.

I've checked the failed screen and this one too fails in an expected way (is time-of-day dependent), so you can ignore it, too.

#125 - 10/26/2017 01:55 PM - Sergey Ivanovskiy

OK. (It is recommended to run TC-JOB-CLOCK-002 after 16-00 of the server time. Thus I will run it later.) Planning start ctrl_c tests now.

#126 - 10/26/2017 08:14 PM - Sergey Ivanovskiy

ctrl_c tests are frozen in two runs. Running this test again.

#127 - 10/27/2017 12:39 AM - Sergey Ivanovskiy

In the last run gso_ctrlc_tests passed, but gso_ctrlc_3way_tests failed for:

1. ctrlc_11_session1
2. ctrlc_11_session3
3. ctrlc_11_session4
2. ctrlc_11_session3
3. ctrlc_11_session4

#128 - 10/27/2017 02:57 PM - Sergey Ivanovskiy

gso_ctrlc_3way_tests tests were passed in this 20171027_124426 run. There is only suspected TC-JOB-CLOCK-002.

#129 - 10/27/2017 02:59 PM - Sergey Ivanovskiy

Greg, Constantin, can this branch be merged into trunc if this test will be OK?

#130 - 10/27/2017 03:33 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Greg, Constantin, can this branch be merged into trunc if this test will be OK?

Is there anything else to work on 3287a? Is the locally spawned client for embedded mode work?

#131 - 10/27/2017 03:37 PM - Sergey Ivanovskiy

Yes, it should work without any additional setup. There are only required to change Hotel GUI embedded and customer's poc embedded to use this interface

```
/**
 * Provide a low-level API to spawn a web client.
 */
public interface WebClientLauncher
{
    /**
     * Spawn a new web client process, returning the redirect URL if successful.
     *
     * @param user
     *         The OS userid to use for the new process.
     * @param pw
     *         The OS password to authenticate the user.
     * @param gui
     *         This specifies the UI mode (true for GUI, false for
     *         ChUI).
     * @param options
     *         List of caller-specified options to be passed on the client command line or
     *         null if no values are specified.
     * @param referrer
     *         The referrer string to provide the web client. If the client gets disconnected
     *         it will be redirected back to this URL.
     * @param requestParameters
     *         Holds additional parameters if they are provided. It is expected that the first
     *         requestParameters0 holds the forwarded host of "X-Forwarded-Host" HTTP header,
     *         requestParameters1 holds the corresponding forwarded protocol that is provided
     *         by "X-Forwarded-Proto" value and requestParameters2 holds the client IP address.
     *
     * @return The redirect URL for the new web client on success. If a failure occurred, the
     *         returned string will be prefixed by {@link WebHandler#ERROR_PREFIX} and followed
     *         by descriptive error text.
     */
    public String spawn(String user,
                       String pw,
                       boolean gui,
                       String[] options,
                       String referrer,
                       String[] requestParameters);
}
```

How to change looks simple.

#132 - 10/27/2017 03:40 PM - Sergey Ivanovskiy

If there are no brokers, then web clients are created locally.

#133 - 10/27/2017 03:42 PM - Constantin Asofiei

OK, so if a broker is started on the client's machine, the server will spawn the FWD client for the embedded app via the broker, on the client's machine?

#134 - 10/27/2017 03:49 PM - Sergey Ivanovskiy

Yes, it is the first case from [#3287-72](#) with the dedicated mode. The broker user is a trusted FWD user used for trusted connection to P2J server by the embedded driver.

```
String[] options =
    {
        "client:cmd-line-option:param=USERID=FWD_TRUSTED_USER|PASSWORD=",
        "client:cmd-line-option:startup-procedure=STARTUP-PROCEDURE.p",
        "client:web:embedded=true"
    };
```

#135 - 10/27/2017 03:53 PM - Sergey Ivanovskiy

I meant this broker user in client.xml for the broker agent.

```
<remote>
  <retry count="10" />
  <retry seconds="10" />
  <spawner file="/home/sbi/opt/spawner/spawn" />
  <agent host="CLIENT_HOST_NAME_OR_IP_ADDRESS"/>
  <agent user="FWD_TRUSTED_USER"/>
  <agent dedicatedMode="true"/>
  <java args="-Xmx512m -Djava.awt.headless=true -XX:MaxPermSize=64m -Djava.library.path=/home/sbi/projects/3287a/build/lib/ -Dfile.encoding=UTF-8" />
  <java classpath="/home/sbi/projects/3287a/build/lib/p2j.jar" />
</remote>
```

#136 - 10/27/2017 04:05 PM - Constantin Asofiei

Ok, then 3287a can be released.

For Hotel GUI and POC: can you make the changes so that it doesn't affect the current way it is supposed to work? Maybe create different client.xml or deploy/client folders...

#137 - 10/27/2017 04:13 PM - Sergey Ivanovskiy

Yes, planning to create deploy/broker/ folder. It seems that I am very needed the tool that can update only one new created account. It helps to preserve all certificates and passwords.

But I haven't prepared this case yet. If the broker isn't planned to be used before 3240a will be merged, then we can postpone changes in the directory.xml for these two projects. Without broker setup the web clients will be spawned locally as it works now.

#138 - 10/27/2017 04:15 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Yes, planning to create deploy/broker/ folder. It seems that I am very needed the tool that can update only one new created account. It helps to preserve all certificates and passwords.

But I haven't prepared this case yet. If the broker isn't planned to be used before 3240a will be merged, then we can postpone changes in the directory.xml for these two projects. Without broker setup the web clients will be spawned locally as it works now.

OK, then delay updating the Hotel GUI and customer POC projects until 3240a is done.

#139 - 10/27/2017 05:35 PM - Constantin Asofiei

And please merge 3287a to trunk.

#140 - 10/27/2017 06:41 PM - Sergey Ivanovskiy

OK. Running majic_regression_testing_tc.xml test case for TC-JOB-CLOCK-002, according to logs it should be completed sooner.

#141 - 10/27/2017 07:22 PM - Sergey Ivanovskiy

The last result 20171027_153157 were not good but the target test TC-JOB-CLOCK-002 passed - OK. Planning to merge into the trunc if there are no objections.

#142 - 10/29/2017 04:18 AM - Sergey Ivanovskiy

3287a was merged into the trunc rev 11183 and archived.

#143 - 02/23/2018 04:52 PM - Greg Shah

We will be doing some testing of this feature using a large customer application. There is a finding in #3438-135 that is one item that needs to be checked.

#144 - 02/24/2018 02:11 AM - Sergey Ivanovskiy

Greg Shah wrote:

We will be doing some testing of this feature using a large customer application. There is a finding in #3438-135 that is one item that needs to be checked.

OK, I will think about it too, it seems that this issue is not related to web client design. Is there possible to make a link to this network resource N: in order to start the browser from the command file or link placed at the target N: resource? Are there differences between Swing client and the browser client if we consider that all these clients depend on some resource placed at the network resource N:?

#145 - 02/24/2018 12:16 PM - Greg Shah

Is there possible to make a link to this network resource N: in order to start the browser from the command file or link placed at the target network resource? Are there differences between Swing client and the browser client if we consider that all these clients depend on some resource placed at the network resource?

It is safe to assume that the client system will have the any application dependencies satisfied in order for the application to work properly.

I don't think we need to do anything pecial in that regard. The local web client should work pretty naturally. The issue is more about client in the cloud where we spawn a new client process when the user connects. Having network resource dependencies that must be satisfied is more tricky in that case.

#146 - 03/09/2020 03:16 PM - Sergey Ivanovskiy

- Related to Feature #2683: reverse proxy implementation added