User Interface - Feature #2368

allow widget output to stream in GUI mode

08/08/2014 03:11 PM - Constantin Asofiei

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	80%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

Related issues:

Related to User Interface - Feature #2373: implemented output to redirected, ...	**New**	
Related to User Interface - Feature #2399: Implement output redirection in GU...	**Rejected**	**09/27/2014**

History

#1 - 08/08/2014 03:12 PM - Constantin Asofiei

- Parent task set to #2252

#2 - 08/08/2014 03:13 PM - Constantin Asofiei

In GUI mode, writing widget to streams requires access to a ChUI driver. See #2252-127.

#3 - 08/08/2014 03:31 PM - Constantin Asofiei

- Target version set to Milestone 12

#4 - 03/22/2016 12:16 PM - Greg Shah

- Target version deleted (Milestone 12)

- Parent task deleted (#2252)

#5 - 03/22/2016 12:17 PM - Greg Shah

Is this really the same task as #2373?

#6 - 03/22/2016 12:23 PM - Constantin Asofiei

Greg Shah wrote:

> Is this really the same task as #2373?

No, it is not. #2373 is a sub-feature of using streams which target the terminal. This task is to allow "driver switching" when output is to stream, so ChUI widgets will be used.

#7 - 04/27/2017 02:19 PM - Greg Shah

This work should include support for the frame option STREAM-IO (runtime is needed).

#8 - 08/29/2017 11:34 AM - Greg Shah

Interesting finding about stream-io, it overrides all view-as clauses and forced them to a text widget type. This was found in #3330-120. I initially thought it was related to the implicit activation of use-text, but that is incorrect as this testcase shows:

```
def temp-table tt
   field num as int view-as radio-set radio-buttons "One", 1, "Two", 2
   field bool as logical view-as toggle-box.

def var rs as int view-as radio-set radio-buttons "One", 1, "Two", 2.
def var tb as log view-as toggle-box.
def var tb2 as log.

create tt.
tt.num = 2.
tt.bool = ?.

/* interactive case, @ base field cannot be used with non-fillin or non-text widegts */
update rs tb tt.num tt.bool tb2 view-as toggle-box with frame interactive-fr.

/* stream-io, overrides all view-as phrases (including schema, var def and explicit/in-line format phrase form
s) */
form rs tb tt.num tt.bool tb2 view-as toggle-box with frame io-fr stream-io.
def var l2 as log.
display 14 @ rs
        l2 @ tb
        29 @ tt.num
        no @ tt.bool
        yes @ tb2
        with frame io-fr.

/* use-text (this doesn't work, the explicit view-as in the var def causes "Invalid use of @ with non-field wi
dget. (3467)":
   form rs tb with frame ut-fr use-text.
   display 14 @ rs
           l2 @ tb with frame ut-fr.
*/

/* use-text (this doesn't work, the schema view-as causes "Invalid use of @ with non-field widget. (3467)":
   form tt.num tt.bool with frame ut-fr use-text.
   display 29 @ tt.num
           no @ tt.bool
           with frame ut-fr.
*/

/* use-text (this doesn't work, the explicit view-as causes "Invalid use of @ with non-field widget. (3467)":
   form tb2 view-as toggle-box with frame ut-fr use-text.
   display yes @ tb2 with frame ut-fr.
*/
```

All widget types in the frame must be overridden at conversion time.

#10 - 09/27/2017 07:37 PM - Constantin Asofiei

Greg Shah wrote:

> This work should include support for the frame option STREAM-IO (runtime is needed).

Isn't STREAM-IO actually more of a conversion-only solution? I don't see any runtime dependency, just force all widgets in this frame to be TEXT at conversion time (and drop all non-text options, otherwise the frame will not be possible to be instantiated if i.e. we don't drop the code which adds items to a comb-box).

#11 - 09/28/2017 08:22 AM - Greg Shah

Possibly. But I wonder if there are layout differences that can only be seen at runtime. For example, I think the FILL-IN widgets may take less space.

That seems like it may have runtime implications. Especially if the frame can be used for interactive purposes in GUI. I haven't tested it.

#12 - 09/28/2017 12:55 PM - Constantin Asofiei

Greg Shah wrote:

> Possibly. But I wonder if there are layout differences that can only be seen at runtime. For example, I think the FILL-IN widgets may take less space.

All widget types are reported as TEXT; the unknown is there are differences in how the default size computed; if an explicit SIZE phrase is present, then that is not discarded.

> That seems like it may have runtime implications. Especially if the frame can be used for interactive purposes in GUI. I haven't tested it.

STREAM-IO makes all widgets TEXT, and 4GL doesn't allow an UPDATE for it, but it can be DISPLAY'ed. There are differences:

1. the font looks like is DEFAULT-FIXED-FONT (or maybe SYSTEM?), not sure, is different than the one used by explicit VIEW-AS TEXT widgets
2. bgcolor/fgcolor is not working
3. might be alignment issues
4. default format is "x(8)"

This proves that STREAM-IO must be marked at the frame, so I'll do it. But the widget types will be forced at conversion time.

A question: can we defer the GUI behaviour for STREAM-IO, if is not in use (and all usage is to stream)?

#13 - 09/28/2017 12:58 PM - Greg Shah

> A question: can we defer the GUI behaviour for STREAM-IO, if is not in use (and all usage is to stream)?

Yes, it can be deferred if it is not in use. I have not checked this... how do you propose to figure it out?

#14 - 09/28/2017 01:23 PM - Constantin Asofiei

Greg Shah wrote:

> > A question: can we defer the GUI behaviour for STREAM-IO, if is not in use (and all usage is to stream)?
>
> Yes, it can be deferred if it is not in use. I have not checked this... how do you propose to figure it out?

Do you know how many references of STREAM-IO are in the project? We could review them manually if there are not that many.

Also, related to bgcolor/fgcolor: I think we might be entering the realm of #2373, as a BGCOLOR set at the widget directly and a BGCOLOR attribute can both be set to different values... and work simultaneously!

At conversion time, this was solved with #3330-98 (forgot about it), but we need to keep the SIZE phrase and TOOLTIP from VIEW-AS.

For now, I will fix the SIZE/TOOLTIP conversion issue and focus on the runtime (GUI-ChUI switch...).

#15 - 09/28/2017 01:50 PM - Greg Shah

> Do you know how many references of STREAM-IO are in the project?

5,514 :(

#16 - 09/28/2017 02:19 PM - Constantin Asofiei

A very high-level description of what needs to be done:

1. GUI-ChUI driver switching; this is tricky, as we need to have OutputManager, Driver, etc for both GUI and ChUI. Each driver will have its own widget registry, with instances of the widget for GUI/ChUI. We can implement a 'lazy' approach to populate the ChUI/redirected registry, so we don't double every widget instance. OTOH, I don't want to complicate the code for ChUI clients... instead, I'll look how this can be done directly in the GUI-related classes.
2. widget state in base widget classes must exist in a common place - I think best place is the widget config, with the fields marked as transient, so they will be kept only on client-side. This way, both GUI and ChUI versions will be able to work with the same state.

#17 - 10/17/2017 07:01 PM - Hynek Cihlar

Constantin, what is the status of the issue? I am asking to figure out whether I can use the branch for an early testing of #1795 (1795a).

#18 - 10/18/2017 07:39 AM - Constantin Asofiei

Hynek Cihlar wrote:

> Constantin, what is the status of the issue? I am asking to figure out whether I can use the branch for an early testing of #1795 (1795a).

I want to make PUT, MESSAGE and EXPORT work, hope to finish today. For FRAME streaming, is a lot more complex.

#19 - 10/18/2017 08:31 AM - Constantin Asofiei

Hynek, see 2368a revision 11180 - you can use PUT, MESSAGE and EXPORT statements to test your printing support. Normal streaming works, both terminal redirection or explicit STREAM.

#20 - 10/18/2017 08:32 AM - Greg Shah

Does it make sense to get this branch tested and merged to trunk as is (and then do the rest of the work in another branch)?

#21 - 10/18/2017 08:33 AM - Constantin Asofiei

Greg Shah wrote:

> Does it make sense to get this branch tested and merged to trunk as is (and then do the rest of the work in another branch)?

Yes, it does - please review. I'll put it into testing, it shouldn't have impact on ChUI mode.

#22 - 10/18/2017 09:29 AM - Greg Shah

Code Review Task branch 2368a Revision 11180

I'm generally fine with the overall approach and changes.

The one concern I have is that the OutputManager now has knowledge of and references to one of its child classes. I would prefer an alternate approach:

- Create an interface that supports exactly the features needed. Have an abstract method that returns an instance that implements that interface.
- Extend the OutputManager with the methods needed and let the GuiOutputManager internally delegate the calls to a hidden instance of ChuiOutputManager.

I don't think you need to change this in the current branch, but I would prefer it to be improved in the next branch.

If it passes testing you can merge to trunk.

#23 - 10/19/2017 01:45 PM - Constantin Asofiei

Revision 11182 of 2368a contains a #3281 fix and javadoc fixes.

#24 - 10/19/2017 01:55 PM - Constantin Asofiei

Something interesting about the 4GL runtime of STREAM-IO and why the GUI apps require this frame option. For this code:

```
def var i as int.
def var ch as char.
i = 12345.
ch = "Abcdef".

form i with frame f1.
output to a.txt.
display i with frame f1.
output close.

form i with frame f2 side-labels.
output to b.txt.
display i with frame f2.
output close.

form i with frame f3 stream-io.
output to c.txt.
display i with frame f3.
output close.

form i with frame f4 side-labels stream-io.
output to d.txt.
display i with frame f4.
output close.
```

there are these results:

1. a.txt

```
---------------i

    12,345
```

2. b.txt

```
i12,345
```

3. c.txt

```
         i
----------
    12,345
```

4. d.txt

```
i: 12,345
```

Notice how the label in first two cases, without STREAM-IO, looks messy... and this is happening only in GUI, ChUI works OK on both Linux and Windows.

#25 - 10/19/2017 02:11 PM - Constantin Asofiei

Some good news: I finally have some code which allows FWD to stream frames in GUI (both named and unnamed streams). I need to refactor it a little (as it's kind of messy right now), clean it up and put into testing.

#26 - 10/19/2017 02:54 PM - Greg Shah

> Something interesting about the 4GL runtime of STREAM-IO and why the GUI apps require this frame option.

Interesting findings. It certainly is easy to see why a developer would use it. The baffling thing: why does Progress require it when generating redirected output? Surely they know when they are outputting to redirected output versus when they render a WIN32 GUI? The output in the non-STREAM-IO case seems useless.

> Some good news: I finally have some code which allows FWD to stream frames in GUI (both named and unnamed streams). I need to refactor it a little (as it's kind of messy right now), clean it up and put into testing.

That is great news!

#27 - 10/20/2017 07:40 AM - Constantin Asofiei

2368a rev 11182 passed testing, it can be released.

#28 - 10/20/2017 07:44 AM - Greg Shah

Please merge to trunk.

#29 - 10/20/2017 08:04 AM - Constantin Asofiei

- % Done changed from 0 to 60

Greg Shah wrote:

> Please merge to trunk.

Merged to trunk rev 11180 and archived.

#30 - 10/20/2017 11:43 AM - Constantin Asofiei

- % Done changed from 60 to 80

- Status changed from New to WIP

Greg, please review 2368b rev 11181. I need to a little more testing, but the major changes I think are OK.

#31 - 10/20/2017 02:24 PM - Greg Shah

Code Review Task Branch 2368b Revision 11181

Wow! I'm pretty impressed with the solution. It is less code and much cleaner than I expected. Well done!

1. BorderedPanel needs a history entry.

2. Is it a problem to have references to WidgetId._DEFAULT_WINDOW_ID in ChuiOutputManager in the new design? In other words, can't this window be a GUI window now?

#32 - 10/20/2017 02:24 PM - Greg Shah

What is left to do in this task (besides testing)?

#33 - 10/20/2017 02:31 PM - Constantin Asofiei

Greg Shah wrote:

> 1. BorderedPanel needs a history entry.

Will fix.

> 2. Is it a problem to have references to WidgetId._DEFAULT_WINDOW_ID in ChuiOutputManager in the new design? In other words, can't this

> window be a GUI window now?

Yes, it looks confusing; ChUI doesn't require window IDs, as selectWindow is a no-op in ChuiOutputManager. So this WidgetId._DEFAULT_WINDOW_ID will have no effect. I'll add some comments.

#34 - 10/20/2017 02:34 PM - Constantin Asofiei

Greg Shah wrote:

> What is left to do in this task (besides testing)?

There is some runtime behaviour in 4GL, when comparing a frame which is redirected and when the same frame is output to UI. At least visibility, realisation, implicit location, down body (this is not created at all), etc for a streamed frame are separate for the same frame's state, when output to UI. I think this is for both ChUI and GUI. We should be good as long as a frame is not used in mixed mode (and from all code I've seen, frames are usually single-purpose, a streamed frame will not output to UI). But I'll do some tests and see how FWD behaves and how easy is to fix it.

#35 - 10/22/2017 04:05 AM - Constantin Asofiei

2368b passed runtime testing and was merged to trunk rev 11181, archived.

#36 - 10/23/2017 11:12 AM - Hynek Cihlar

Constantin, with the latest trunk the following code sample abends.

```
message "Show def window".
output to test.txt.
display "Test display".
wait-for close of this-procedure.
```

with

```
Caused by: java.lang.ClassCastException: com.goldencode.p2j.ui.client.StreamableManager$2 cannot be cast to co
m.goldencode.p2j.ui.client.gui.driver.GuiDriver
     at com.goldencode.p2j.ui.client.WindowManager.processWindowEvent(WindowManager.java:1695)
     at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.processEvent(WindowGuiImpl.java:1353)
     at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent(ThinClient.java:16793)
     at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker(ThinClient.java:16240)
     at com.goldencode.p2j.ui.chui.ThinClient.pop(ThinClient.java:15248)
     at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15231)
     at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15155)
     at com.goldencode.p2j.ui.chui.ThinClient.applyWorker(ThinClient.java:14916)
     at com.goldencode.p2j.ui.chui.ThinClient.waitForEvent(ThinClient.java:14185)
     at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11936)
     at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:11515)
     at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11463)
     at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:11417)
```

As seen in the exception call stack, the exception is caused during the wait-for call. When that is removed, the program executes fine. Note that the abends happen when the system event queue handles a message, so you have to for example move the mouse cursor into the window.

#37 - 10/23/2017 11:24 AM - Hynek Cihlar

Another one.

```
message "Show def window".
output to "test.txt".
def var i as int.
repeat i = 0 to 100:
  display "Test" i.
end.
wait-for close of this-procedure.
```

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.gui.driver.GuiPrimitivesImpl.screenWidth(GuiPrimitivesImpl.java:172)
    at com.goldencode.p2j.ui.client.OutputManager.screenWidth(OutputManager.java:829)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.screenWidth(AbstractWidget.java:2288)
    at com.goldencode.p2j.ui.client.ZeroColumnLayout.maxWidth(ZeroColumnLayout.java:1688)
    at com.goldencode.p2j.ui.client.ZeroColumnLayout.minimumSize(ZeroColumnLayout.java:554)
    at com.goldencode.p2j.ui.client.Frame.doLayoutWorker(Frame.java:1762)
    at com.goldencode.p2j.ui.client.Frame.doLayout(Frame.java:1595)
    at com.goldencode.p2j.ui.client.Frame.postprocess(Frame.java:3954)
    at com.goldencode.p2j.ui.client.WidgetRegistry.pushDefinition(WidgetRegistry.java:667)
    at com.goldencode.p2j.ui.chui.ThinClient.lambda$pushOneDef$15(ThinClient.java:8460)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:15211)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:15155)
    at com.goldencode.p2j.ui.chui.ThinClient.pushOneDef(ThinClient.java:8451)
    at com.goldencode.p2j.ui.chui.ThinClient.pushScreenDefinition(ThinClient.java:8399)
```

#38 - 10/23/2017 11:24 AM - Constantin Asofiei

Hynek, the key difference is that you have the unnamed stream redirected when wait-for starts. I'm looking into it.

#39 - 10/23/2017 04:11 PM - Constantin Asofiei

2368c rev 11182 should fix these two cases.

#40 - 10/24/2017 09:10 AM - Greg Shah

Code Review Task Branch 2368c Revision 11182

I'm fine with the changes. Go ahead and put them through runtime testing and get them into trunk.

#41 - 10/24/2017 09:24 AM - Greg Shah

Alternatively, these can be merged into 1795b since that will be going through testing soon and it is dependent on your changes.

#42 - 10/24/2017 11:48 AM - Constantin Asofiei

Greg Shah wrote:

> Alternatively, these can be merged into 1795b since that will be going through testing soon and it is dependent on your changes.

Hynek, you can merge 2368c into 1795b if you want.

#43 - 10/24/2017 12:45 PM - Hynek Cihlar

Constantin Asofiei wrote:

> Greg Shah wrote:
>
> > Alternatively, these can be merged into 1795b since that will be going through testing soon and it is dependent on your changes.
>
> Hynek, you can merge 2368c into 1795b if you want.

Ok, will do.

#44 - 10/24/2017 03:21 PM - Hynek Cihlar

2368c merged in 1795b and dead-archived.

#45 - 10/28/2017 10:14 AM - Hynek Cihlar

1795b was merged to trunk as revision 11184.

#46 - 11/01/2017 12:50 PM - Greg Shah

Constantin: Did you ever check to see if STREAM-IO frames have different behavior when used interactively?

If I recall that was the last item open in this task, correct?

#47 - 11/01/2017 12:56 PM - Constantin Asofiei

Greg Shah wrote:

> Constantin: Did you ever check to see if STREAM-IO frames have different behavior when used interactively?

Some findings are in #2368-12

> If I recall that was the last item open in this task, correct?

There is also mixed usage of a streamed frame - i.e. if same frame is used both with streams and in GUI, I'm not yet sure what state is shared between these two modes; and here I mean coordinates, fonts and other attributes.

#48 - 11/01/2017 01:02 PM - Greg Shah

> We should be good as long as a frame is not used in mixed mode (and from all code I've seen, frames are usually single-purpose, a streamed frame will not output to UI). But I'll do some tests and see how FWD behaves and how easy is to fix it.

This mixed mode (from #2368-34) is the part I was remembering. Is it the only thing left?

Perhaps we can defer this, but we may want to at least detect when it occurs and write a log entry.

#49 - 11/01/2017 01:04 PM - Constantin Asofiei

Greg Shah wrote:

> > We should be good as long as a frame is not used in mixed mode (and from all code I've seen, frames are usually single-purpose, a streamed frame will not output to UI). But I'll do some tests and see how FWD behaves and how easy is to fix it.
>
> This mixed mode (from #2368-34) is the part I was remembering. Is it the only thing left?

Related to streamed frames, yes. To be clear, issues mentioned in #2368-12 are not fixed.

> Perhaps we can defer this, but we may want to at least detect when it occurs and write a log entry.

OK, I'll think of a way to log this.

#50 - 11/01/2017 01:31 PM - Greg Shah

> To be clear, issues mentioned in #2368-12 are not fixed.

OK, then log this case too (VIEW of a STREAM-IO frame).

We will defer work on both since it seems unlikely to be an important production use case.

#51 - 12/23/2017 09:59 AM - Hynek Cihlar

There are numerous places where ClassCastException is thrown when in GUI redirected mode and an ALERT-BOX or an error message box is attempted to be shown by the legacy app logic.

The problem is that OutputManager.getDriver() will return the driver implementing the redirection which the non-streamable widgets (and other non-widget places that may ignore the redirection) cast to GuiDriver. The cast obviously throws the cast exception.

This is a system problem. There are different cases where the call site of OM.getDriver() legitimately assumes a GuiDriver instance, whether this is context dependent or not. Like an ALERT-BOX which must be displayed even when redirected, or when a window or menu is redrawn during redirected mode, etc.

I am not sure what the correct solution is. OM.getDriver() could be altered from outside with a flag that would make the method return the non-redirection driver:

```
void AertBoxGuiImpl.show()
{
   OutputManager.setIgnoreRedirection(true);
   GuiDriver gd = (GuiDriver) OutputManager.getDriver();
   ... do the other stuff ...
   OutputManager.setIgnoreRedirection(false);
}
```

but this would also negatively influence the logic that must get the redirection driver. Ultimately it would be good to come up with a mechanism to avoid to have to revisit all the OutputManager.getDriver() calls.

#52 - 12/23/2017 04:10 PM - Constantin Asofiei

Hynek, can you give me a simple standalone test which shows a few of the failures you describe in the previous note?

#53 - 12/23/2017 05:33 PM - Hynek Cihlar

Constantin Asofiei wrote:

> Hynek, can you give me a simple standalone test which shows a few of the failures you describe in the previous note?

```
output to bogus.txt.
message "This will abend" view-as alert-box.
```

The named stream redirection seems to work OK since the streamable OM in the GUI OM is redirected only during the respective 4GL statements and the code that needs the GUI driver isn't executed.

#54 - 12/27/2017 04:03 PM - Hynek Cihlar

- File ThinClient.java.diff added

Hynek Cihlar wrote:

> The named stream redirection seems to work OK since the streamable OM in the GUI OM is redirected only during the respective 4GL statements and the code that needs the GUI driver isn't executed.

The solution is to force interactive mode for the message box initialization and its event loop. See the attached diff.

#55 - 12/28/2017 03:20 AM - Greg Shah

I'm OK with the approach, it seems correct.

#56 - 03/04/2018 08:59 AM - Greg Shah

STREAM-IO has gap marking as stubs for runtime. As far as I know, we fully support it except for the interactive usage of a STREAM-IO frame. Is that correct?

#57 - 03/04/2018 09:09 AM - Constantin Asofiei

Greg Shah wrote:

> STREAM-IO has gap marking as stubs for runtime. As far as I know, we fully support it except for the interactive usage of a STREAM-IO frame. Is that correct?

The runtime stuff in #2368-12 and #2368-24 is not supported.

#58 - 03/05/2018 05:18 PM - Greg Shah

> > STREAM-IO has gap marking as stubs for runtime. As far as I know, we fully support it except for the interactive usage of a STREAM-IO frame. Is that correct?
>
> The runtime stuff in #2368-12 and #2368-24 is not supported.

Do we implement any of the runtime features of STREAM-IO or is it really just stubs?

#59 - 03/06/2018 01:57 AM - Constantin Asofiei

Greg Shah wrote:

> > > STREAM-IO has gap marking as stubs for runtime. As far as I know, we fully support it except for the interactive usage of a STREAM-IO frame. Is that correct?
> >
> > The runtime stuff in #2368-12 and #2368-24 is not supported.
>
> Do we implement any of the runtime features of STREAM-IO or is it really just stubs?

The STREAM-IO behaviour is managed by the conversion rules (i.e. all widgets are TEXT for these kind of frames, we assume USE-TEXT for everything). Also, we are setting this config option at the frame's runtime correctly. Only the weird findings from the notes I mentioned are missing the runtime support.

So no, there are no stubs in place.

#60 - 03/06/2018 08:32 AM - Greg Shah

Is it correct to say this is "partial" runtime support?

#61 - 03/06/2018 08:38 AM - Constantin Asofiei

Greg Shah wrote:

> Is it correct to say this is "partial" runtime support?

Yes, mark it 'runtime partial' and maybe a comment to point to this task/note. The problem with STREAM-IO is better said that, without this, streaming widgets in GUI works 'weird' (see #2368-24). So, if we encounter streaming issues in GUI, most likely the developer missed the STREAM-IO option at the frame. As I doubt someone would rely on the normal behavior...

Files

ThinClient.java.diff	2.52 KB	12/27/2017	Hynek Cihlar