Omnis Studio v11.1

NOW Get Ahead

ODC 2024, Peter Kelly

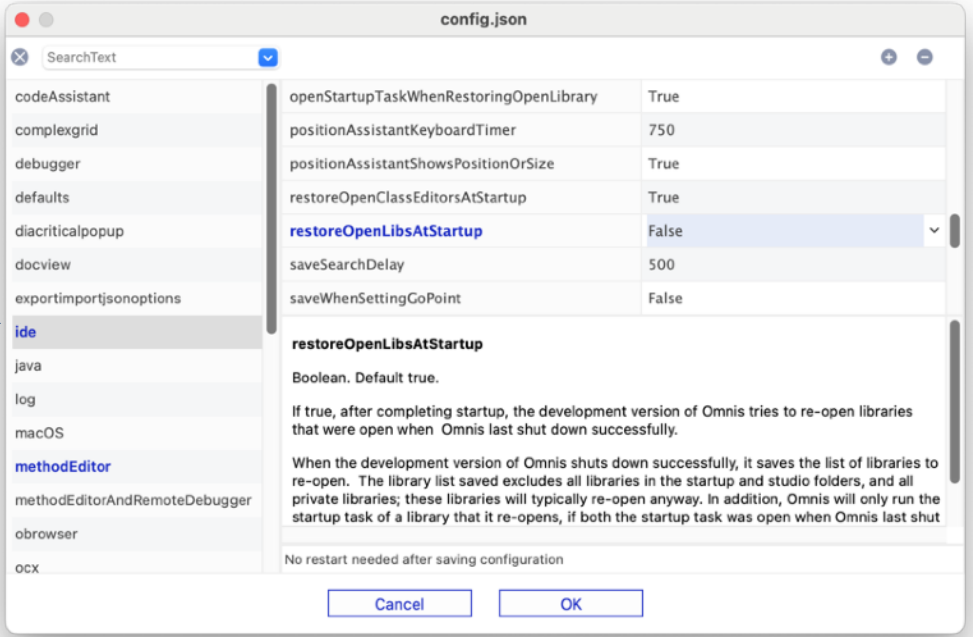
Omnis Config

Userconfig.json

- The new userconfig.json file
  - *Added to the Omnis tree and used to store config changes or additions.*
  - *The core looks in userconfig.json first, then config.json*
  - *Changes made using the config editor are saved to userconfig.json*
  - *Userconfig.json can be preserved between Omnis versions.*

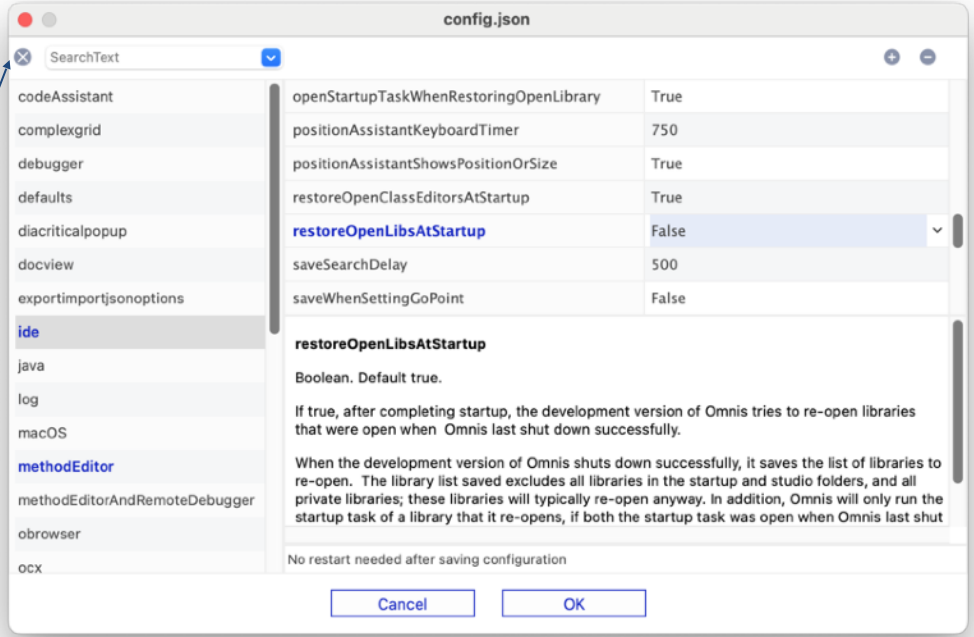
Userconfig.json

- Updated config editor
  - *The editor will **visually** show what groups and group items have been changed.*



Userconfig.json

- Updated config editor
  - *The editor will **visually** show what groups and group items have been changed.*
  - ***Search** support added to the config editor*
  - *It is recommended you do not alter config.json, ideally use the editor to make changes.*



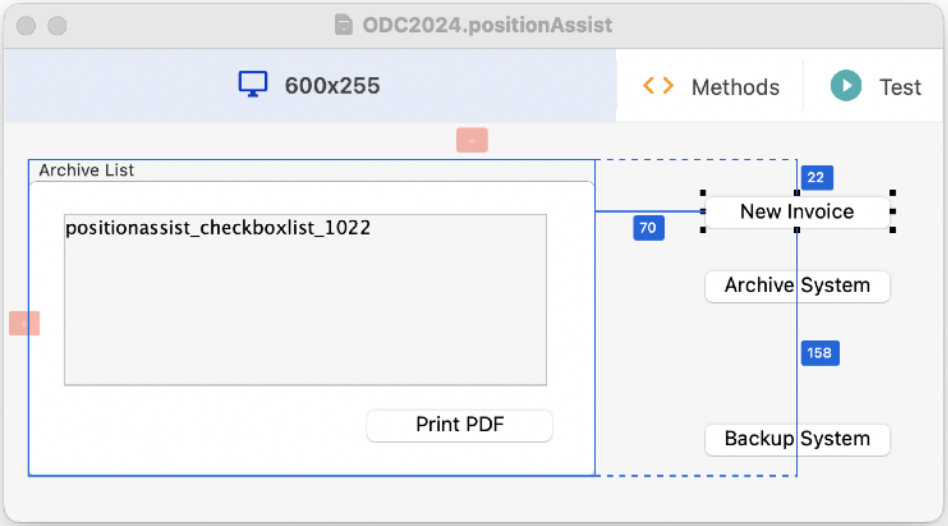
Config Additional Changes

- Additional changes
  - *$**getconfigjson**() or $**setconfigjson**() now take an optional parameter (pass as kTrue) to access config.json as by default the functions operate on userconfig.json*
  - *Some Omnis preferences previously stored in **omnis.cfg** (binary) have been moved to config.json (see new prefs section of the config editor)*
  - *A new **positions.cfg** file exists in the tree and used to store window positions.*
    This file can be preserved between Omnis versions.

Position Assistant

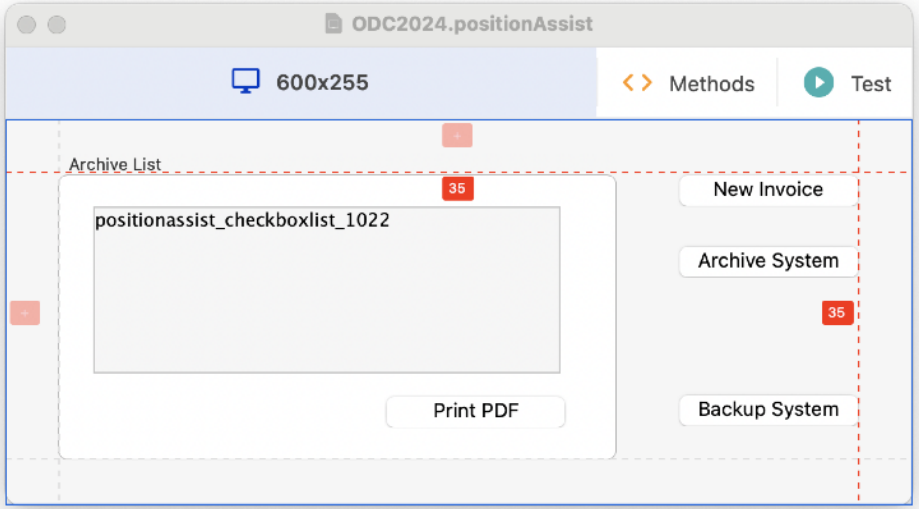
Position Assistant

- Distance Measurements
  - *Select an object or objects in design mode*
  - *With **command(Ctrl) key held down** mouse over other objects (inspecting)*
  - *Supported in window class and Remote forms*



Position Assistant

- Design Guides
  - *$designguides have been added to window & remote form classes*
  - *Can be set in a superclass and inherited*
  - *Objects snap to guides when moved/sized.*
  - *Guides can be added,removed or sized in design mode*
  - *Only shown when moving, sizing or inspecting (command/ctrl key)*



JS Markdown

omnis

JS Markdown

- Markdown is a new xcomp to render a basic set of markdown elements
  - *Markdown is supported on remote forms as a new control*
  - *The **Basic Syntax** set of markdown elements are supported.*
    See MarkDownGuide.org - cheat sheet
  - *Local and external links supported.*
  - *Tasklists will be processed and rendered but cannot be toggled*

JS Markdown

- Visual Appearance
  - *Various properties control visual rendering of some markdown elements*
  - *Markdown code blocks are automatically added using 3 backticks at the start and end of a text block*
  - *There is a great markdown sample in the HUB*

JS Markdown

- *Markdown is also supported on the desktop client via oBrowser*
  - *Implemented as an HTML control*
  - *Set $htmlcontrolsusehttp to kTrue*
  - *Select Markdown from $htmlcontroloptions*
  - *Same render rules from JS mode apply for Desktopmode*

Java Worker

omnis

JavaWorker

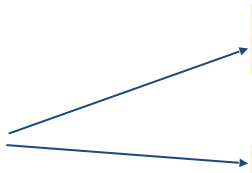
- New java worker available in the Omnis tree
    - *Java is not installed by default. Java version 17 is required*
    - *JavaWorker is similar to other workers.*
    - *$init() differs as this allows optional control over the startup of the JVM*

      eg. CLASSPATH, Paths to the JVM, JVM Options
    - *The Javaworker core implementation lives in the clientserver folder*
    - *Java modules you want to interact with live in subfolders in the javaworker tree root folder*

JavaWorker

- An example in its simplest form
  - *Create a java module and extend from our Omnis **OModule** class*

    This class handles the interface between the Omnis worker and Java



```
package net.omnis.OmnisTest;
import net.omnis.OmnisCalls.*;
import java.util.Map;
import java.util.HashMap;
public class Test extends OModule
{
        public Response test(Map<String, Object> pParams)
        {
                Map<String, Object> data = new HashMap<>();
                data.put("my_return, "my_return_value");
                return new SendResponse(data);
        }
}
```

JavaWorker

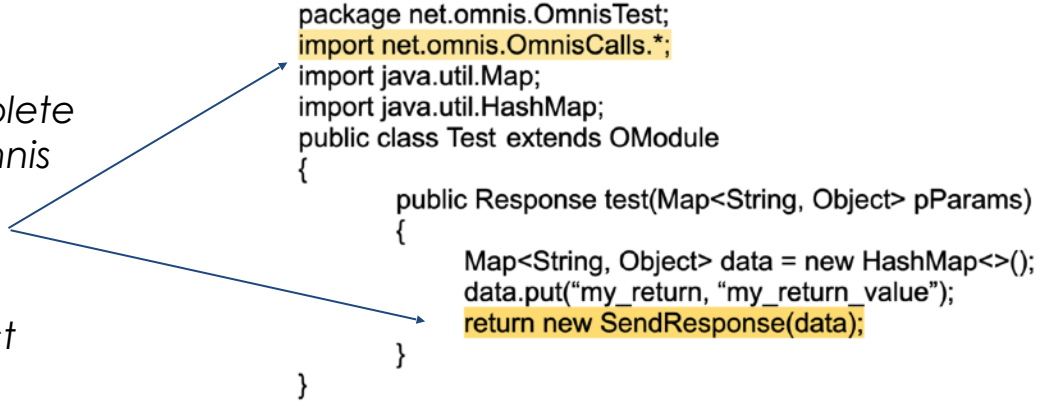
- Create your module
  - *Write your function content*
  - *Process parameters from Omnis*
  - *Send a response to Omnis*

```
package net.omnis.OmnisTest;
import net.omnis.OmnisCalls.*;
import java.util.Map;
import java.util.HashMap;
public class Test extends OModule
{
        public Response test(Map<String, Object> pParams)
        {
            Map<String, Object> data = new HashMap<>();
            data.put("my_return, "my_return_value");
            return new SendResponse(data);
        }
}
```

JavaWorker

- Responding to Omnis
  - *When the java method is complete you should return a state to Omnis*

    SendResponse()
    Will invoke ***$methodreturn()***
    in your Omnis javaworker object



```
package net.omnis.OmnisTest;
import net.omnis.OmnisCalls.*;
import java.util.Map;
import java.util.HashMap;
public class Test extends OModule
{
        public Response test(Map<String, Object> pParams)
        {
            Map<String, Object> data = new HashMap<>();
            data.put("my_return, "my_return_value");
            return new SendResponse(data);
        }
}
```

  - *If you wish to return an error, use* ***SendError()*** *to invoke* ***$methoderror()*** *in the worker.*

  - *Once compiled, add your module to the Omnis tree*
    See the online help for more information about tools and building java modules.

JavaWorker

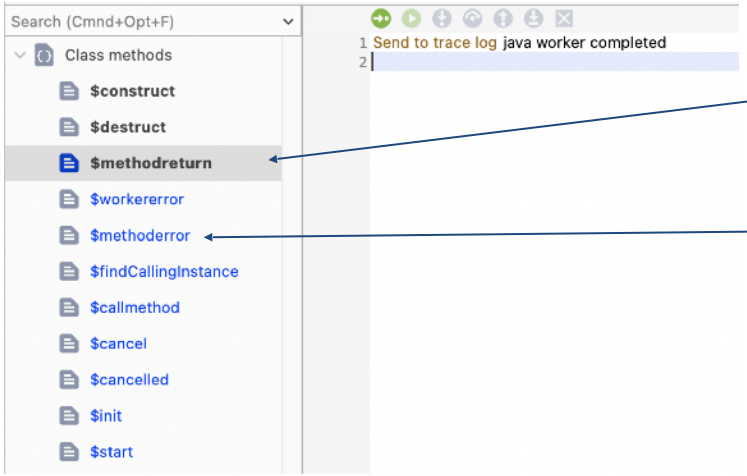
- Working within Omnis
  - *Create an object reference*
  - *Create instance with $newref*
  - *Call $init(), $start()*
  - *$callmethod into your java module*

```
#
# iJavaWorker - Objectref
#
Calculate iJavaWorker as $objects.oJavaWorker.$newref()
#
Do iJavaWorker.$init()
Do iJavaWorker.$start()
Do iJavaWorker.$callmethod("net.omnis.OmnisTest.Test","test") Returns #F
```

```
package net.omnis.OmnisTest;
import net.omnis.OmnisCalls.*;
import java.util.Map;
import java.util.HashMap;
public class Test extends OModule
{
        public Response test(Map<String, Object> pParams)
        {
                Map<String, Object> data = new HashMap<>();
                data.put("my_return, "my_return_value");
                return new SendResponse(data);
        }
}
```

JavaWorker

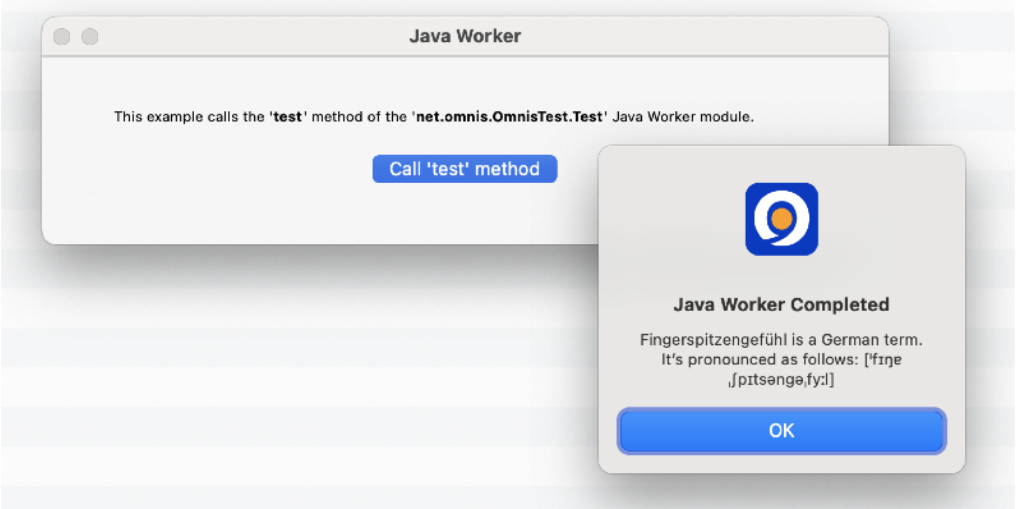
- Handling a response
  - *In your Javaworker*
  - *Override $**methodreturn***
  - *Override **$methoderror***



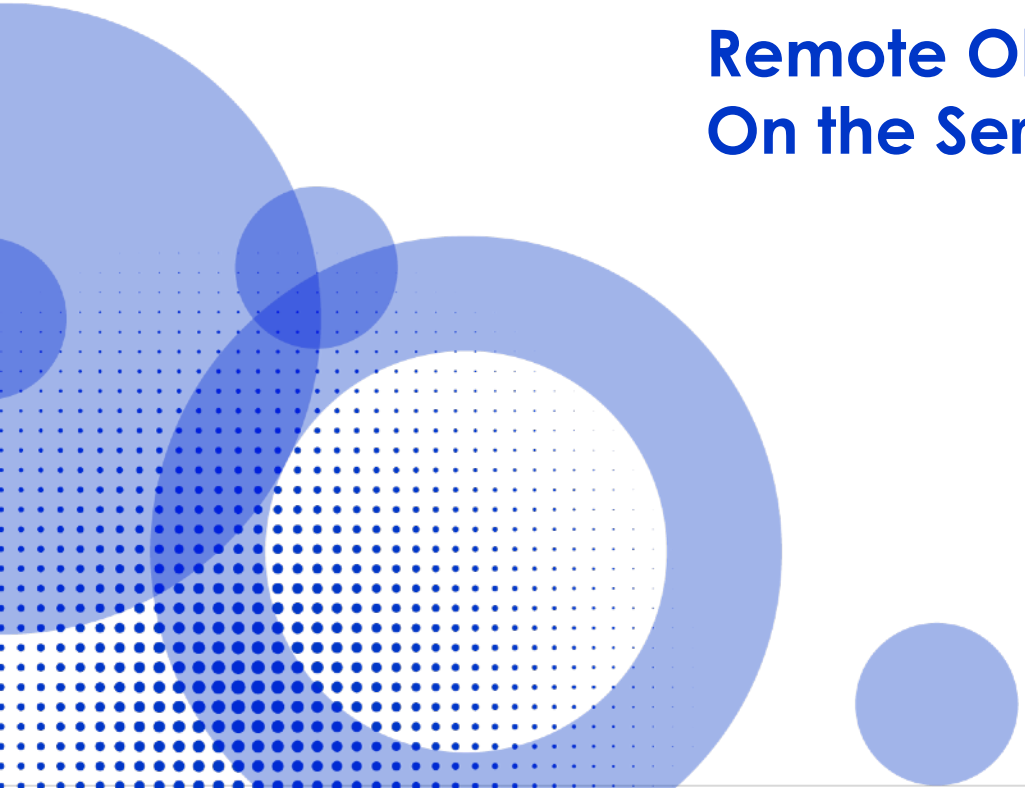

```
package net.omnis.OmnisTest;
import net.omnis.OmnisCalls.*;
import java.util.Map;
import java.util.HashMap;
public class Test extends OModule
{
        public Response test(Map<String, Object> pParams)
        {
            Map<String, Object> data = new HashMap<>();
            data.put("my_return, "my_return_value");
            return new SendResponse(data);
        }
}

// return new SendError(data);
```

JavaWorker

- See the online help for java toolsets & building information
- This is now the recommended route when using Java modules
- Great example supplied in the Omnis HUB



Remote Objects
On the Server



omnis

Remote Objects on the Server

- Remote objects can now be used on the server
  - *A great way to create common code for both client and server.*

- To use a remote object on the server, you can :
  - *Use a remote object as a superclass for a normal Object class*
  - *Select a remote object as a subtype of an object variable.*
  - *Use $new() and $newref() methods of the remote object*

Remote Objects on the Server

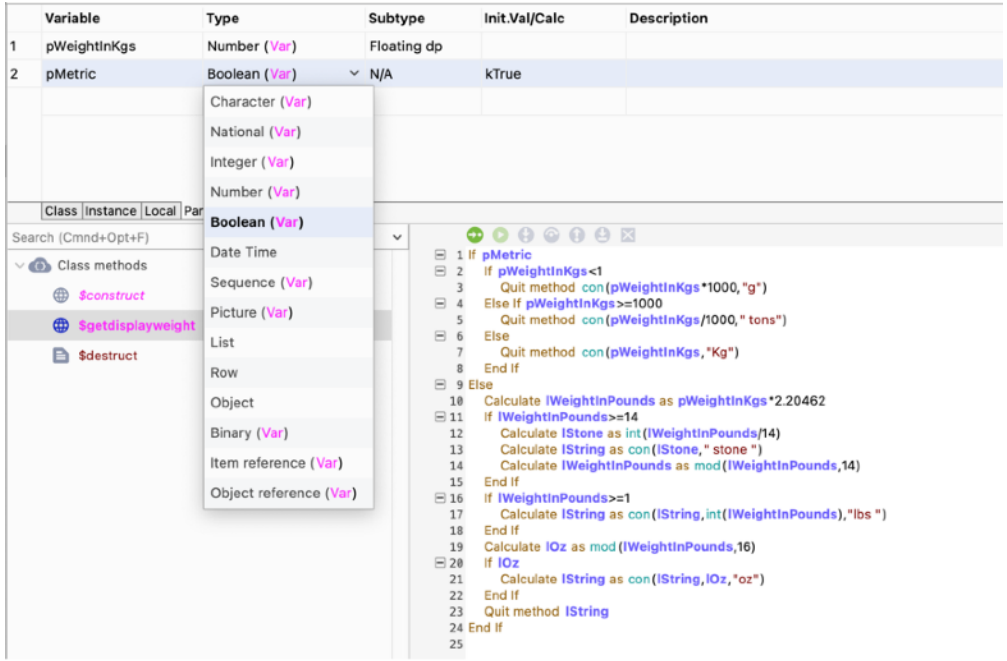
- Restrictions
  - *Remote Object method code on the server has the same limitations as client side executed code*
  - *If a remote form is using an **object instance variable** and the **subtype** is a remote object, the client and server do not share the same object, each side has a copy of the object.*

Client code on the Server

- Control over the data type of a ‘var’ variable on the server

  Given client executed code, variables on the client that would be treated as **Var,** on the server can now be given a type.

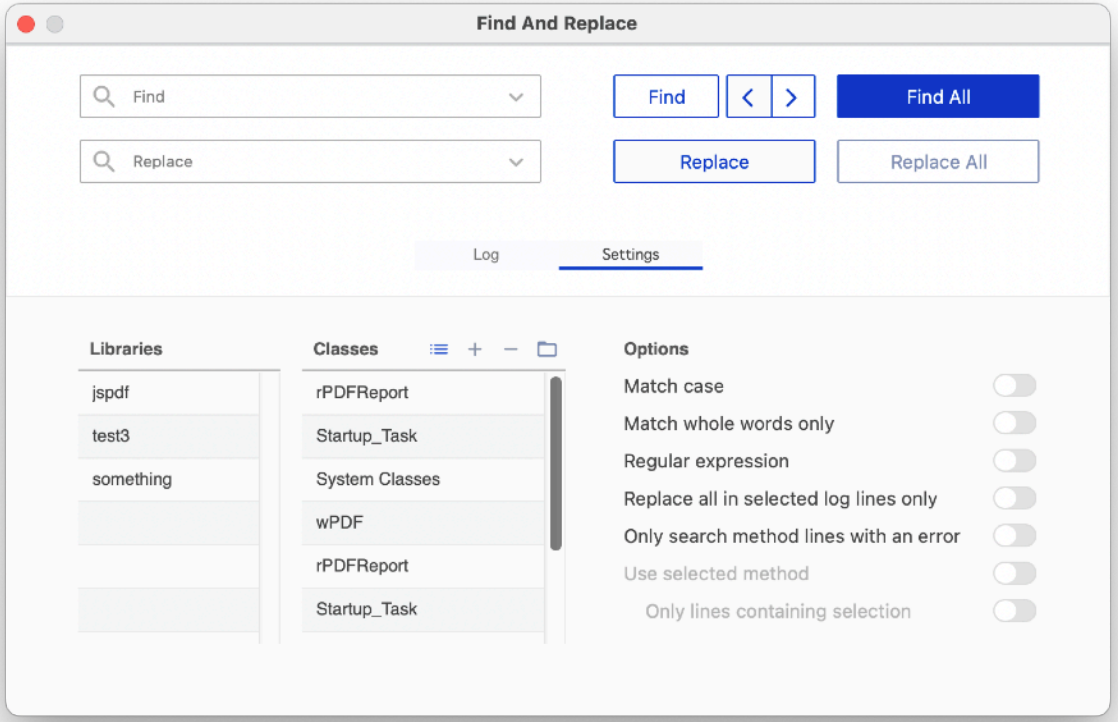
  eg. pMetric in this example is a Boolean, not a character



Find & Replace

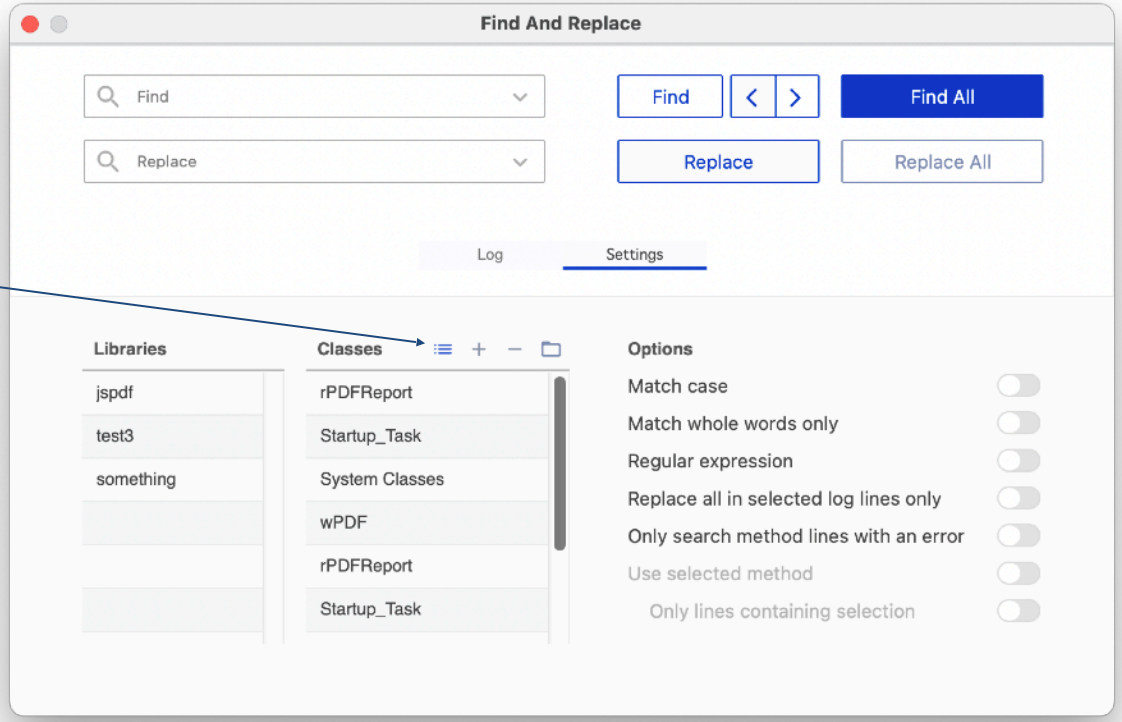
Find and Replace - Redesign

- Updated Design
  - *Updated UI continues with the new Studio 11 design*



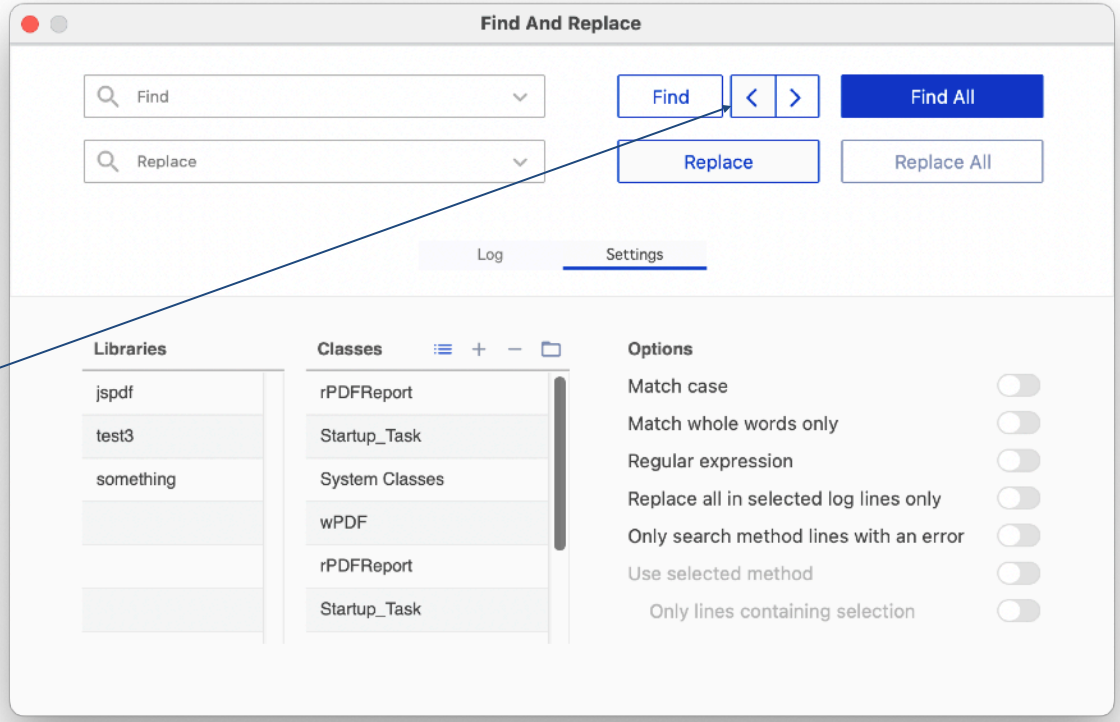
Find and Replace - Redesign

- Updated Design
  - *Updated UI continues with the new Studio 11 design*
  - *A New Select All Classes option*



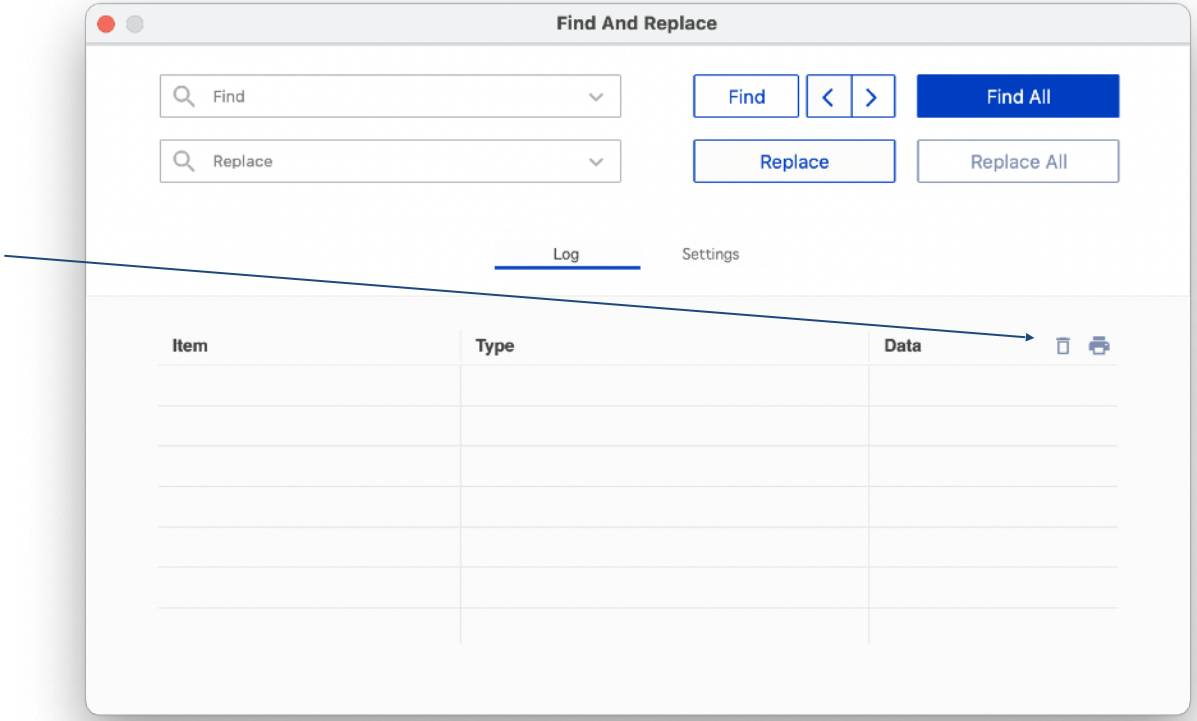
Find and Replace - Redesign

- Updated Design
  - *Updated UI continues with the new Studio 11 design*
  - *New Select All Classes option*
  - *Added Find Previous*



Find and Replace - Redesign

- Updated Design
  - *Updated UI continues with the new Studio 11 design*
  - *Clear Log and Print now icons in list header.*



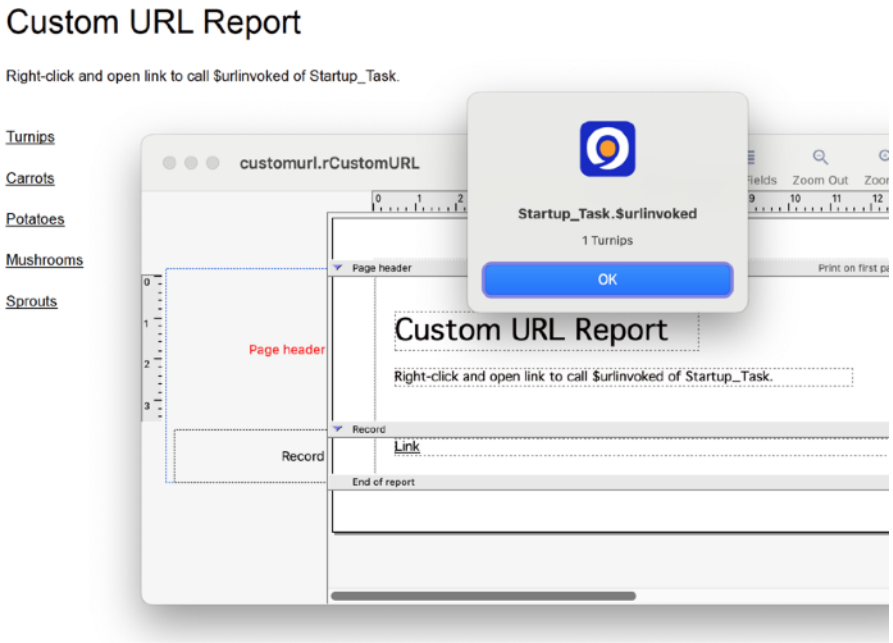
Custom URLs

Custom URLs

- Custom URL schemes added to Omnis
  - *On macOS these must be defined in the info.plist*
  - *On Windows these are defined in the section **customURLSchemes** in config/userconfig.json*
  - *A scheme using the Omnis version number is added by default eg. **Omnis 11.1** creates a scheme **studio111://***
  - *Once Omnis has started, schemes will be registered.*

Custom URLs

- Notifications in Omnis
  - *If Omnis is not running, it will start when a URL link is clicked*
  - *At least 1 parameter must be included in the URL scheme **lib***
  - ***$urlinvoked** in the startup task will be called in the specific library **lib***
  - *Ideally URLs should be escaped*
    *You can use **OW3.$escapeuritext()** to encode a URL.*
  - *Great example in HUB linking from a PDF*



PDF Universal Accessibility

PDF/UA Support

- PDF/UA (PDF Universal Accessibility)

  - *Many companies have requirements to create PDF/UA **compliant** documents. To enable PDF/UA, we've added a new PDF subset and version.*
  - *New PDF Device subset **kDevOmnisSubsetPDFUA***
  - *New PDF Device version **kDevOmnisPDFVersion17***

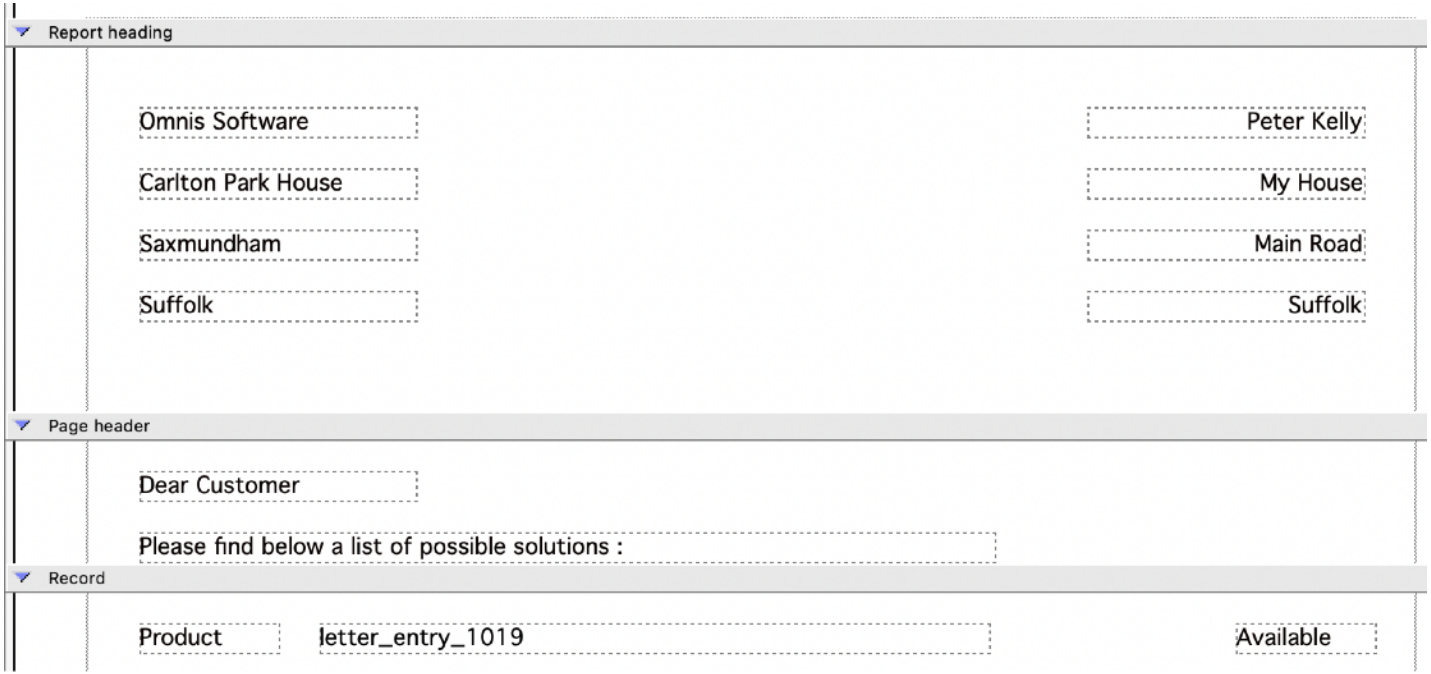
    Given an accessibility reader, PDF/UA specifies the PDF should be read in a specific order, an order that makes sense.

PDF/UA Support

- Current Omnis Report Engine - How it works
  - *Object locations can be manipulated using position sections and $print*
  - *Design mode shows the report structure but not the final order of objects*
  - *Preview is a more complete stage to review a final object order*
  - *The final order of objects controls what is sent to the PDF Device*
  - *Omnis Preview, by default orders objects from left to right, top to bottom Unaltered - This could lead to a incorrect reading order*

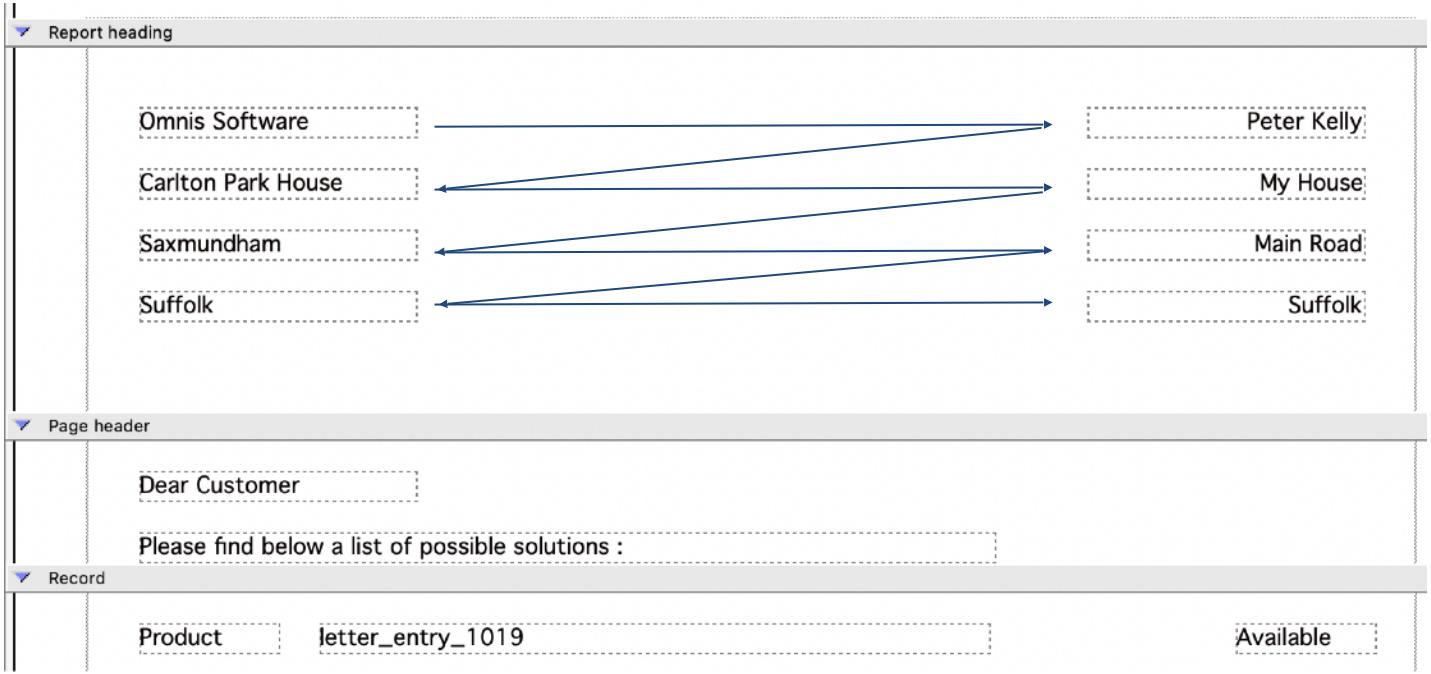
PDF/UA Support

- What is an incorrect reading order?



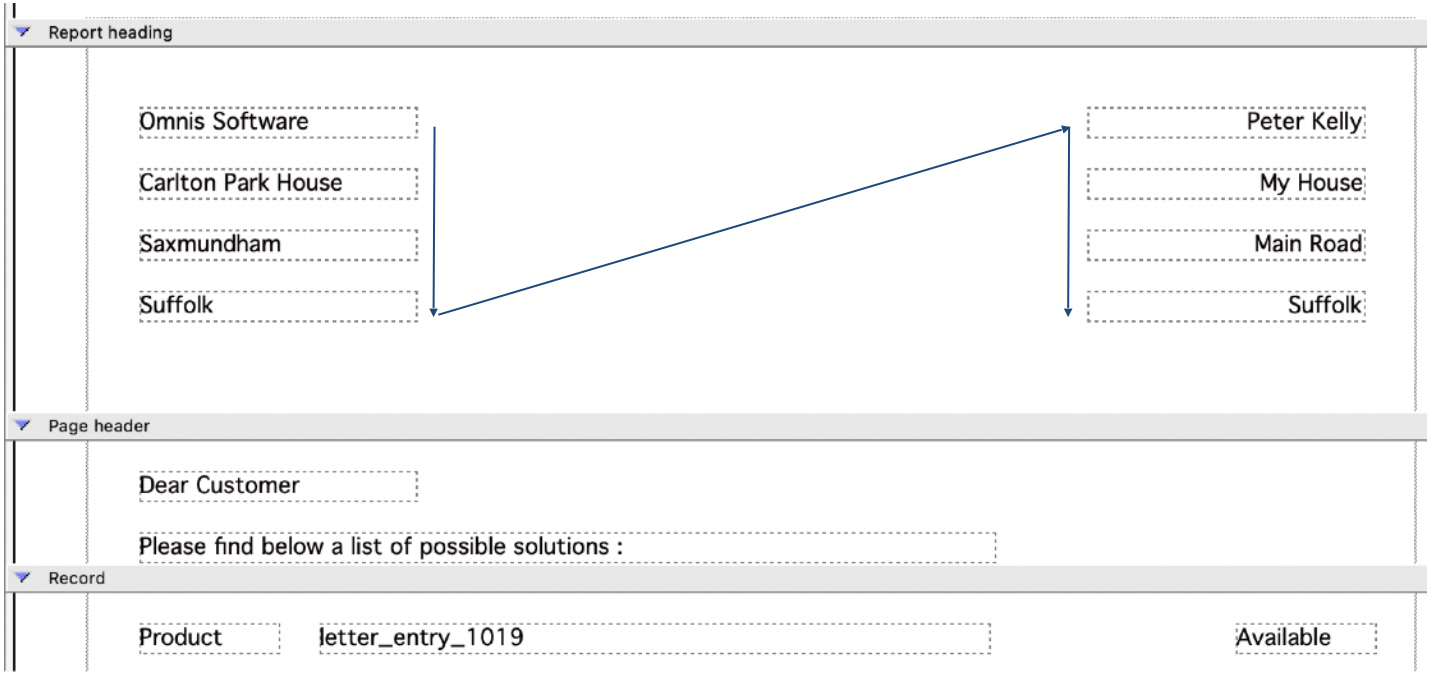
PDF/UA Support

- Omnis by default processes, left to right, top to bottom



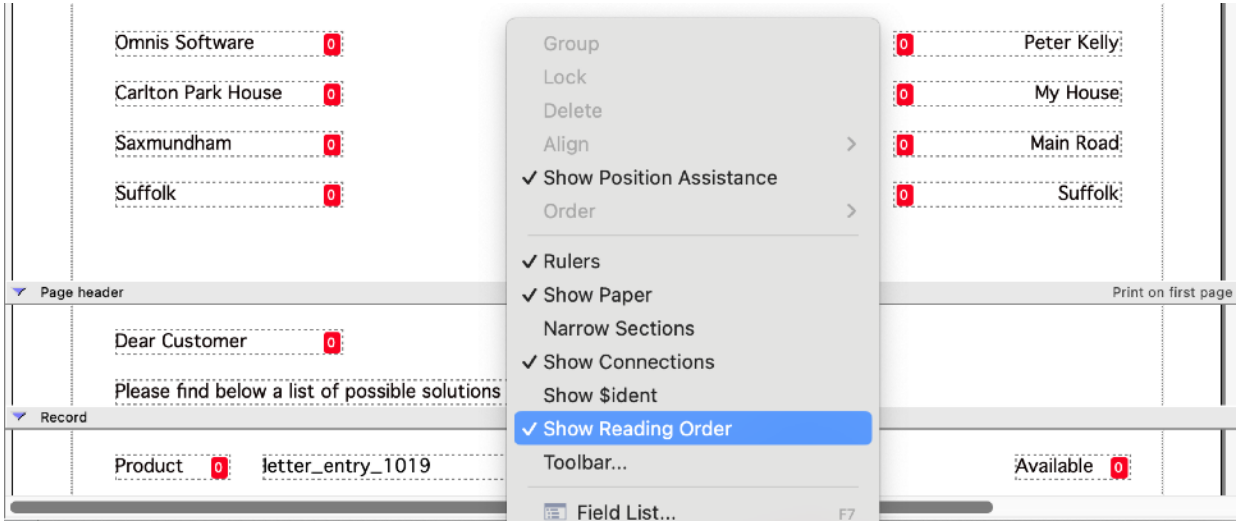
PDF/UA Support

- Ideally, a PDF/UA reader would like objects in a more natural order



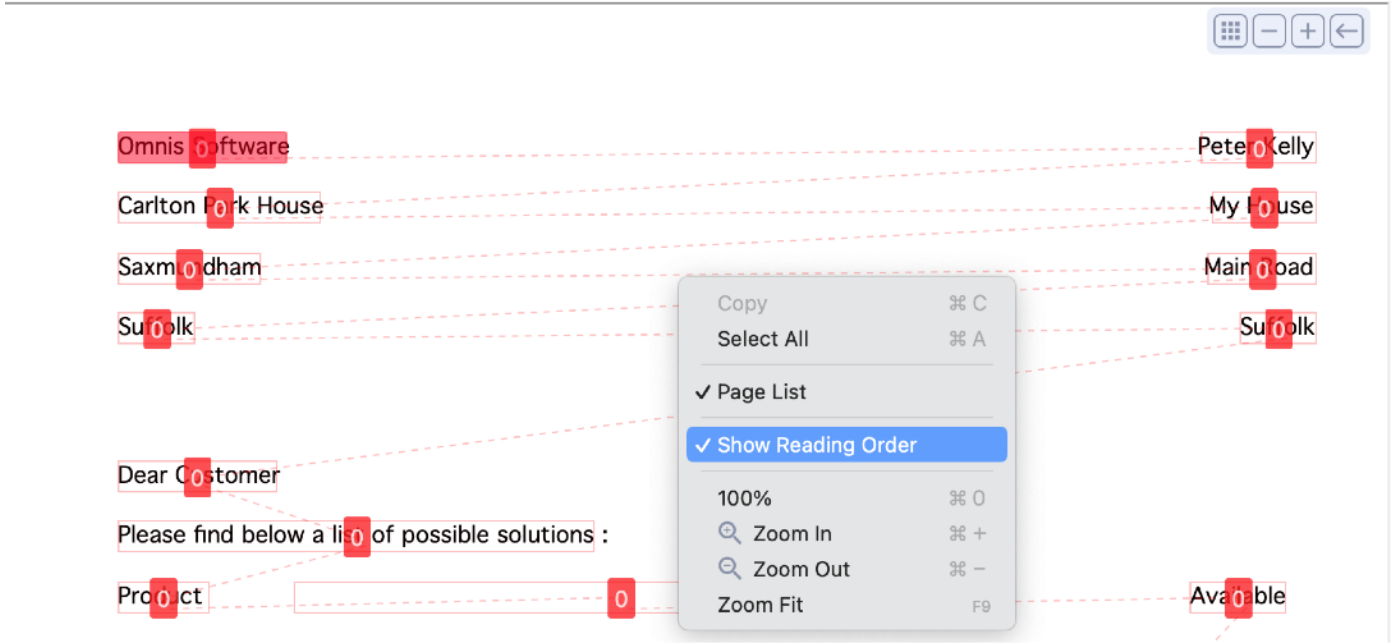
PDF/UA Support

- PDF/UA - Creating a better output
  - *New option in design mode to see the reading order*
  - *Items in red with a reading order of zero are not defined*



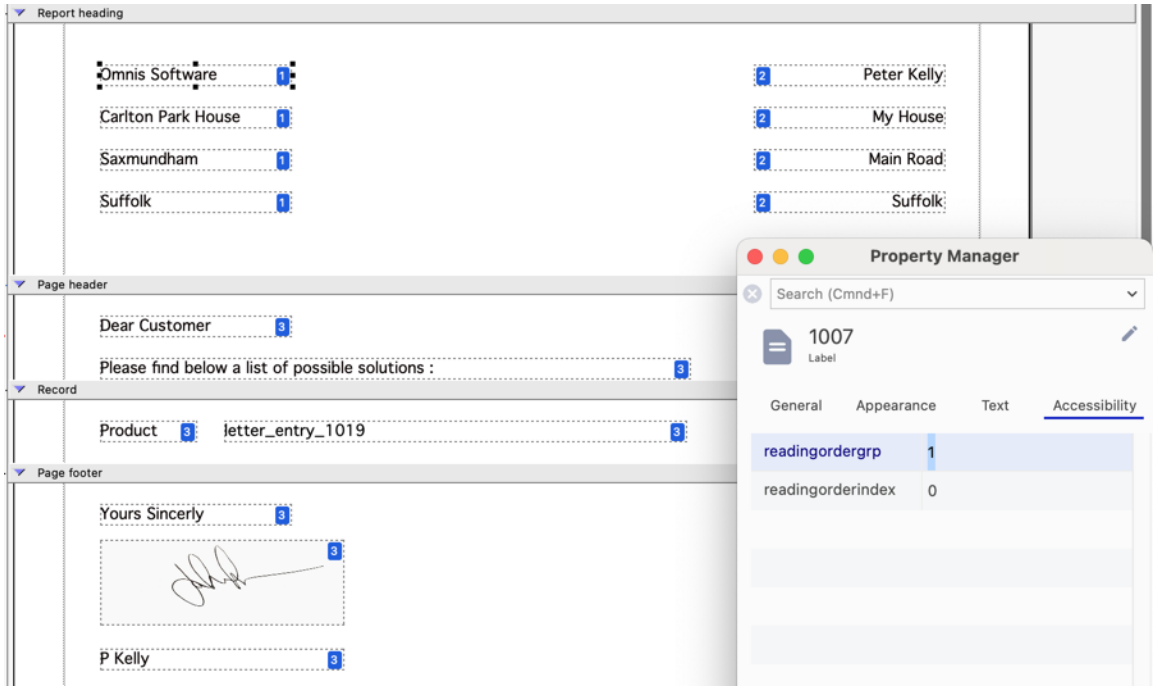
PDF/UA Support

- *Visual reading order aid available in preview mode*



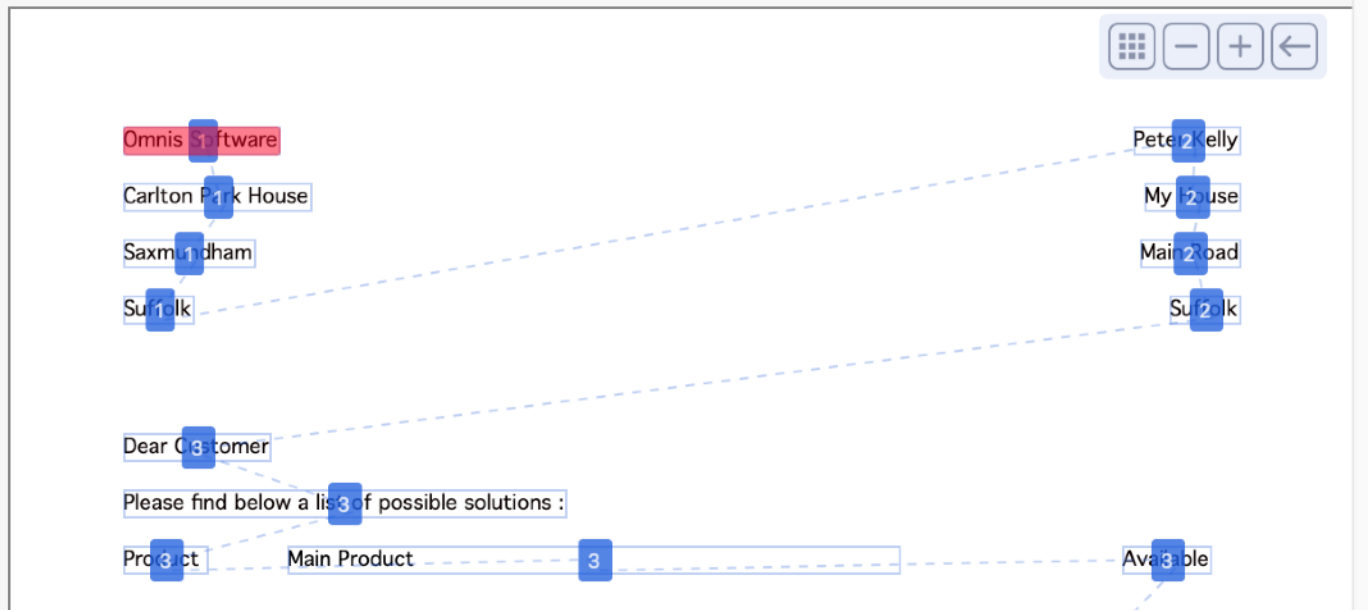
PDF/UA Support

- Controlling Object Reading Order - Design mode
  - *$readingordergrp*
  - *$readingorderindex*



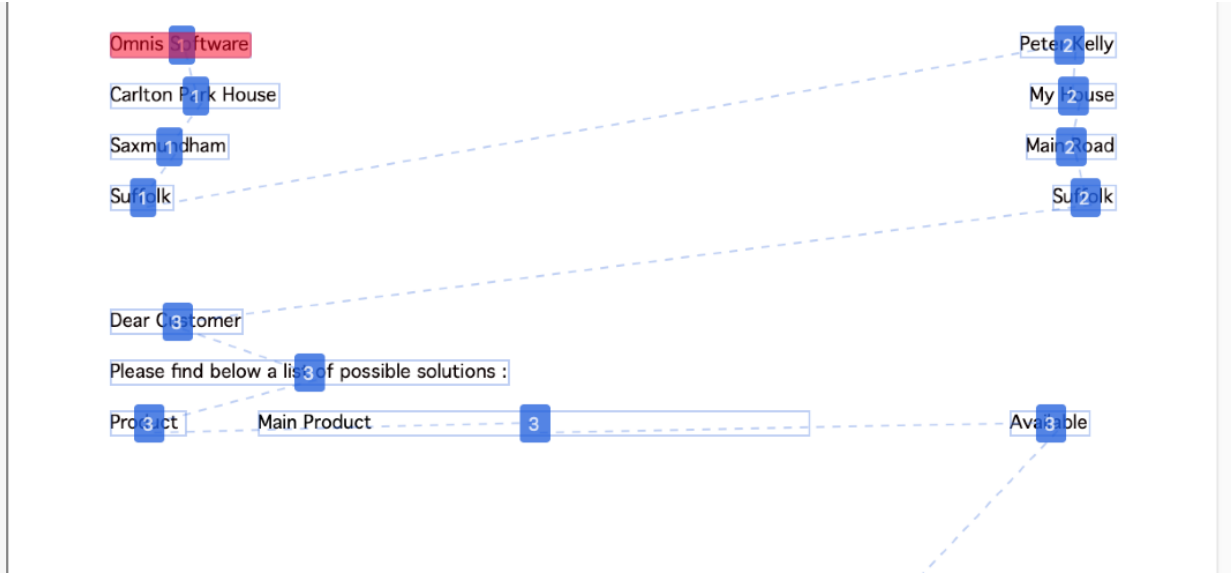
PDF/UA Support

- Now we can see how the reading order changes the object output
- One we are happy with the order, we can commit to PDF



PDF/UA Support

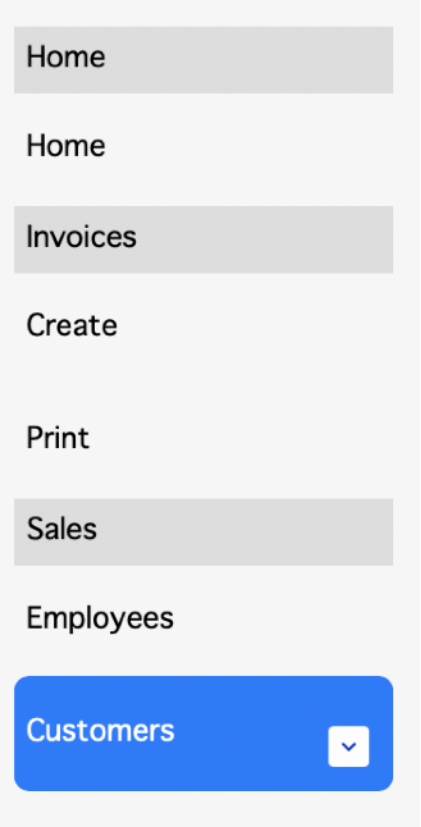
- Summary : Review the report in preview mode
- Tweak the reading order in design mode and re-review the report in preview
- Set the subset and version of the PDF device before printing to PDF



Extra Items

Other features

- Tab Strip
  - *New $expandedtabs mode*
    Shows all tabs expanded in their groups
  - *Supported in vertical mode only*
  - *Control over font, size, colors*
    and styles of groups
  - *Scroll buttons auto scroll on mouse enter*



Other features

- HTTP Worker Object
  - *AWS Signature V4 Authentication (* kOW3httpAuthTypeAWSv4Support)
  - *NTLM Authentication (* kOW3httpAuthTypeNTLM)

  See WhatsNew for more information for these new authentication types

- JS - HTML Link Control
  - *Single link control to open another browser or jump to another control.*

- ES Module Support for JS Worker
  - *The JS Worker can now load the newer format 'ECMAScript' modules, as well as 'CommonJS' modules.*