

Dear Valued Partners,

We are pleased to announce that EasyBuilderX (EBX) V1.01 is being officially released on July 22, 2026.

Key Feature Highlights

1. [Device] Dynamic Property Modification, Events, and Methods
 - Dynamic Property Modification: An **Enabled** property has been added, allowing devices to be enabled or disabled during runtime. Certain connection parameters (e.g., IP Address, Port) now support dynamic modification at runtime.
 - New State and Error Code Properties: A Connection **State** property has been introduced, along with an **Error Code** property to facilitate real-time diagnostics.
 - Device Event Supported: Four event notifications are now provided.
 - Connection status changed
 - Connected
 - Disconnected
 - Fault occurred

These events can be used to trigger **Embedded Window** widgets or other tasks.

- Device Method Supported: Two methods are now available.
 - Update interface config — Applies modified connection parameter settings.
 - Clear fault — Manually clears a specific communication fault and re-establishes the connection.

2. [Variable Library] Bitstring Data Type Supported

A new Bitstring data types have been added, including Byte, Word, DWord, and LWord. Users can use the Bitstring data type to expand a single word-length variable into multiple independent bits, enabling external access to individual bits at runtime.

3. [String Table] Feasibly binding to Properties and String Variables

- Expanded Binding Types: Previously, the String Table did not support binding to string-type variables or properties. This update officially enables binding to String /

WString variables, as well as direct binding to system object Properties (e.g., Hardware Key).

- Improved Design Flexibility and Performance: Previously, displaying dynamic text such as "Temperature = 35 °C" required concatenating multiple text widgets. With the dynamic string binding functionality, this can now be achieved with a single text object.
- Screen-Scoped Property Binding: Strings can now insert and display properties of other widgets within the same screen (e.g., binding a string to the **Value** property of a Numeric widget on the same Window, reflecting input value changes in real time at runtime).

4. [String Table] Cross-String Table Binding

- Cross-String Table Binding: Strings can now be directly bound to strings in another String Table.
- Dynamic String Index: At runtime, a variable ID can be used to dynamically reference different strings within a String Table.

5. Inline Condition Expression

- For Boolean (BOOL) Input Properties: Users can now write condition expressions directly in the property panel, eliminating the need to switch to the Condition Library to create them.
- For Action Configuration (e.g., If / While / Wait control flow): Users can now write condition expressions directly in the Action field — The previous version restricted If / While / Wait control flows to only True or False conditions.
- Condition Scope Introduction: If a condition expression needs to be shared across widgets, the inline condition expression can be promoted to a standalone condition using the **Promote to Standalone** function, making it globally visible. The Condition Library now introduces three scope levels:
 - Application (e.g., variables)
 - Front-end (e.g., language settings)
 - View Scope (e.g., Screen / Window)

The scope hierarchy enforces that narrower scopes may depend on broader scopes, but broader scopes must not depend on narrower scopes.

6. UDT Member Binding — Individual Binding functionality for Structured Data Type Members

When a widget's property is of a Structured Data Type (UDT), it is now possible to individually bind each sub data member within the structure.

Binding Rules

- **Previous Limitation:** The entire structure had to be bound as a whole to a single structure variable; sub data members could not be independently bound to separate variables.
- **Individual Binding:** Users can now expand the UDT-type property and individually bind different sub data members (e.g., "Temperature", "Status") to separate variables.

Operation Guide

- I. In the property panel, select a property along with a bind button and click the bind button with the left mouse button.
- I. If the property's data type is a Structured Data Type, the default variable menu only allows binding to variables of the same structure type.
- II. Click the bind button again — the variable menu will display the structure variable's sub data members, allowing individual sub data member binding.

7. [Arithmetic Variable] Editor UX Improvements

Editor UI/UX Improvements: To precisely identify widgets, variable and property paths now include prefix identifiers (e.g., Variable: , Property:) for quick differentiation.

Error Prevention and Protection Mechanisms:

- **Invalid Reference Warning:** If a referenced variable is deleted, the Arithmetic Variable editor immediately highlights it in red (indicating an Invalid state).
- **Circular Reference Detection:** Prevents infinite loops caused by mutual variable references (Circular Reference).

8. [Screen/Window] Copy Shortcut Operation Supported

A new Ctrl + Drag shortcut has been added to the window design canvas for duplicating objects via mouse drag.

Operation Guide

- I. Select one or more objects, then hold the Ctrl key.
- II. Drag the selected objects to the target position using the left mouse button.
- III. Release the mouse button — a new object with identical properties and configuration will be created at that position.

9. [Multi-Copy] Function

The Screen/Window canvas now supports matrix-style batch duplication and arrangement of a single widget, with automatic address offset.

Operation Guide

Automatic Address Offset: When the address offset option is checked, the system detects the variable bound to the widget (e.g., Tag_0) and automatically increases or decreases the variable index for each duplicated instance (e.g., Tag_1, Tag_2, Tag_3, etc.), eliminating the need for manual editing.

10. Widget's Text Property — Editing UX Improvements

The multi-language editing workflow for widget text properties has been optimized. Users can now edit text content and translations directly in the property panel without leaving the Screen/Window canvas, significantly reducing the number of times switching to the String Library.

Key Improvements

- **Inline Editing Window:** An "Edit" button has been added to the right side of the Text property field. Clicking it opens a dedicated sub-window.
- **Multi-Language Input fields with Auto-Focus:** The dedicated sub window displays input fields for all enabled languages simultaneously, with focus automatically locked to the input field upon opening.

11. EBX Project and IDE Default Font — Noto Sans

The default multi-language font for the EBX software interface and projects has been unified to the open-source font Noto Sans. Noto Sans covers character sets for all major global writing systems.

12. Widget's Text Property — Dynamic String and Font Fallback Mechanism

Widget text properties now support binding to dynamic string variables, with a robust Font Fallback mechanism.

Font Application and Fallback Order

When a widget renders text, the system resolves and applies fonts in the following priority order:

- **Widget Font:** The specific font selected by the user in the widget's property panel takes highest priority.

- Language Font Profile: If the Widget Font cannot render a specific character, the system falls back to the font assigned by the project's language settings.
- System Default Font: If neither of the above can render the character, the system ultimately falls back to the OS built-in Noto Sans font, ensuring text is never rendered as blank.

13. Picture Library Improvements

Image Name Editing and Display Supported:

- Double-Click to Rename: In the Picture Library, double-clicking an image name enters edit mode for renaming.
- Hover Tooltip: Hovering the cursor over a thumbnail in the Picture Library or Widget's Picture property panel displays a tooltip showing the image's full name.

14. [Log] Panel Compilation Message Error Filtering and Jump-to-Source Navigation

- "Errors Only" filter: In the compilation message window, enabling this option filter to display only Error messages.
- Jump-to-Source Navigation: Double-clicking any error message causes the System to automatically navigate to the target window that caused the compilation error, with focus locked on the relevant widget's configuration field.

15. Standardized Error Codes

The EBX error code reference table has been reorganized and standardized. Error code details can be found in the Property Information section of the property panel.

16. [Recipe] Functional Changes

- Recipe Record Name Mechanism:

Functional changes to the Recipe Database: Previously, the system relied entirely on ID/Index as the identifier when saving and modifying recipe records. This made it difficult to track recipe modification history through the Audit Log when intermediate records were deleted or new records were re-added. This update changes the identification mechanism to **Record Name**, using a String as the recipe's unique primary key. Each recipe record name must be unique within the same recipe data table and cannot be duplicate.

- Auto-Naming: When creating a new Recipe Record in the recipe record panel (specifically, **Element** section) without manually specifying a record name, the

system automatically generates a unique record name based on the built-in naming rule.

- Recipe Method Behavior Changes: addRecord and addRecordFromDevice
 - Adjusted addRecord behavior: Now only adds a single recipe record with each field automatically populated with the minimum value of its defined range.
 - New addRecordFromDevice: Reads the current values of the recipe's "Bind to" variables and creates a new recipe record based on the retrieved data.

17. [Audit Trail]

The Audit Trail provides a comprehensive system and user activity tracking mechanism to meet regulatory requirements for GMP or FDA laws. It records complete audit information including Who, When, Where, Value Before Modification, Value After Modification, and Reason for Change (Comment required), with support for Confirmation and Electronic Signature (E-Signature) functions.

- Confirmation Level Settings
 - Log Only: Records silently in the background without interrupting the user's workflow.
 - Confirmation Required: A confirmation dialog appears during the operation. The user must enter a **Comment** to complete the modification.
 - Electronic Signature: The highest security level. An authentication dialog appears during the operation. The user must re-enter their credentials and provide a **Comment**. The modification or operation is only completed after successful verification.

- Applicable Scope

Applicable to the following scopes:

- User Administration: Automatic recording.
- Audit Trail: Automatic recording.
- Runtime: Automatic recording.
- Variable: User-configurable.
- Recipe: User-configurable.
- Operation Action: User-configurable. Audit Trail steps can be inserted into specific Action sequences. If E-Signature verification fails during an Action flow, all subsequent actions are immediately aborted.

18. [Audit Trail Viewer]

The Audit Trail Viewer widget is used to browse audit trail history records directly on the HMI screen. The displayed column configuration includes Record ID, Date/Time, User, Operator Station, Object, Description, and Comment.

19. [Backup Object]

The Backup widget supports CSV universal format output, time range calculation, and file tamper-proof checksum verification.

Configuration Details

- **Storage Path Restriction:** Backup storage to the HMI root directory is not supported. The only output path is directed to a designated external storage device.
- **Data Source:**
 - Data Sampling
 - Alarm
 - Audit Log
- **Range Mode:**
 - **Relative Mode:** Automatically rolls back from the trigger time. For example: roll back 8 hours or 24 hours. Supports Anchor alignment to the nearest minute, hour, or day. For example, if triggered at 10:37 with alignment set to "hour" and a 16-hour lookback, the system calculates the data range starting from 10:00 going back 16 hours.
 - **Interval Mode:** Specifies a start time and end time.
- **File Naming and Splitting**
 - **File Naming:** Supports embedded time format codes and string variables.
 - **File Splitting:** Supports setting a maximum record count per file. When the limit exceeds, the CSV file is automatically split into multiple files.
- **Execution Mode:**

Two method calls are provided for executing backups:

- **Sync:** The backup operation blocks subsequent steps until the backup data is fully written to the file, then continues to the next Action.
- **Async:** This mode triggers the backup and immediately proceeds to the next Action without being held for the backup process to complete.

- **Status** and **Progress** Properties:
 - Status: Provides three backup state values: Idle, Running, and Failed.
 - Progress: Provides percentage progress information from 0–100.

20. Date/Time and Duration Time

- New Time-Related Data Types

The Variable Library now supports IEC 61131-3 time-based data types, including TIME, LTIME, DATE_AND_TIME, and LDATE_AND_TIME. The Backup widget's range type property requires these data types.

- New [Date/Time] and [Duration Time] Widgets

Users can use the Date/Time and Duration Time widgets as time or duration pickers on the HMI screen at runtime to populate time parameters into the Backup widget's range type properties, completing the time parameter setup.

21. System Object

A new System Object interface has been added, providing a unified system object management interface to facilitate future expansion of all system-level parameter configuration features in EBX.