Runtime Infrastructure - Bug #3852

improve runtime handling of helper application idiom

01/07/2019 02:34 AM - Sergey Ivanovskiy

Status:	New		**Start date:**	01/07/2019
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**case_num:**	
vendor_id:	GCD		**version:**	

Description

History

#1 - 01/07/2019 02:40 AM - Sergey Ivanovskiy

Updated by Eric Faulhaber about 1 month ago

4GL applications commonly use the idiom of

OS-COMMAND [NO-WAIT] START <helper application> [<parameters>].

to launch helper applications, such as Excel or a file viewer/editor, often passing a filename as a parameter.

We convert this to something like:

ProcessOps.launch(new String[] { "START", <helper application>, [<parameters>,] }, false, false);

This works ok if the FWD client is a Swing client running on the same platform as the original application, with the specified helper application available.

However, it does not work in the web client case, nor in the case where the client is running on a system where the original helper application does not exist (e.g., on a different OS). For this case, we need to improve the intelligence of our handling of this idiom by enabling content to be delivered to the client system, where the browser's mapping of helper applications by MIME type can take over.

Since not every invocation of ProcessOps.launch can be assumed to work this new way, the directory should contain a mapping of original helper applications (e.g., excel.exe) to MIME types. Those which are mapped in the directory should handle the content passed as a parameter to the original helper application (e.g., a CSV or XLS filename) by delivering it to the client via openMimeResource. Helper applications for which there is no such mapping in the directory should behave the same way as today. The lookup should normalize any helper application names passed to ProcessOps.launch, to account for qualified paths, mixed case as appropriate to the original application's platform, etc.

Need some feedback: for cases where the Swing client is running on the same platform as the original application, do we go to this directory-mapped approach, or do we handle the process launch as we do today? Or do we attempt the process launch and failover to the directory-mapped approach? Perhaps this last option is also a directory setting?

Updated by Greg Shah about 1 month ago

It does not matter what type of client (Swing or web) is being used. Web clients can in fact use OS-COMMAND for non-interactive things. You rightly mention that Swing does have the ability to use all possible cases (including programs that are interactive). But we don't want to depend upon the client type for this feature.

By default we only implement this if the FWD client platform OS is different from the 4GL legacy platform OS. This is the same calculation we already make for other reasons.

I do think we should add a flag to force this mapping even for platforms that match (which are the same). This would be an override of the default behavior.

https://blogs.msdn.microsoft.com/vsofficedeveloper/2008/05/08/office-2007-file-format-mime-types-for-http-content-streaming-2/
lists extension and mime type pairs for Microsoft Office file formats.

Added these new settings to clientConfig container

```
        <node class="container" name="clientConfig">
          <node class="string" name="spawner">
```

```
          <node-attribute name="value" value="/opt/spawner/spawn"/>
        </node>
........................................................................
        <node class="boolean" name="resolveFilePaths">
          <node-attribute name="value" value="FALSE"/>
        </node>
        <node class="container" name="helper-applications">
          <node class="entry" name="entry1">
            <node-attribute name="key" value="excel.exe"/>
            <node-attribute name="value" value="application/vnd.openxmlformats-officedocument.spreadsheetml.
sheet"/>
          </node>
        </node>
      </node>
```

#2 - 01/07/2019 02:45 AM - Sergey Ivanovskiy

Found that the current version doesn't support the START command options. From https://ss64.com/nt/start.html

```
Syntax
      START "title" [/D path] [options] "command" [parameters]
```

There are different OS commands to start a target application.