

Beckhoff Training - Key Terms

TwinCAT HMI Key Terms and Technologies

Beckhoff Training - Key Terms

Table of Contents

1	TwinCAT HMI Key Terms.....	3
1.a	HMI Creator - The graphical interface	3
1.b	HMI Framework - The framework handling resource communication between client and server	3
1.c	HMI Server	4
2	The foundational technologies of TwinCAT HMI.....	5
2.a	Web Languages and Technologies	5
2.b	Server Languages and Technologies	6
2.c	TwinCAT HMI Framework	7
2.d	TwinCAT HMI Server	7

TwinCAT HMI Key Terms and Technologies

1 TwinCAT HMI Key Terms

(InfoSys Links + quick reference definition)

1.a HMI Creator - The graphical interface

- **Designer** - A graphical editor for the HMI. Controls from the toolbox are dragged into here for creating the HMI. The graphical interface is either for the follow three types -
 - **View** - The main host for controls. Controls, Content, and User Controls are placed in a View. Views cannot go in other hosts
 - **Content** - A mid level host for controls. All non-Views can be placed in a Content, which then has to be placed in a Region, which is then placed in a View
 - **User Control** - A low level host for controls. User Controls can only host Controls or other User Controls, which are then placed in Contents or Views
- **Toolbox** - The window that hosts all the HMI controls than can be dragged onto the Designer
 - **Controls** - A control that can be acted upon by a user or PLC runtime and in turn can act upon other controls or PLC runtimes
 - **Properties** - The Visual Studio properties window will display a set of properties for the control - Control Attributes, Events, and Permissions
 - **Attributes** - Specific control attributes that describe the logic and render on a control - Color, Text, etc...
 - **Data Binding** - Binds an attribute to a specific Server, Internal, or Control mapped Symbol
 - **Function Binding** - Allows JavaScript functions and TcHmi Functions to directly control the attribute - based on logic and/or other called attributes
 - **Events** - Events the control listens for to then perform an action
 - **Actions and Conditions** - Actions and Conditions the control performs when an Event triggers
 - **Permissions** - Control access based on the logged in user's permissions
 - Operate - being able to interact with a Control in the client
 - Observe - being able to view a control's properties in the client
- **HMI Configuration** - A top view of all the HMI Framework
 - **Symbols** - Points of data associated with a server, internal, or control mapped attribute that can be linked to other control, framework, or server attributes and be acted upon
 - **Data Types** - Data types the HMI handles - PLC Data Types auto convert to HMI when bound - [General](#) / [Framework](#) / [Server](#)
 - **Localization** - The locale editor for handling HMI international support

1.b HMI Framework - The framework handling resource communication between client and server

- **TcHmi API** - The main API for Framework communication
- **Client** - Any supported HTML5 browser
- **Functions** - Default or custom created functions added to the TcHmi API
- **CodeBehind** - A global library of functions or event handlers
- **Framework Control** - A custom control made by a user from a code template

TwinCAT HMI Key Terms and Technologies

1.c HMI Server

- [Server Extension](#) - Server side extensions to the HMI server functionality. ADS Communication for example is enabled through the ADS Server Extension
 - [Configuration](#) - A configurable set of server extension parameters that can be saved as a preset for a publish to a stand alone server
 - [Stand Alone Server](#) - A separate executable server that is launched on a system that would then be the host for all clients to load from. Like the XAR for TwinCAT, but for TcHmi
-

TwinCAT HMI Key Terms and Technologies

2 The foundational technologies of TwinCAT HMI

TwinCAT HMI introduces a suite of web technologies new to TwinCAT 3's engineering platform. These technologies are vast libraries enabling rich media applications. Our TwinCAT HMI Framework utilizes the following to create a human machine interface within our TwinCAT software system:

2.a Web Languages and Technologies

(External resources and definitions - W3Schools, MDN, MSDN)

HTML5 - Hypertext markup language - 5th iteration: Rich media support

- The main web technology in rendering our HMI in the client.
- TwinCAT HMI - Code View, Framework Extensions
- [W3Schools](#)
- [MDN](#)

CSS - Cascading Style Sheets: Allows styling of HTML elements

- TwinCAT HMI - Themes, Framework Extensions
- Used to style all our controls
- [W3Schools](#)
- [MDN](#)

JS - Javascript - An implementation of ECMA-262. The basic scripting language for interacting with the interface

- TwinCAT HMI - Function Editor, CodeBehind, Framework Extensions
- Used to perform the main logic and communication in all our controls
- [W3Schools](#)
- [MDN](#)

WS - WebSocket - A communications protocol, providing full-duplex communication channels over a single [TCP](#) connection

- TwinCAT HMI - The protocol used to communicate data between the HMI Server and Client
- All websocket frames are viewable within the client allowing the user to read all traffic
- Supports WSS, WebSockets Secure for encrypted traffic

[Cheat Sheet](#) - An all in one cheat sheet covering basic references for HTML5, CSS, JS, and JQ (jQuery).

Click to show additional web technologies used...

Canvas - HTML5 graphic renderer capable of drawing graphs instructed with JavaScript.

- Used to render our TcHmi Graphs and Charts.
- [W3Schools](#)
- [MDN](#)

TwinCAT HMI Key Terms and Technologies

SVG - Scalable Vector Graphics - Current SVG 2.0: Wider support for modern web technology iterations. Ideal for web and uses less storage space

- Our main graphic type in our included Media Gallery
- [W3Schools](#)
- [MDN](#)

JSON - Javascript Object Notation based on the ECMA-262 standard.

- Used to structure our controls as JavaScript Objects with properties
- [W3Schools](#)
- [MDN](#)

JSON Schema - Annotates and validates JSON documents. We use [draft 04](#).

- Used to parse our controls into data structures for communication between other controls and our server

AJAX - Asynchronous JavaScript And XML: Not a language, a communication technique

- Used for our asynchronous web server communication and dynamic HTML communication
- [W3Schools](#)
- [MDN](#)

TypeScript - A superset of Javascript, adds data typing support

- Used to define our data structures as different types
- [TypeScript Types and Custom Types](#)

2.b Server Languages and Technologies

.NET (Framework) - "The .NET Framework is a development platform for building apps for web, Windows, Windows Phone, Windows Server, and Microsoft Azure. It consists of the common language runtime (CLR) and the .NET Framework class library, which includes a broad range of functionality and support for many industry standards."

- TwinCAT HMI runs on a .NET server install on your chosen system which allows the communication and hosts the HMI resources
- TcHmi Server, ADS extension, Event Logger extension...
- See our [system requirements](#) page for the latest .NET or .NET Core SDK required
- [MSDN](#)

.NET Core - ".NET Core is a general purpose development platform maintained by Microsoft and the .NET community on [GitHub](#). It is cross-platform, supporting Windows, macOS and Linux, and can be used in device, cloud, and embedded/IoT scenarios."

- .NET Core expands upon the .NET platform allowing more functionality
- [MSDN](#)

TwinCAT HMI Key Terms and Technologies

C# - The language used with .NET and .NET Core

- The main language for TwinCAT HMI Server Extensions
- [MSDN](#)

Additional server technologies used...

[Lua](#) - Designed primarily for embedded use in host applications

[SQLite](#) - An embedded SQL database engine

[OpenSSL](#) - A software library for applications that secure communications over computer networks

[RapidJSON](#) - JSON parser and generator for C++

[WSAPI](#) - An API that abstracts the web server from Lua web applications

2.c TwinCAT HMI Framework

Our HMI Framework comes with the following JavaScript libraries -

- [TcHmi.js](#) - Beckhoff Library
 - Our main TcHmi Framework library. [InfoSys HMI API](#) - Navigate in the left side panel
- [Jquery.js](#) - 3rd Party
 - Extends basic JavaScript and HTML interactions

2.d TwinCAT HMI Server

Our HMI Server Framework comes with the following extensions - Some of these extensions do not have official documentation yet

- [ADS](#) - Our extension that handles communication between PLC runtimes and the HMI Framework. Will convert PLC datatypes to HMI useable datatypes
- TcHmiEventLogger - Handles communication between the main TwinCAT Event Logger library, and the HMI Framework
- TcHmiLua - Server extension for handling the TcHmiSrv Configuration page
- TcHmiSqliteHistorize - Handles the historize symbol functionality for trending symbols
- TcHmiSqliteLogger - The event logger handler for TcHmi
- [TcHmiSrv](#) - Our main extension that loads all other extensions, registering their domains and communications on launch
- TcHmiTextStorage - Stores sever settings within a small database
- TcHmiUserManagement - Handles user permissions with user profiles belonging to predefined groups