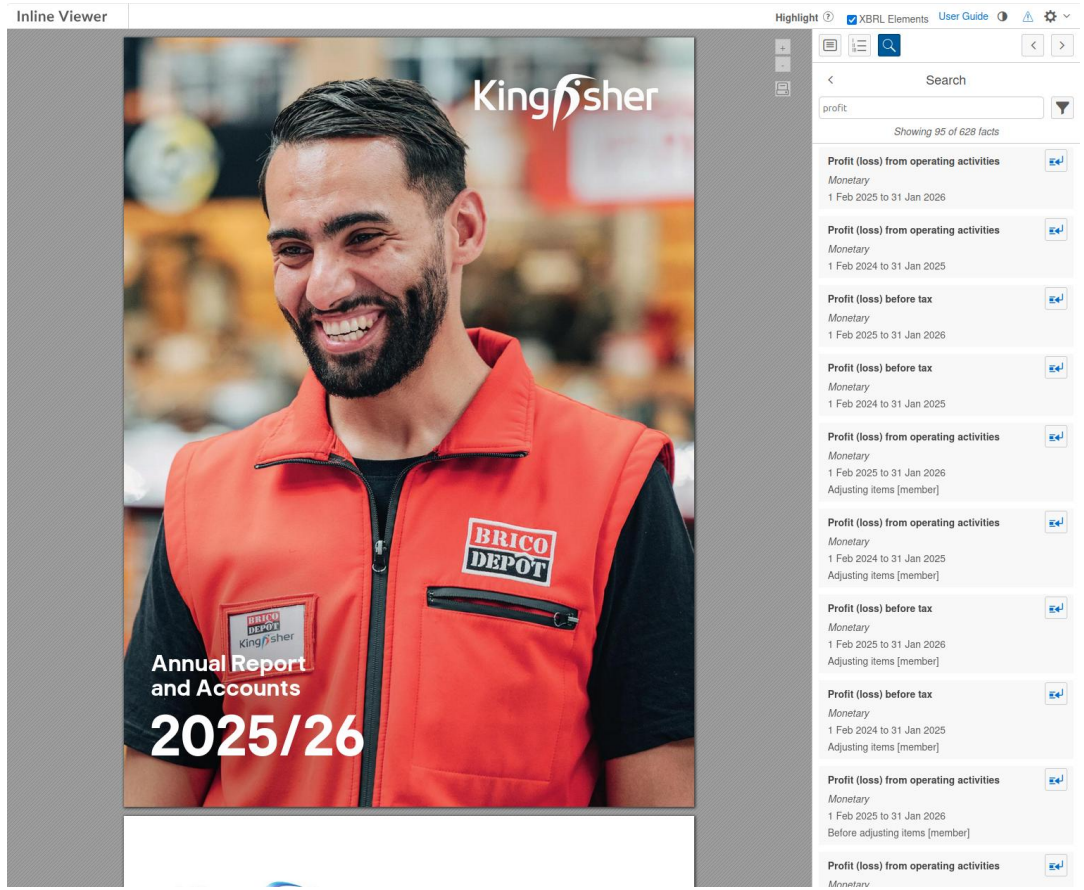
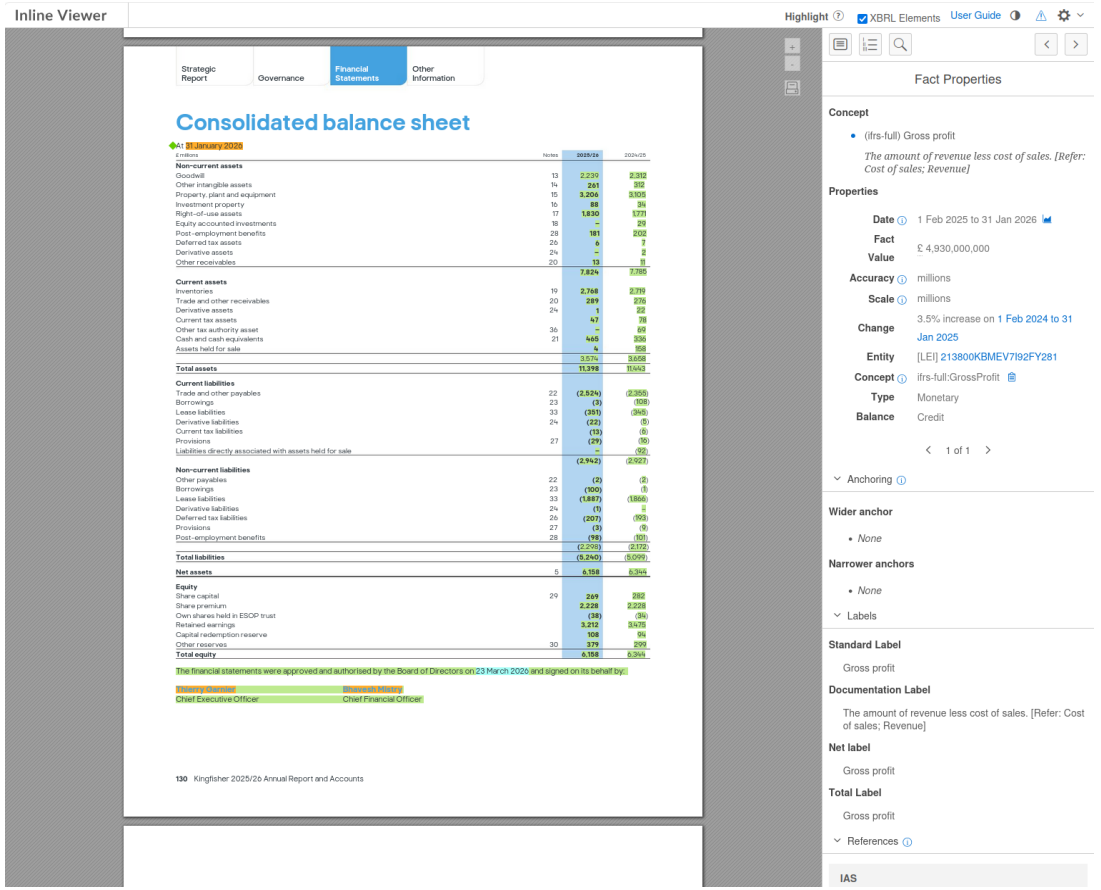

# Next steps for Inline XBRL

- Paul Warren
- XBRL International

# Inline XBRL today

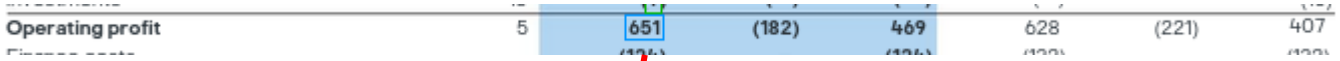


# Inline XBRL today

- Provides structured data in "open" reports
- Preparers retain control over both content and appearance of their reports
- Provides interactive reports to consumers
- Greatly enhances the accuracy and effectiveness of AI analysis

# How does Inline XBRL work



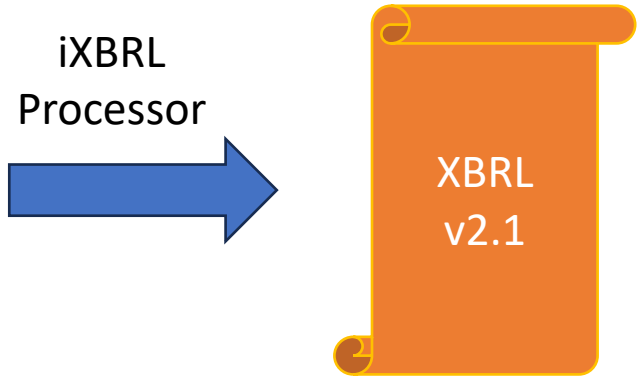
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Operating profit | 5 | 651 | (182) | 469 | 628 | (221) | 407 |

```
<ix:nonFraction
    contextRef="c-3"
    decimals="-6"
    format="ixt5:num-dot-decimal"
    id="fact-59"
    name="ifrs-full:ProfitLossFromOperatingActivities"
    scale="6" unitRef="u-2">651</ix:nonFraction>
```

Transformation

XBRL Metadata

iXBRL Processor



# Timeline

- 2008 - Inline XBRL v1.0 initial Public Working Draft
- 2010 – Inline XBRL v1.0 Recommendation
- 2013 – Inline XBRL v1.1 Recommendation
- 2026 – Inline XBRL v1.1 Errata update

*Inline XBRL embeds XBRL v2.1 syntax into XHTML, and produces an XBRL v2.1 (xBRL-XML) Report. XBRL v2.1 was finalised in 2003.*

13 years to our first errata release?

It must be pretty close to perfect!

# Known issues - XHTML

- Uses XHTML and XML Schema
- HTML 5 was finalised in 2014
- Lack of HTML 5 elements is not a practical problem for design (see ESEF filings!)
- … but it is a perceived problem.
- Lack of HTML 5 semantic elements is an issue for accessibility

# Known issues – xBRL-XML

- Inline XBRL is closely tied to XBRL v2.1's XML syntax
- Re-uses XBRL v2.1 syntax directly for contexts and units
- Specification tells you how to convert Inline XBRL to an XBRL v2.1 report
- Open Information Model was finalised in 2021
- Inline XBRL needs updating

# Known issues – use of custom elements

- Inline XBRL relies on embedding custom XML elements into HTML
- Can increase DOM size significantly
- Creates complexity for viewer software
- Elements are named after what they're not ("nonFraction", "nonNumeric")
- Not a big problem, but not how we'd do it today.

# How would we do it today?

Disclaimer: just my own ideas* – no firm plan to do this.

* Mostly not even my ideas

# Option 1:

- HTML 5 supports custom data attributes
- Modest changes to use data attributes instead of custom elements.

```
<span
    data-ix-name="ifrs:Profit"
    data-ix-scale="6"
    data-ix-decimals="-6"
    data-ix-format="ixt:num-comma-dot" >123</span>
```

# Dimensions – Option 1a

```
<span
    data-ix-name="ifrs:Profit"
    data-ix-scale="6"
    data-ix-decimals="-6"
    data-ix-format="ixt:num-comma-dot"
    data-ix-dimensions='{"ifrs:Region":"iso3166:GB","units":"iso4217:GBP"}'
    >123</span>
```

Block of JSON as an attribute value?

# Dimensions – Option 1b

```
<span
    data-ix-name="ifrs:Profit"
    data-ix-scale="6"
    data-ix-decimals="-6"
    data-ix-format="ixt:num-comma-dot"
    data-ix-dim-ifrs-Region="iso3166:GB"
    data-ix-dim-units="iso4217:GBP"
    >123</span>
```

One attribute per dimension? Needs care with naming conventions

# Option 2 – embed xBRL-JSON & link

- Minimise changes to HTML – mostly just needs ID attributes
- Transform becomes validation – eases consumption from trusted sources

```
<span id="id1">123</span>

<script type="application/json">
   {
        "f101": {
            "value": "123000000",
            "decimals": -6,
            "dimensions": {
                "concept": "ifrs:Profit",
                "units": "iso4217:GBP",
                "ifrs:Region": "iso3166:GB",
            },
            "ixt:format": "ixt:num-comma-dot",
            "ixt:source": [ "id1" ]
        }
    }
</script>
```

# Option 2

- Values are duplicated – what about text blocks?
- Exclude values for text blocks?
- Real value from tagging text blocks is from the linking?

```
<span id="id1">123</span>

<script type="application/json">
   {
        "f101": {
            "value": "123000000",
            "decimals": -6,
            "dimensions": {
                "concept": "ifrs:Profit",
                "units": "iso4217:GBP",
                "ifrs:Region": "iso3166:GB",
            },
            "ixt:format": "ixt:num-comma-dot",
            "ixt:source": [ "id1" ]
        }
    }
</script>
```

# Next steps

- We've repeatedly agreed that we should do a minimal update to Inline XBRL to accommodate HTML 5
- We've repeatedly failed to do this because of the unmaintainability of the XML Schemas
- Inline XBRL is a great success, but the syntax is old
- Maybe it's time for a less minimal modernisation?