

ExB XML Feed Widget

Widget Design Specification | Draft

ArcGIS Experience Builder Developer Edition

1. Purpose

This widget enables ArcGIS Experience Builder (Developer Edition) apps to consume and display data from XML-based feeds — including custom government XML schemas and standard RSS 2.0 feeds — without requiring that data to be hosted in a feature service. The widget supports two complementary modes of operation that can be enabled independently or together.

Mode A — Visual Display

Fetch, parse, and render feed items as styled cards within the ExB app. Items are rendered using a user-authored Markdown template with token substitution, giving full control over layout and content without writing code. Status values drive background color coding via a configurable map.

Mode B — Spatial Data Source (optional)

When a matching ArcGIS feature service is configured, the widget performs a runtime join between feed items and feature service records using a shared ID field (e.g. `clid / ClosureID`). This produces an ExB data source that exposes feed items with geometry, enabling downstream map widget interaction — including zoom-to-feature and selection synchronization — through ExB's standard message/action framework.

Note: Mode B requires the feed data and a corresponding feature service to share a common ID. This is confirmed for the King County road closures use case. Other feeds may support Mode A only.

2. Reference Feed — King County Road Closures

The primary reference feed for this specification is the King County My Commute road closures XML feed. This is a custom XML schema (not RSS 2.0), which is typical of government transportation data systems built in the mid-2000s to early 2010s. Its structure is representative of a large class of similar legacy feeds that remain in active production use.

Feed Endpoint

<https://gismaps.kingcounty.gov/mycommute/rssEmbed.aspx>

XML Schema

```
<roadclosures>
  <item>
    <clid>3204</clid>
    <status>Closed</status>
    <location>SE Green Valley Rd from 166th Ln SE to #16812</location>
    <description>Tree down</description>
    <region>Southeast</region>
    <community>Auburn</community>
    <maintDiv>4</maintDiv>
    <effectiveDate>3/12/2026 2:09:00 AM</effectiveDate>
    <changeDate>3/12/2026 4:05:00 AM</changeDate>
  </item>
</roadclosures>
```

Notable characteristics of this feed:

- No spatial data. Geometry comes entirely from the companion feature service.
- Status is a clean discrete field with four values: Closed, Restricted, Upcoming, Open.
- Description is free-text prose, frequently containing plain-text URLs.
- Datetime strings are locale-formatted (M/D/YYYY h:mm:ss AM/PM), not ISO 8601.
- The root element is <roadclosures>, not <rss> or <channel>.

Companion Feature Service

A matching polyline feature layer is available at the King County ArcGIS Server instance:

```
https://gismaps.kingcounty.gov/arcgis/rest/services/RoadAlerts/
KingCo_Road_Alerts/MapServer/0
```

This layer's ClosureID field corresponds directly to the <clid> element in the feed. The feature layer carries road segment geometry in Web Mercator (WKID 102100). This join enables Mode B spatial integration.

The My Commute map also supports deep-linking to a specific closure via URL parameter:

```
https://gismaps.kingcounty.gov/MyCommute/default.aspx?clid=3204
```

Note: This URL pattern can be used as a fallback zoom mechanism if the ExB message/action approach is not viable in a given deployment.

3. Template Token System

The widget uses a Markdown template with double-brace tokens to control how each feed item is rendered. Tokens are substituted before the Markdown is processed by the MD-to-HTML

renderer. This approach keeps the rendering logic in familiar territory and allows non-developers to customize display.

3.1 Field Tokens

The following tokens map directly to XML elements in the King County feed. For other feeds, the token names are dynamically generated from the XML element names present in the first fetched item.

XML Element	Template Token	Notes
<clid>	{{clid}}	Unique closure ID. Join key to feature service.
<status>	{{status}}	Closed / Restricted / Upcoming / Open. Drives color mapping.
<location>	{{location}}	Road name and limits description.
<description>	{{description}}	Full narrative. May contain plain-text URLs.
<region>	{{region}}	Geographic region (Northeast, Southeast, etc.).
<community>	{{community}}	City or community name.
<maintDiv>	{{maintDiv}}	Maintenance division number (1–6).
<effectiveDate>	{{effectiveDate}}	When the closure became active. RFC-style datetime string.
<changeDate>	{{changeDate}}	Last modified datetime.

3.2 Filter Tokens (Derived)

Filters transform a token's value before rendering. They are separated from the token name by a pipe character and accept an optional argument.

Token + Filter	Behavior
{{effectiveDate "MMM D, YYYY"}}	Formatted date using a format string.
{{changeDate "MMM D, YYYY h:mm A"}}	Formatted datetime.
{{description autolink}}	Auto-converts plain-text URLs in the description to <a> tags.
{{clid mapLink}}	Renders a "View on Map" hyperlink using the My Commute URL pattern.

3.3 Example Template

The following template produces a card similar to the existing King County My Commute embed:

```
### {{location}}

**Status:** {{status}} | **Region:** {{region}} | **Community:**
{{community}}

{{description | autolink}}

*Effective: {{effectiveDate | "MMM D, YYYY"}}* . {{clid | mapLink}}
```

Note: The status value is also used independently for color coding the card background. It does not need to appear in the template for color coding to work.

4. Settings Panel

The widget configuration panel in the ExB builder exposes the following settings. All settings persist to the widget's config JSON.

Setting	Control Type	Description
Feed URL	Text input	URL of the XML feed endpoint.
Root item element	Text input	XML element name for each item. Default: "item".
Refresh interval	Number (minutes)	How often to re-fetch the feed. Default: 5 min.
Card template	Markdown editor	MD template using {{token}} syntax. Live preview pane alongside.
Status field	Dropdown (field name)	Which XML field drives status-based color coding.
Status color map	Key-value pairs	Map status values to background colors. Defaults pre-populated.
Date format string	Text input	Global date format applied to date token filters. Default: "MMM D, YYYY".
Feature service URL	Text input (optional)	REST endpoint of a matching feature layer for spatial join.
Join field (service)	Text input	Field name on the feature service to join against. Default: "ClosureID".
Join field (feed)	Text input	XML element name on the feed to join with. Default: "clid".

Setting	Control Type	Description
Max items	Number	Limit rendered items. 0 = show all.
Filter by status	Multi-select checkboxes	Optionally hide items by status value.

4.1 Status Color Defaults

The following default color map is pre-populated when a status field is configured. All values are user-overridable.

Status Value	Default Color (Hex)	Semantic Label
Closed	F4CCCC	Red-tinted (danger)
Restricted	FFF2CC	Yellow-tinted (caution)
Upcoming	CFE2F3	Blue-tinted (informational)
Open	D9EAD3	Green-tinted (clear)

5. Widget Architecture

The widget is built using the ExB Developer Edition SDK (React + TypeScript + Jimu framework). It consists of the following primary files:

- widget.tsx — Runtime component. Handles fetch, parse, token substitution, and render.
- setting.tsx — Builder-time configuration panel. Exposes all settings from Section 4.
- config.ts — TypeScript interface for the widget config object.
- manifest.json — Widget metadata and dependency declarations.
- utils/xmlParser.ts — XML fetch and parse logic. Returns typed item array.
- utils/tokenRenderer.ts — Token substitution and filter pipeline. Calls MD renderer.
- utils/featureJoin.ts — (Mode B) Queries feature service by ID list, returns geometry map.
- utils/dataSourceBuilder.ts — (Mode B) Constructs jimmu DataSource from joined records.

5.1 Runtime Data Flow — Mode A

1. Widget mounts. Reads config (feed URL, template, color map, refresh interval).
2. Fetches XML feed via browser fetch API.
3. Parses XML, extracts items using configured root element name.
4. For each item: substitutes tokens into MD template, applies filters, passes to MD renderer.
5. Renders resulting HTML into card components with status-driven background color.

6. Sets interval timer for refresh. On refresh, re-executes steps 2-6.

5.2 Runtime Data Flow — Mode B (adds to Mode A)

7. After XML parse, collects all clid values from items.
8. Queries feature service: WHERE ClosureID IN (id1, id2, ...).
9. Builds geometry map: clid → polyline geometry.
10. Constructs jimu DataSource records, each combining XML fields + geometry.
11. Registers data source with ExB runtime.
12. List item click triggers DataSource record selection, which ExB propagates to connected Map widget via standard message/action.

5.3 CORS Considerations

Government XML feeds are frequently served without CORS headers, which will block browser fetch requests. The following mitigation options should be evaluated in order of preference:

- Confirm the target feed allows cross-origin requests (test in browser console first).
- If CORS is blocked, implement a lightweight server-side proxy as part of the Dev Edition deployment.
- As a last resort, use a CORS proxy service for development/testing only — not for production.

Note: The King County feed URL endpoint should be tested for CORS headers before development begins.

6. Markdown Renderer Integration

The widget builds on an existing Markdown-to-HTML renderer developed in a prior project. The integration requires the following:

- The renderer must be importable as a CommonJS or ESM module within the ExB Dev Edition webpack build.
 - Token substitution runs as a pre-pass before the MD string is sent to the renderer, so the renderer receives clean Markdown with no unresolved tokens.
 - Raw HTML passthrough behavior must be confirmed. The King County description field is plain text, but other feeds may supply HTML blobs in description fields. A config toggle — "treat description as raw HTML" vs. "treat as plain text" — should be exposed.
 - The renderer should be invoked per-item, not globally, so each card is independently sanitized.
-

7. ExB Integration Points

7.1 Message Actions (Mode B)

When Mode B is active and a data source is registered, the widget should publish the following message when a list item is clicked:

- Message type: DATA_RECORDS_SELECTION_CHANGE
- Payload: selected DataRecord (contains ClosureID, geometry, and all XML fields as attributes)

A connected Map widget can subscribe to this message and respond with zoom-to-extent or highlight behavior using standard ExB message/action configuration — no custom code required on the Map widget side.

7.2 Widget Output Data Source (Mode B)

The widget registers a custom data source that other ExB widgets (Table, List, Feature Info) can connect to. This enables the feed data to participate in the full ExB data ecosystem — filtering, selection, and cross-widget synchronization.

7.3 No-Code Fallback (Mode A only)

When Mode B is not configured, clicking a list item can optionally open the King County My Commute deep-link URL in a new browser tab using the `clid | mapLink` token filter. This requires no ExB integration plumbing and works in AGOL-deployed apps as well.

8. Open Questions

The following items require resolution before or during development:

Topic	Detail
CORS on feed endpoint	The King County feed may not include CORS headers, requiring a proxy or server-side fetch. Determine if a lightweight proxy is needed for the dev environment.
Feature service accessibility	Confirm the KingCo_Road_Alerts MapServer/0 endpoint is publicly accessible without authentication.
Map widget integration mechanism	Decide whether list item selection drives the map via ExB message/action framework or via direct URL deep-link (e.g. MyCommute?clid=X).
MD renderer dependency	Determine whether the existing MD-to-HTML renderer is a standalone module that can be imported as a dependency, or if it needs to be inlined.
ArcGIS Online vs.	Custom widgets require Dev Edition deployed to a web server for AGOL, or

Topic	Detail
Enterprise target	Enterprise 11.x registration. Confirm target deployment environment.
Feed format generality	Decide whether the widget strictly supports standard RSS 2.0 vs. arbitrary XML vs. both, as this affects the settings panel complexity.

9. Out of Scope (v1)

- Atom feed support (different envelope schema from RSS 2.0).
- JSON feed support (separate parsing path warranted in a future version).
- Write-back or editing of feed data.
- Offline/cached feed storage across sessions.
- Arcade expression support for styling (blocked by ExB's FeatureLayer-only Arcade requirement; not applicable to XML feed data).
- Geocoding-based spatial join (adds latency and API cost; clid/ClosureID join is preferred).

Status: Draft — subject to revision as development begins.