FanXE
Fantastic XML Editor

Thomas C. Allard
tallard@frb.gov

Banking Analysis Section
Division of Monetary Affairs
Board of Governors of the Federal Reserve System
20th Street & Constitution Ave NW, Washington DC, 20551
Introduction

GUI overview
Introduction

- GUI overview
- Command Line Interface
Introduction

- GUI overview
- Command Line Interface
- Application Examples
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - ItemSet: Report Form Meta Data
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - ItemSet: Report Form Meta Data
  - BCBreaks: Storing XML in PostgreSQL
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - **ItemSet**: Report Form Meta Data
  - **BCBreaks**: Storing XML in PostgreSQL
  - **SDMX**: Time Series XML Data
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - **ItemSet**: Report Form Meta Data
  - **BCBreaks**: Storing XML in PostgreSQL
  - **SDMX**: Time Series XML Data
- Choosing Packages
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - **ItemSet**: Report Form Meta Data
  - **BCBreaks**: Storing XML in PostgreSQL
  - **SDMX**: Time Series XML Data
- Choosing Packages
  - **tdom** versus **tcldom**
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - **ItemSet**: Report Form Meta Data
  - **BCBreaks**: Storing XML in PostgreSQL
  - **SDMX**: Time Series XML Data
- Choosing Packages
  - **tdom** versus **tcldom**
  - **Tkhtml** versus **ihtml**
Introduction

- GUI overview
- Command Line Interface
- Application Examples
  - **ItemSet**: Report Form Meta Data
  - **BCBreaks**: Storing XML in PostgreSQL
  - **SDMX**: Time Series XML Data
- Choosing Packages
  - **tdom** versus **tcldom**
  - **Tkhtml** versus **ihtml**
  - **Tix** versus **BLT**
GUI Overview
Tree Pane

XML Documents (DOMs)

Call Report Index
- Uniform Item Set
  - 00001-09996: Hidden
  - 09997-14999: Income
  - 15000-19999: Expense
  - 20000-24999: Net income
  - 25000-39999: Assets
  - 40000-44999: Securities (held to maturity)
  - 41000-41999: Securities (available for sale)
  - 42000-44999: Total securities
  - 45000-54999: Liabilities
  - 55000-59999: Capital accounts
  - 60000-60999: Regulatory capital
  - 61000-62999: Miscellaneous
  - 63000-64999: Maturity and repricing
  - 65000-67399: Net charge-offs
  - 67400-69699: Gross charge-offs
  - 69700-71999: Recoveries
  - 72000-74299: Delinquencies
  - 74299-76599: Past-due 30 days or more
  - 76600-78899: Nonaccrual
  - 78900-79099: Renegotiated (troubled) debt
  - 79100-79999: Past-due 90 days or more
  - 80000-89999: Off-balance sheet
  - 90000-94999: Schedule K Assets
  - 95000-99999: Schedule K Liabilities

Fed Nomenclature Tree
- Size Groups
- Seasonal Adjustment
- New Items
- FanXE Configuration
- User Preferences
Tree Pane

- Call Report Index
  - Uniform Item Set
    - 00001-09996: Hidden
    - 09997-14999: Income
    - 15000-19999: Expense
    - 20000-24999: Net income
    - 26000-39999: Assets
      - 26000-26999: Total Assets
      - 27000-27999: STFBM
      - 28000-28999: Total Assets
      - 29000-39999: Securities (held to maturity)
      - 41000-41999: Securities (available for sale)
      - 42000-44999: Total securities
      - 45000-54999: Liabilities
      - 55000-59999: Capital account
      - 60000-60999: Regulatory capital
      - 61000-62999: Miscellaneous
      - 63000-64999: Maturity and repricing
      - 65000-67399: Net charge-offs
      - 67400-69699: Gross charge-offs
      - 69700-71999: Recoveries
      - 72000-74299: Delinquencies
      - 74299-76599: Past-due 30 days or more
      - 76600-78999: Nonaccrual
      - 79000-79099: Renegotiated (troubled) debt
      - 79100-79999: Past-due 90 days or more
      - 80000-89999: Off-balance sheet
      - 90000-99999: Schedule-K Assets
      - 90000-99999: Schedule-K Liabilities
  - Fed Nomenclature Tree
  - Size Groups
  - Seasonal Adjustment
  - New Items
  - FanXE Configuration
  - User Preferences
Tree Pane

- Call Report Index
  - Uniform Item Set
    - 00000-09999: Hidden
    - 09999-13999: Income
    - 14000-19999: Expense
    - 20000-24999: Net Income
    - 25000-39999: Assets
      - 25000: STF
      - 26000: STFB
      - 29200: STFBAM
    - 40000-49999: Securities (held to maturity)
    - 50000-51999: Securities (available for sale)
    - 42000-44999: Total securities
    - 45000-54999: Liabilities
    - 55000-59999: Capital account
    - 60000-60999: Regulatory capital
    - 61000-62999: Miscellaneous
    - 63000-64999: Maturity and repricing
    - 65000-67399: Net charge-offs
    - 67400-69699: Gross charge-offs
    - 69700-71999: Recoveries
    - 72000-74299: Delinquencies
    - 74299-76599: Past due 30 days or more
    - 76600-78899: Nonaccrual
    - 78900-79099: Renegotiated (troubled) debt
    - 79100-79999: Past due 90 days or more
    - 80000-89999: Off-balance sheet
    - 90000-94999: Schedule-K Assets
      - 90000: STFA
      - 90100: STFB
    - 95000-99999: Schedule-K Liabilities
  - Fed Nomenclature Tree
  - Size Groups
  - Seasonal Adjustment
  - New Items
  - FanXE Configuration
  - User Preferences
Tree Pane

Selecting a node on the tree will apply a style sheet and display the results in the View Pane.
25000 STFBA Total Assets

Formula

Date: 1984/09-9999/12
= +CAL25300 +CAL37300 (Depth 10)
= +STFBAI +STFBAN

Notes: RCFD2170
Selecting the `Edit` button will apply another style sheet and open the results in the Tab Paned.
Tab Pane
### Tab Pane

![Image of a Tab Pane with a screenshot of Tab X of a Fantastic XML Editor (FanXE) window showing a spreadsheet with a section of XML Code]

**Spreadsheet Section:**

<table>
<thead>
<tr>
<th>N</th>
<th>Historical</th>
<th>Fedname</th>
<th>Quarterly Average</th>
<th>Attributable (Domestic Foreign)</th>
<th>Booked (Domestic Foreign)</th>
<th>Market Value Securities</th>
<th>Book Value Securities</th>
<th>Taxable Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>25000</td>
<td>HO</td>
<td>STFBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Total Assets

**Type:** Formula

**Date:** 19840930 99991231

**Depth:** 10

**Alias:** +STFBAI +STFBAN

**XML Code:**

```xml
<Section Range="25000-39999" Title="Assets">
  <Series N="25000" Fedname="STFBA" Total Assets>
    <Item Type="Formula" BeginDate="19840930" EndDate="99991231"/>
    <Formula Depth="10">
      <CAL f="*">25300</CAL>
      <CAL f="*">37300</CAL>
      < Alias >+STFBAI +STFBAN</Alias>
    </Formula>
  </Item>
  <Note>RCFD2170</Note>
</Series>
```

**Notes:**

- RCFD2170
Tab Pane

### Total Assets

<table>
<thead>
<tr>
<th>Historical</th>
<th>FedName</th>
<th>Quarterly Average</th>
<th>Attributable (Domestic Foreign)</th>
<th>Booked (Domestic Foreign)</th>
<th>Market Value Securities</th>
<th>Book Value Securities</th>
<th>Taxable Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>25000</td>
<td>XDA</td>
<td>XDF</td>
<td>XDA</td>
<td>XDU</td>
<td>XBV</td>
<td>XBV</td>
<td>XTE</td>
</tr>
</tbody>
</table>

**Description:** Total Assets

**Type:** Formula

**Date:** 19840930  99991231

**Depth:** 10

**Alias:** +STFBAI +STFBAN

```xml
<Section Range="25000-39999" Title="Assets">  
  <Series N="25000" Fedname="STFBA" Title="Total Assets">  
    <Item Type="Formula" BeginDate="19840930" EndDate="99991231">  
      <Formula Depth="10">  
        <CAL f="*">25300</CAL>  
        <CAL f="*">87300</CAL>  
        <Atlas> +STFBAI +STFBAN</Atlas>  
      </Formula>  
    </Item>  
  </Series>  
</Section>
```
Tab Pane
Tab Pane
Tab Pane
Tab Pane

![Diagram of Tab Pane](image)

### XML Code Example

```xml
<Section Range="25000-39999" Title="Assets">
  <Series №="25000" Fname="STFBA">Total Assets
    <Item Type="Formula" BeginDate="19840930" EndDate="99991231">
      <Formula Depth="10">
        <CAL f="+">25300</CAL>
        <CAL f="+">37300</CAL>
        <Alias> +STFBAI +STFBAN</Alias>
      </Formula>
    </Item>
  </Series>
</Section>
```

### Table

<table>
<thead>
<tr>
<th>N</th>
<th>Historical</th>
<th>Fname</th>
<th>Quarterly Average</th>
<th>Market Value</th>
<th>Book Value</th>
<th>Taxable Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>25000</td>
<td>HO</td>
<td>XQA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Add an Input Series
- Add a Formula
- Add a Split
- Add a Placeholder
GUI Configuration
“Name” of the element to be displayed in the Tree.
GUI Configuration

“Name” of the element to be displayed in the Tree.
“Name” of the element to be displayed in the Tree.
The “Title” contains Tcl code to create the Tree label.
set title "[node @Range]: [node @Title]"

This code is executed in a safe Tcl interpreter. The node command is aliased to a tdom node object.

The final value of $title will be the label in the Tree.
set title "[node @Range]: [node @Title]"
“Browse” contains the name of a style sheet that is applied when selecting a node in the Tree.
GUI Configuration

“Edit” contains the name of a style sheet that is applied when the button is pressed.
"DisplayStyle" contains the name of a **Tix** DisplayStyle object.
GUI Configuration

DisplayStyles control all font properties in the Tree.
Pane Layout
Special Links

25000 STFBA Total Assets

**Formula**

*Date*: 1984/09-9999/12

= +CAL25300+CAL37300 (Depth 10)

= +STFBAI +STFBAN

**Notes**: RCFD2170
Special Links

<input type="image" name="edit" src="http://FanXE/Edit" value="Edit">

25000 STFBA Total Assets

Formula

Date: 1984/09-9999/12

= +CAL25300 + CAL37300 (Depth 10)
= +STFBAI + STFBAN

Notes: RCFD2170
Special Links

The FanXE hostname links to internal images.
Special Links

<a href="fanxe://UniformSeries//Series[@N='25300']">
The fanxe URI scheme links to XPath expressions.
The `fanxe` URI scheme links to XPath expressions.

**UniformSeries** is the DOM name.
The fanxe URI scheme links to XPath expressions.

UniformSeries is the DOM name.

//Series[@N='25300'] is the XPath expression.
The fanxe URI scheme links to XPath expressions.

UniformSeries is the DOM name.

//Series[@N='25300'] is the XPath expression.

If multiple nodes match the XPath, the first one is opened and others are highlighted. Middle-clicking opens the first 10 matches in new tabs.
Special Links

SDMX Editor/Generator

- Open a Structure file.
- Open a Config file.
- Open a Key Family Schema.
- Create a new Structure file.
- Create a new Config file.

- The sdmx manual page.

Click the icon above to return to this screen.

<table>
<thead>
<tr>
<th>Convert a Structure file to Schema(s)</th>
<th>Create a Data file from a Config file</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>DOM</td>
<td>ID</td>
</tr>
<tr>
<td>h15 struct</td>
<td>H15</td>
</tr>
<tr>
<td>Common Structure File</td>
<td>FRB Common Structure</td>
</tr>
</tbody>
</table>
Special Links

<a href="http://FanXE/build?struct&h15_struct.xml">FanXE — Fantastic XML Editor</a>
The FanXE hostname links to public methods.
Special Links

build is the method name.

SDMX Editor/Generator

- Open a Structure file.
- Open a Config file.
- Open a Key Family Schema.
- Create a new Structure file.
- Create a new Config file.

The sdmx manual page.

Click the icon above to return to this screen.

<p>| Convert a Structure file to Schema(s) |</p>
<table>
<thead>
<tr>
<th>DOM</th>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>h15_struct</td>
<td>H15</td>
<td>H.15 Selected Interest Rates</td>
</tr>
<tr>
<td>Common Structure File</td>
<td></td>
<td>FRB Common Structure</td>
</tr>
</tbody>
</table>

<p>| Create a Data file from a Config file |</p>
<table>
<thead>
<tr>
<th>DOM</th>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>h15_config</td>
<td>H15</td>
<td>H.15 Selected Interest Rates</td>
</tr>
</tbody>
</table>
<a href="http://FanXE/build?struct&h15_struct.xml">

struct and h15_struct.xml are the arguments to the method.

</a>
Shell script to instantiate object and execute a method.
Command Line Interface

- Shell script to instantiate object and execute a method.

- Usage:
  
  fanxe [CLASS OPTIONS --] [method [ARGS]]
Shell script to instantiate object and execute a method.

Usage:
fanxe [CLASS OPTIONS --] [method [ARGS]]

Example:
fanxe -useRCS 0 -- \
  transform myfile.xml stylesheet.xsl \ 
  -parameters ’param1 v1 param2 v2’
Command Line Interface

Shell script to instantiate object and execute a method.

Usage:
fanxe [CLASS OPTIONS --] [method [ARGS]]

Example:
fanxe -useRCS 0 -- \transform myfile.xml stylesheet.xsl \-parameters ’param1 v1 param2 v2’

Default method, if X is available, is GUI.
**transform method**

- Runs an XSLT transformation.
transform method

- Runs an XSLT transformation.
- Uses `tdom`'s XSLT engine.
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds xsl:document support (writes to multiple files from a single transformation).
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds xsl:document support (writes to multiple files from a single transformation).
- Includes FanXE extended XSLT functions.
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds `xsl:document` support (writes to multiple files from a single transformation).
- Includes FanXE extended XSLT functions.
  - `fanxe:tcl` will execute a string in a safe interpreter.
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds xsl:document support (writes to multiple files from a single transformation).
- Includes FanXE extended XSLT functions.
  - fanxe:tcl will execute a string in a safe interpreter.
  - fanxe:mtime will return file modification time (file not included in safe interpreters).
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds xsl:document support (writes to multiple files from a single transformation).
- Includes FanXE extended XSLT functions.
  - fanxe:tcl will execute a string in a safe interpreter.
  - fanxe:mtime will return file modification time (file not included in safe interpreters).
- exslt elements work with tdom.
transform method

- Runs an XSLT transformation.
- Uses tdom’s XSLT engine.
- Adds xsl:document support (writes to multiple files from a single transformation).
- Includes FanXE extended XSLT functions.
  - fanxe:tcl will execute a string in a safe interpreter.
  - fanxe:mtime will return file modification time (file not included in safe interpreters).
- exslt elements work with tdom.
- FanXE adds partial support for exslt:seconds and exslt:date functions with Tcl code.
transform method

- Runs an XSLT transformation.
- Uses \texttt{tdom}'s XSLT engine.
- \texttt{xsl:document} support (writes to multiple files from a single transformation).
- Includes \textbf{FanXE} extended XSLT functions.
  - \texttt{fanxe:tcl} will execute a string in a safe interpreter.
  - \texttt{fanxe:mtime} will return file modification time (file not included in safe interpreters).
- \texttt{exslt} elements work with \texttt{tdom}.
  - \textbf{FanXE} adds partial support for \texttt{exslt:seconds} and \texttt{exslt:date} functions with Tcl code.
  - Subclasses can add their own XSLT functions.
Application Examples

**ItemSet:** Report Form Meta Data

- Driving factor in motivation and design.
Application Examples

**ItemSet**: Report Form Meta Data

- Driving factor in motivation and design.
- Documents stored under revision control.
Application Examples

**ItemSet:** Report Form Meta Data

- Driving factor in motivation and design.
- Documents stored under revision control.
- Difference reports across revisions.
Application Examples

ItemSet: Report Form Meta Data

- Driving factor in motivation and design.
- Documents stored under revision control.
- Difference reports across revisions.
- CGI interface.
Application Examples

**BCBreaks**: Storing XML in PostgreSQL

- XML data edited by multiple users.
Application Examples

BCBreaks: Storing XML in PostgreSQL

- XML data edited by multiple users.
- Added `<loadDOM>`, `<saveDOM>` and `<saveNode>` elements to FanXE configuration.
Application Examples

**BCBreaks**: Storing XML in PostgreSQL

- XML data edited by multiple users.
- Added `<loadDOM>`, `<saveDOM>` and `<saveNode>` elements to *FanXE* configuration.
- `<loadDOM>` and `<saveDOM>` contain Tcl code that is called to load or save the entire document.
BCBreaks: Storing XML in PostgreSQL

- XML data edited by multiple users.
- Added `<loadDOM>`, `<saveDOM>` and `<saveNode>` elements to FanXE configuration.
  - `<loadDOM>` and `<saveDOM>` contain Tcl code that is called to load or save the entire document.
  - `<saveNode>` contains Tcl code that is called whenever a node is changed.
Application Examples

**BCBreaks**: Storing XML in PostgreSQL

- XML data edited by multiple users.

- Added `<loadDOM>`, `<saveDOM>` and `<saveNode>` elements to FanXE configuration.
  - `<loadDOM>` and `<saveDOM>` contain Tcl code that is called to load or save the entire document.
  - `<saveNode>` contains Tcl code that is called whenever a node is changed.

- **BCBreaks’ `<saveNode>`** script logs the change and issues `NOTIFY tableUpdated` to PostgreSQL server.
**Application Examples**

**BCBreaks**: Storing XML in PostgreSQL

- XML data edited by multiple users.
- Added `<loadDOM>`, `<saveDOM>` and `<saveNode>` elements to FanXE configuration.
  - `<loadDOM>` and `<saveDOM>` contain Tcl code that is called to load or save the entire document.
  - `<saveNode>` contains Tcl code that is called whenever a node is changed.
- BCBreaks’ `<saveNode>` script logs the change and issues NOTIFY `tableUpdated` to PostgreSQL server.
- GUI listens for `tableUpdated` and changes are applied immediately.
Application Examples

SDMX: Time Series XML Data

- Application front-end.
Application Examples

**SDMX:** Time Series XML Data

- Application front-end.
- Users open their own XML documents, not just what was defined in the configuration file.
Application Examples

**SDMX**: Time Series XML Data

- Application front-end.
- Users open their own XML documents, not just what was defined in the configuration file.
- Files can be opened by cloning the structure of other XML documents defined in the configuration file.
tdom versus tcldom

- Initially supported both.
tdom versus tcldom

- Initially supported both.
- tdom had an XSLT bug I couldn’t work around.
tdom versus tcldom

- Initially supported both.
- tdom had an XSLT bug I couldn’t work around.
- Rolf Ade fixed that bug and all others that I’ve reported.
**tdom versus tcldom**

- Initially supported both.
- **tdom** had an XSLT bug I couldn’t work around.
- Rolf Ade fixed that bug and all others that I’ve reported.
- **tdom** has the fastest XSLT engine.
Tkhtml versus ihtml

Already using [incr Tcl] and the iwidgets panedwindow.
Tkhtml versus ihtml

- Already using [incr Tcl] and the iwidgets panedwindow.
- ihtml did not support nested tables well.
Tkhtml versus ihtml

- Already using [incr Tcl] and the iwidgets panedwindow.
- ihmtl did not support nested tables well.
- Tkhtml does not support CSS.
Tkhtml versus ihtml

- Already using [incr Tcl] and the iwidgets panedwindow.
- ihtml did not support nested tables well.
- Tkhtml does not support CSS.
- Little ongoing development of Tkhtml.
Tkhtml versus ihtml

- Already using [incr Tcl] and the iwidgets panedwindow.
- ihtml did not support nested tables well.
- Tkhtml does not support CSS.
- Little ongoing development of Tkhtml.
  - Stable and no major bugs.
Tkhtml versus ihtml

- Already using [incr Tcl] and the iwidgets.
- ihtml did not support nested tables well.
- Tkhtml does not support CSS.
- Little ongoing development of Tkhtml.
  - Stable and no major bugs.
  - A new version in the works?
**Tix versus BLT**

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** *versus* **blt::hierbox**
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** versus `blt::hierbox`
  - Tree must be fast (C-coded).
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** versus `blt::hierbox`
  - Tree must be fast (C-coded).
  - **Tix** won by slight (insignificant) margin.
**Tix versus BLT**

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** versus **blt::hierbox**
  - Tree must be fast (C-coded).
  - **Tix** won by slight (insignificant) margin.
  - I’ve been working with **Tix** longer.
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** versus **blt::hierbox**
  - Tree must be fast (C-coded).
  - **Tix** won by slight (insignificant) margin.
  - I’ve been working with **Tix** longer.
  - DisplayStyles simplify configuration.
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.

  - **tixTree** versus **blt::hierbox**
    - Tree must be fast (C-coded).
    - **Tix** won by slight (insignificant) margin.
    - I’ve been working with **Tix** longer.
    - **DisplayStyles** simplify configuration.

- **blt::tabset** versus **tixNoteBook**
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).

- For every **Tix** widget, **BLT** (and others) have a counterpart.

- **tixTree** versus **blt::hierbox**
  - Tree must be fast (C-coded).
  - **Tix** won by slight (insignificant) margin.
  - I’ve been working with **Tix** longer.
  - DisplayStyles simplify configuration.

- **blt::tabset** versus **tixNoteBook**
  - Detachable tabs.
Tix versus BLT

- **BLT** has some irreplaceable commands (e.g. `blt::exec`).
- For every **Tix** widget, **BLT** (and others) have a counterpart.
- **tixTree** versus **blt::hierbox**
  - Tree must be fast (C-coded).
  - **Tix** won by slight (insignificant) margin.
  - I’ve been working with **Tix** longer.
  - DisplayStyles simplify configuration.
- **blt::tabset** versus **tixNoteBook**
  - Detachable tabs.
  - Detachable tabs.
BLT has some irreplaceable commands (e.g. `blt::exec`).

For every Tix widget, BLT (and others) have a counterpart.

tixTree versus `blt::hierbox`
- Tree must be fast (C-coded).
- Tix won by slight (insignificant) margin.
- I’ve been working with Tix longer.
- DisplayStyles simplify configuration.

`blt::tabset` versus tixNoteBook
- Detachable tabs.
Availability

- http://wklink.freeshell.org/
- Not always up-to-date. Ask!
- Created as part of my work for the Federal Government.

Publications Incorporating U. S. Government Works

Works by the U. S. Government are not eligible for U. S. copyright protection. For works published on and after March 1, 1989, the previous notice requirement for works consisting primarily of one or more U. S. Government works has been eliminated. However, use of a notice on such a work will defeat a claim of innocent infringement as previously described provided the notice also includes a statement that identifies either those portions of the work in which copyright is claimed or those portions that constitute U. S. Government material.


Copies of works published before March 1, 1989, that consist primarily of one or more works of the U. S. Government should have a notice and the identifying statement.
Thanks

- John Ousterhout for creating **Tcl/Tk**.
- Jeff Hobbes and the rest of the **Tcl/Tk** core team maintaining everything.
- Michael McLennan for **[incr Tcl]**.
- Jochen Löwer and Rolf Ade for **tdom**.
- D. Richard Hipp for **Tkhtml**.
- George Howlett for **BLT**.
- Ioi Lam for **Tix** (and others who have kept it working)
- Conference organizers.
- The entire community for providing answers whenever I’ve had questions.