Trimble

Tekla Structures Open API

Agenda

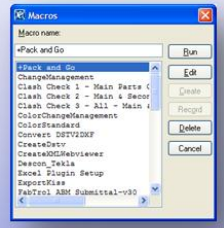
- What is the API
- Key areas of the API
- Examples of tools utilizing the key areas
- How can you utilize the API
- Strategy to start
- What are hardware and software needs

What is the Tekla API?

What is the Tekla Open API?

- API = Application Programming Interface.
- Customization of Tekla Structures without having to change the core.
- Utilizing industry standard Microsoft .NET technology.
- Creates a competitive advantage at lower cost.

What is the Tekla Open API?



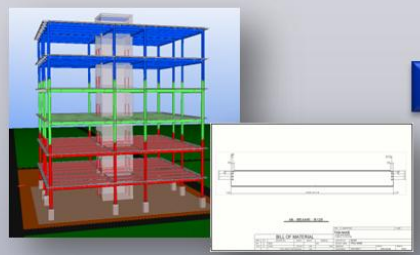
Macros

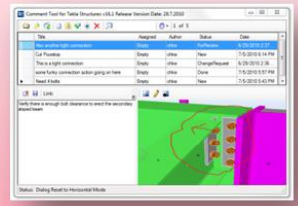
(*.cs)

Recorded scripts calling commands in the Tekla Structures interface for Model and Drawings. (C#, API Library can be used here)

API Library

Create, Modify, Delete, Extract Information, Set User Attributes, Navigate, Call Macros.

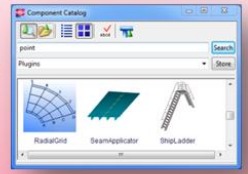


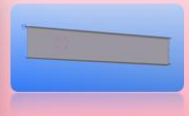


External Applications

(*.exe)

Tools that connect and interact with Tekla Structures





Plugins

(*.dll)

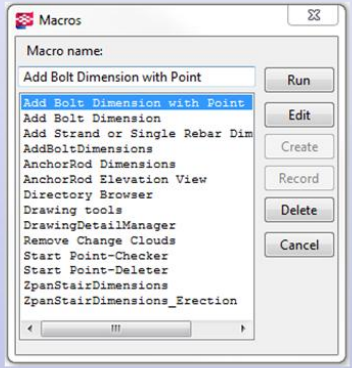
Parametric, run inside Tekla Structures, fast.

Where is the Tekla Open API?

- Included inside your Tekla Structures installation

Macros

Tools>Macros



XS_MACRO_DIRECTORY
C:\ProgramData\Tekla Structures*%ver%*\environments\usimp\us_common\macros**modeling** or **drawings**

API Library

C:\Program Files\Tekla Structures*%ver%*\nt\bin\plugins

- Tekla.Structures.dll
- Tekla.Structures.Model.dll
- Tekla.Structures.Drawing.dll

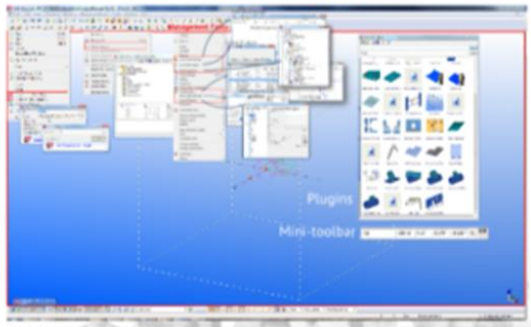
- Tekla.Structures.Dialog
- Tekla.Structures.Datatype
- Tekla.Structures.Plugins
- Tekla.Structures.Catalogs
- Tekla.Structures.Analysis

Who is using the Tekla Open API?

> Tekla Developers

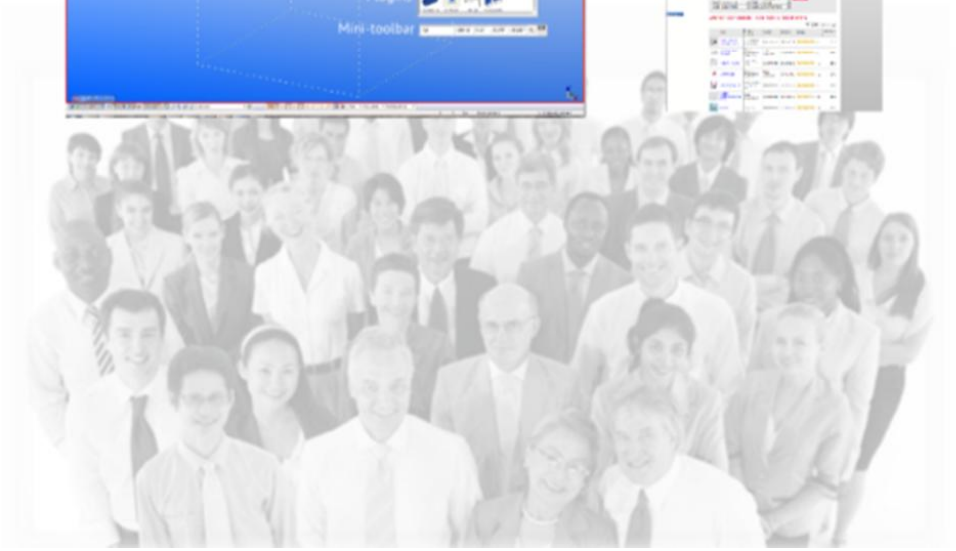
- More features are made by Open API
- Extensions
- Pugins

> Tekla Structures Out-of-box features



> Extranet Extensions

- Hundreds of extensions
- Thousands of downloads



> Partners

> Trimble Integration Solutions

Trimble, Meridian Systems, Accubid, QuickPen, ThingMagic, ICT

> Engineering & Construction Services

HNTB, BlueScope Steel, Thornton Tomasetti, Candraft, Walter P Moore

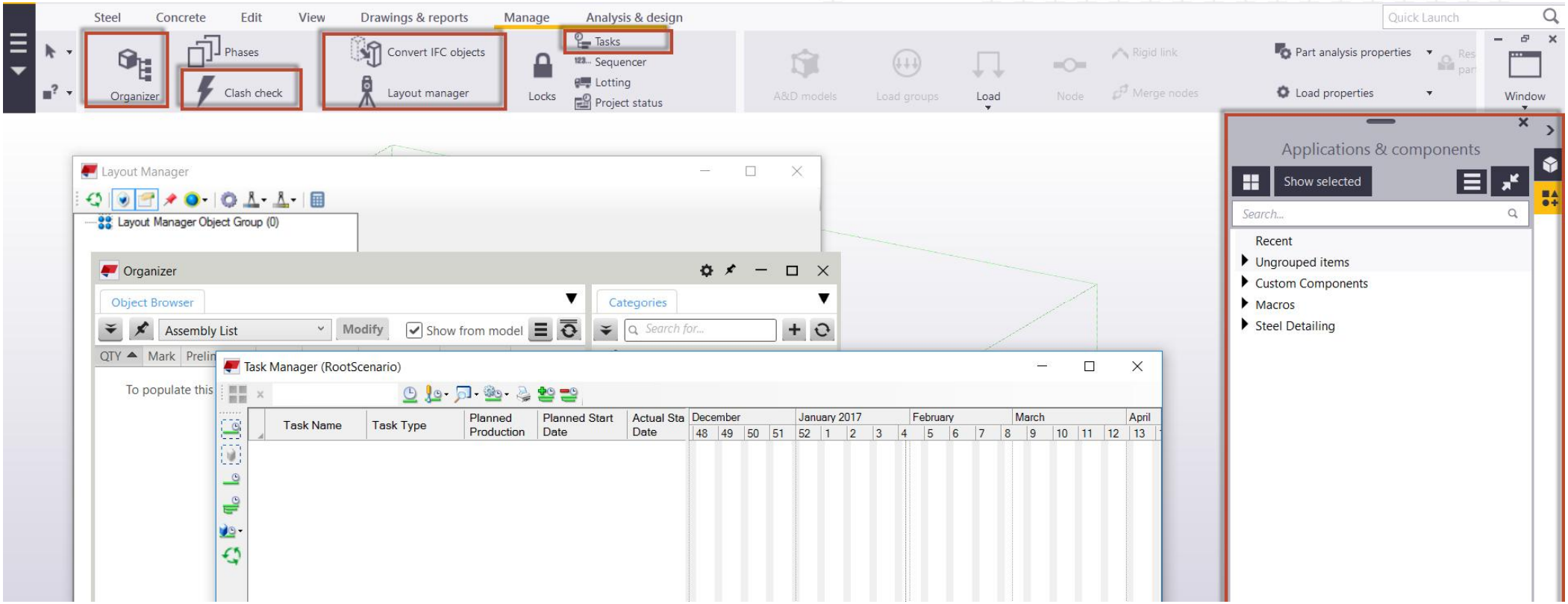
> Building Product Partners

CONXTECH, TECHNYX, New Millennium, Vulcraft

> Production & Automation Partners

FabTrol MRP, BETSY, HGG, FICEP, BVBS, construsteel, Steel Projects

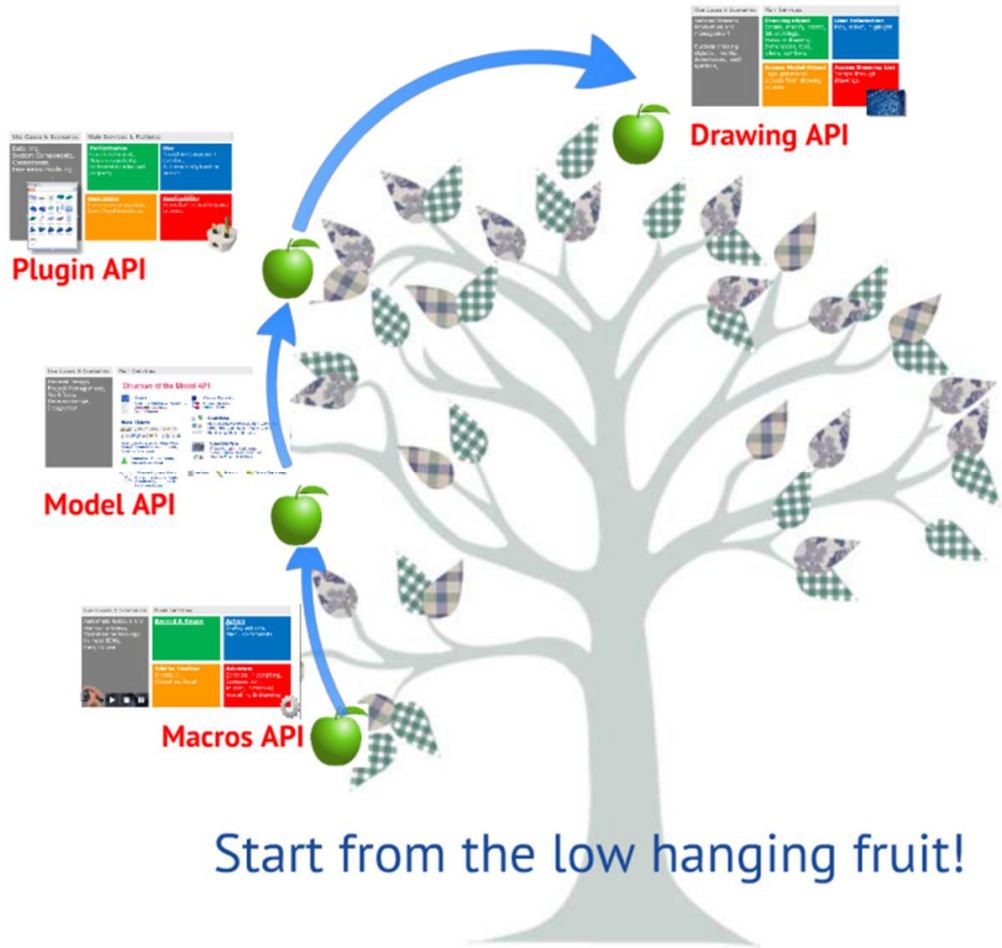
Tekla Headquarters API Usage



Tekla USA API Usage

- Auto Task Suite
- Detail Manager
- Construction Sequencer
- Bar Checker
- Layout Point Applicator
- Concrete Tools
- Reference Model Importer
- Curved Stanchions
- Hyperlink Manager
- Workflow Manager
- Find Number
- Label Tool
- Handrail Dimensioning
- Beam Marks
- Cover Dimension
- Component Variables
- Drawing Break Line
- Drawing Match Line
- Solid2Plates
- Compare Objects
- Copy Uda
- Uda List
- Anchor Rod tools
- Base Plate Applicator
- Combine Part Marks
- Combine Rebar Marks
- Crane Console
- Data Exchanger
- Plan Level Marks

What is in Tekla Open API?



Tekla Open API

- **Macro API**

  Record and play commands and dialog actions

- **Model API**

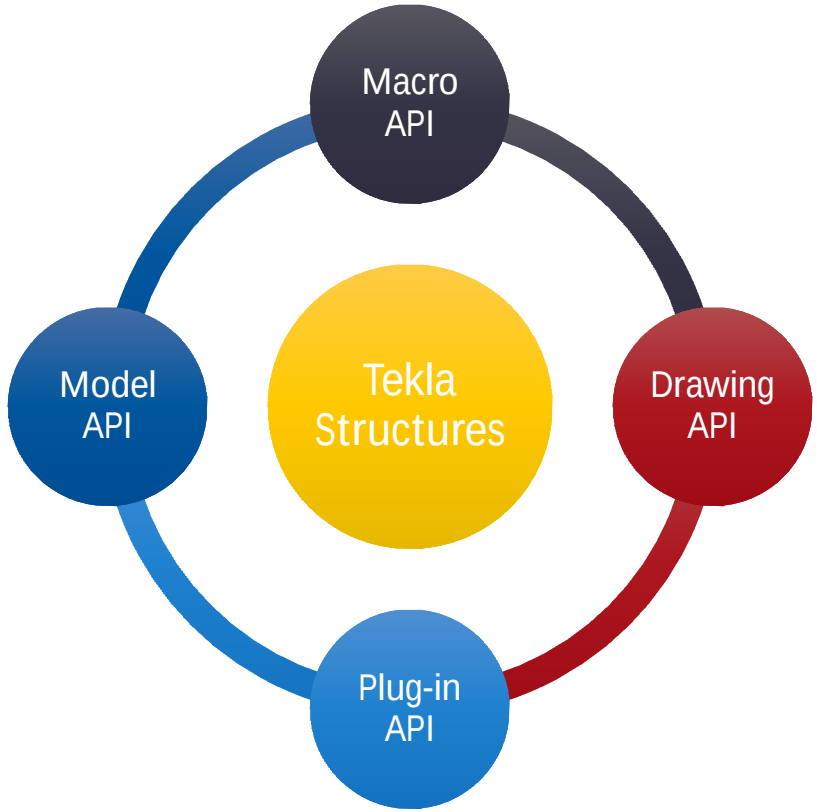
  Create and modify model objects and info

- **Drawing API**

  Create and modify drawings and drawing objects

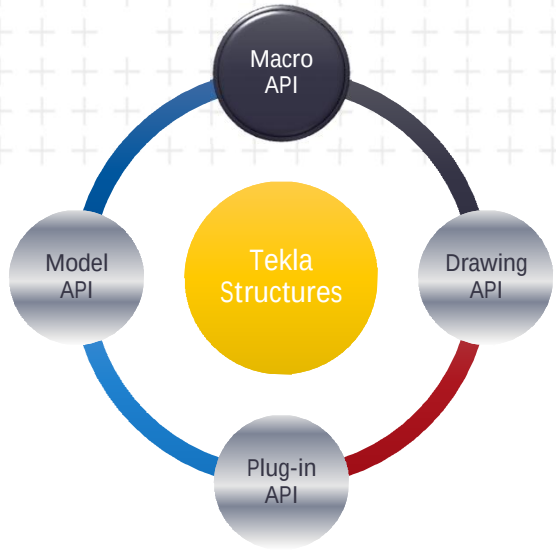
- **Plug-in API**

  Create intelligent objects in the model




Use Cases & Scenarios
Main Services
Automate tedious and manual process, Common technology in most BIMs, Easy to use
Record & Reuse
Actors
Dialog actions, Menu commands
Add to Toolbar
Shortcut, Global vs. local
Advance
Editable in scripting, Composition, In conjunction w/ modeling & drawing
Macros API

Tekla Open API - Macros



- Macro API
  - Record menu commands and dialog actions
  - Playback recorded macros
  - Edit and enhance recorded macros (C#)
    - Combine with model and drawing API
  - Add macros to user menus, toolbars, shortcuts
    (modeling & drawing editor macros are separate)

This is what the Model Macro Looks Like

- Record a macro from the Advanced Features option in the Applications & components catalog.
- Select the macro, right-click, select Edit in the context menu.

```
// Generated by Tekla.Technology.Akit.ScriptBuilder

namespace Tekla.Technology.Akit.UserScript
{
    public class Script
    {
        public static void Run(Tekla.Technology.Akit.IScript akit)
        {
            akit.Callback("acmdDisplayObjectRepresentationDialog", "ViewMenu", "main_frame");
            akit.ValueChange("diaObjectRepresentation", "get_menu", "Color_ThisStatus");
            akit.PushButton("attrib_get", "diaObjectRepresentation");
            akit.PushButton("dia_modify", "diaObjectRepresentation");
            akit.PushButton("dia_cancel", "diaObjectRepresentation");
        }
    }
}
```

This is what the Drawing Macro Looks Like

- Record a macro from the Advanced Features option in the Applications & components catalog.
- Select the macro, right-click, select Edit in the context menu.

```
// Generated by Tekla.Technology.Akit.ScriptBuilder

namespace Tekla.Technology.Akit.UserScript
{
    public class Script
    {
        public static void Run(Tekla.Technology.Akit.IScript akit)
        {
            akit.Callback("acmd_display_selected_drawing_object_dialog", "", "main_frame");
            akit.TableSelect("pmark_dial", "gr_mark_elements", 15);
            akit.Activate("pmark_dial", "gr_mark_elements");
            akit.ValueChange("gr_mark_text", "gr_text", "(CTR'D)");
            akit.PushButton("gr_mark_prompt_ok", "gr_mark_text");
            akit.PushButton("pmark_modify", "pmark_dial");
            akit.PushButton("pmark_cancel", "pmark_dial");
        }
    }
}
```

Macros Example “Swap Handles”

C:\ProgramData\Tekla Structures\2016i\environments\usimp\macros\modeling\Swap Handles.cs - Notepad++

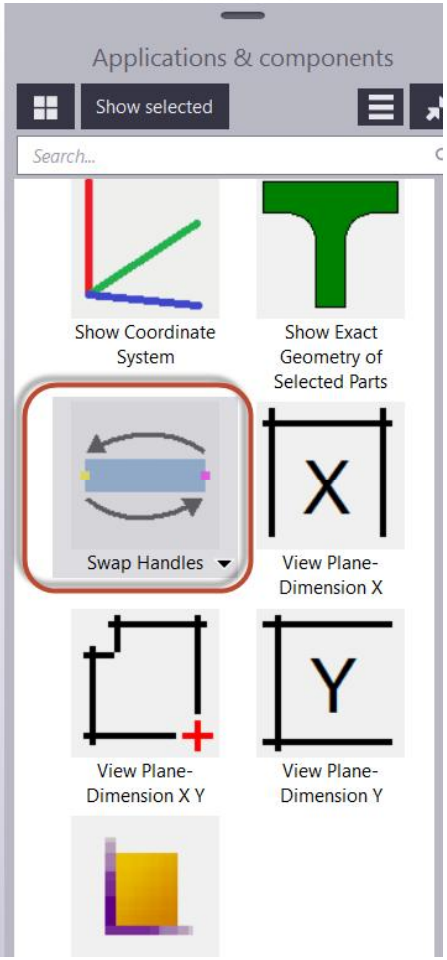
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Swap Handles.cs

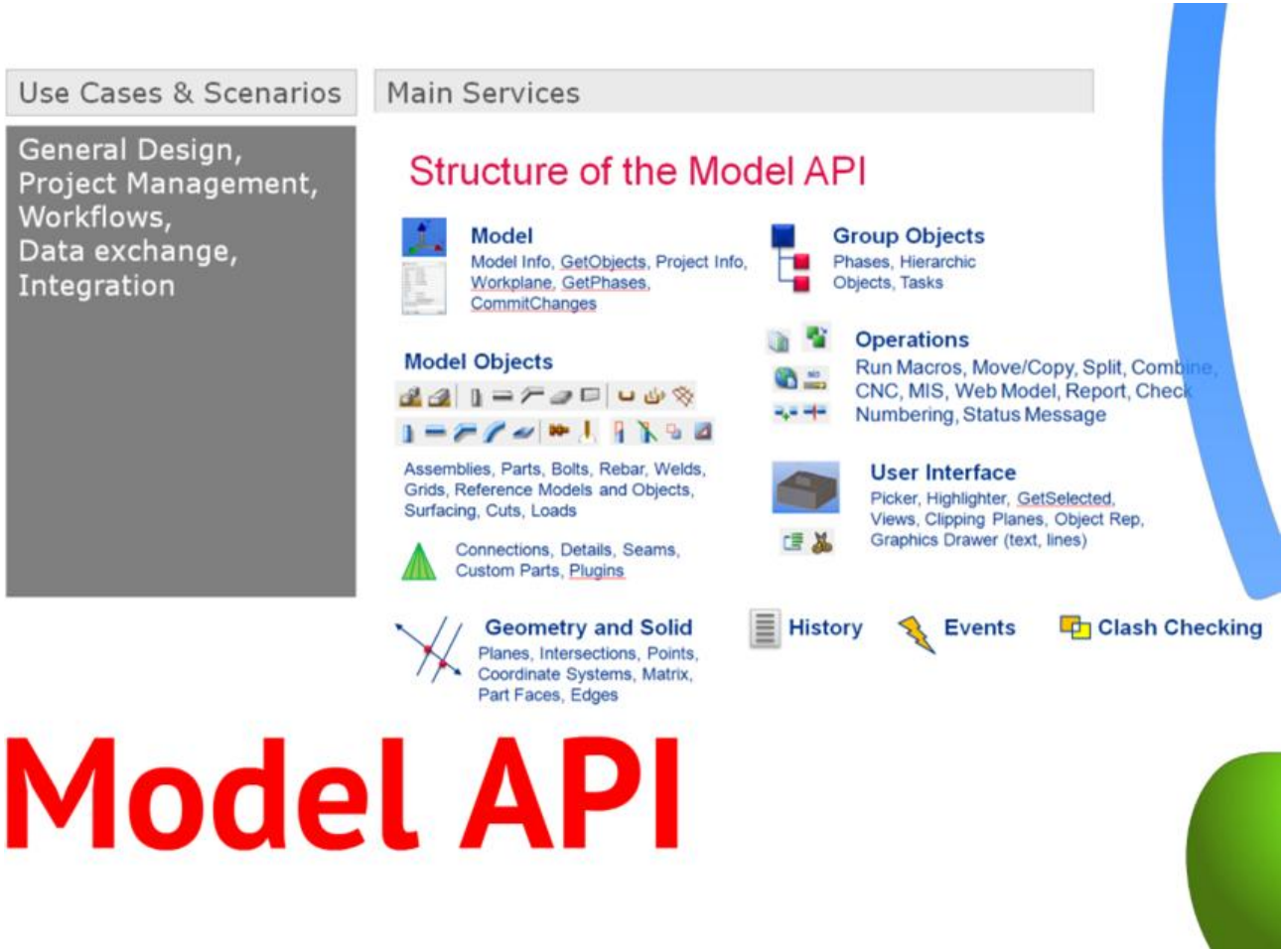
```
using System;
using System.Collections;
using System.Diagnostics;
using Tekla.Structures.Model;

namespace Tekla.Technology.Akit.UserScript
{
    /// <summary>
    /// Internal class for running logic
    /// </summary>
    public class Script
    {
        /// <summary>
        /// Internal method run automatically by Tekla Structures if using as raw c# file
        /// </summary>
        /// <param name="akit">Passed argument automatically by core when using as macro</param>
        public static void Run(Tekla.Technology.Akit.IScript akit)
        {
            try
            {
                new SwapHandles();
            }
            catch (Exception ex)
            {
                Trace.WriteLine(ex.Message + ex.StackTrace);
            }
        }
```

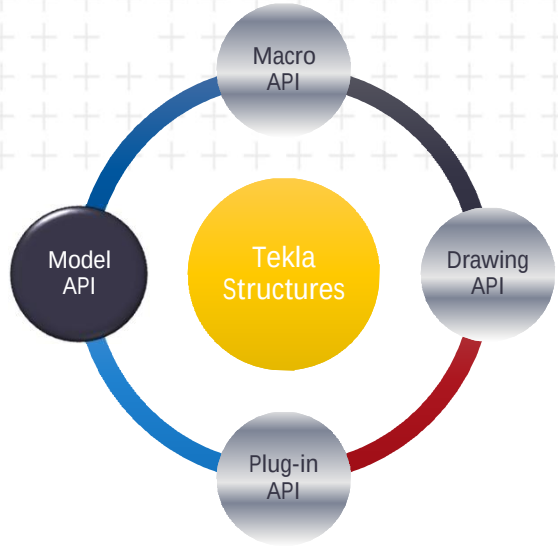
C# source file | length : 4,331 lines : 129 | Ln : 1 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS



Model API


Use Cases & Scenarios
Main Services
General Design,
Project Management,
Workflows,
Data exchange,
Integration
Structure of the Model API
Model
Model Info, GetObjects, Project Info, Workplane, GetPhases, CommitChanges
Group Objects
Phases, Hierarchic Objects, Tasks
Operations
Run Macros, Move/Copy, Split, Combine, CNC, MIS, Web Model, Report, Check Numbering, Status Message
Model Objects
Assemblies, Parts, Bolts, Rebar, Welds, Grids, Reference Models and Objects, Surfacing, Cuts, Loads
Connections, Details, Seams, Custom Parts, Plugins
User Interface
Picker, Highlighter, GetSelected, Views, Clipping Planes, Object Rep, Graphics Drawer (text, lines)
Geometry and Solid
Planes, Intersections, Points, Coordinate Systems, Matrix, Part Faces, Edges
History
Events
Clash Checking
Model API

Tekla Open API - Model



- Model API
  - Connect to a running Tekla Structures model
  - Create, modify, and delete model objects
    - Read and write object attributes
    - Read and write user defined attributes
    - Get report properties for objects
  - Interact with the user
    - Get currently selected objects
    - Prompt user to pick objects and locations
    - Select and highlight objects for the user
  - Access catalogs (material, bolt, profile, etc.)
  - Create and manipulate model views

Structure of the Model API

Model

Model Info, GetObjects, Project Info, Workplane, GetPhases, CommitChanges

Grouped Objects

Phases, Hierarchic Objects, Tasks

Model Objects

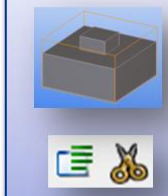
Assemblies, Parts, Bolts, Rebar, Welds, Grids, Reference Models and Objects, Surfacing, Cuts, Loads

Connections, Details, Seams, Custom Parts, Plugins

Operations

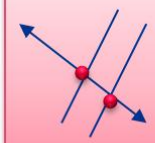
Run Macros, Move/Copy, Split, Combine, CNC, MIS, Web Model, Report, Check Numbering, Status Message

UI (User Interface)



Picker, Highlighter, GetSelected, Views, Clipping Planes, Object Rep, Graphics Drawer (text, lines)

Geometry



Planes, Intersections, Points, Coordinate Systems, Matrix

Solid

Part Faces, Edges

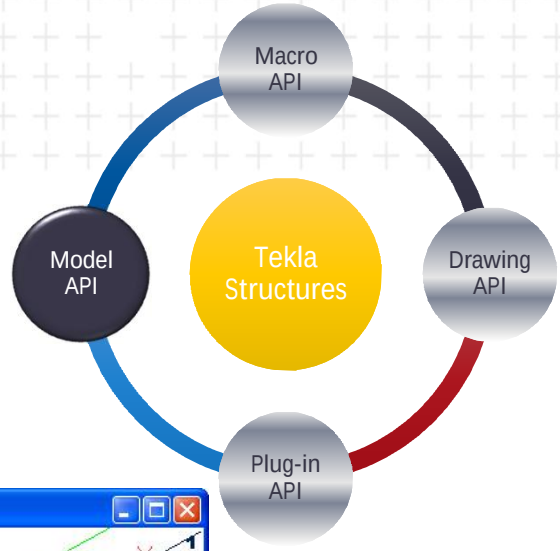
History

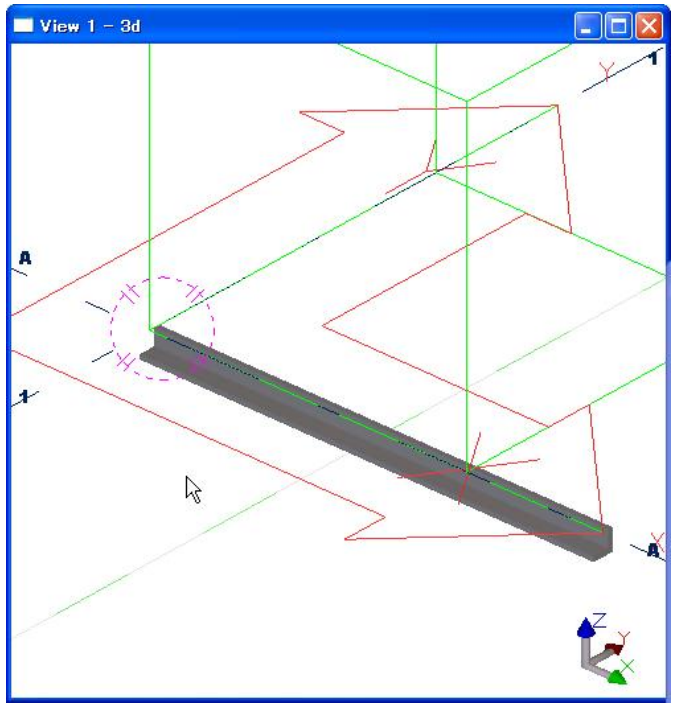
Events

Clash Checking

Model API – Sample program

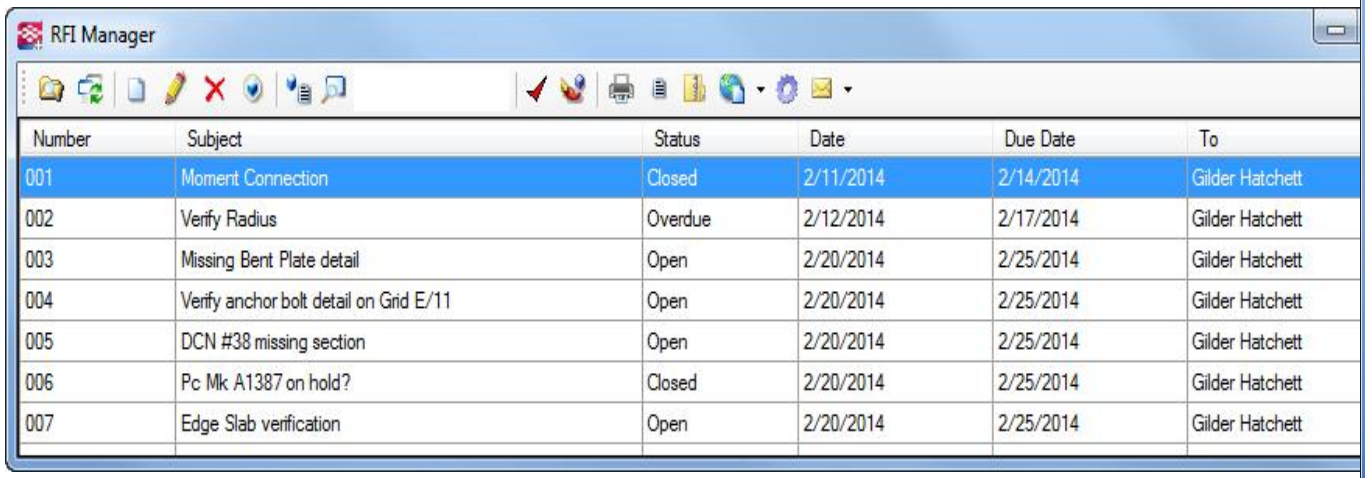
```
using System;
using Tekla.Structures.Model;
using Tekla.Structures.Geometry3d;
namespace ConsoleApplication1
{
 class Program
 {
  static void Main(string[] args)
  {
   Model myModel = new Model();
   Beam  myBeam = new Beam();
   myBeam.StartPoint = new Point( 0.0, 0.0, 0.0);
   myBeam.EndPoint   = new Point( 1000.0, 0.0, 0.0);
   myBeam.Profile.ProfileString = "L45*45*9";
   myBeam.Insert();
   myModel.CommitChanges();
  }
 }
}
```

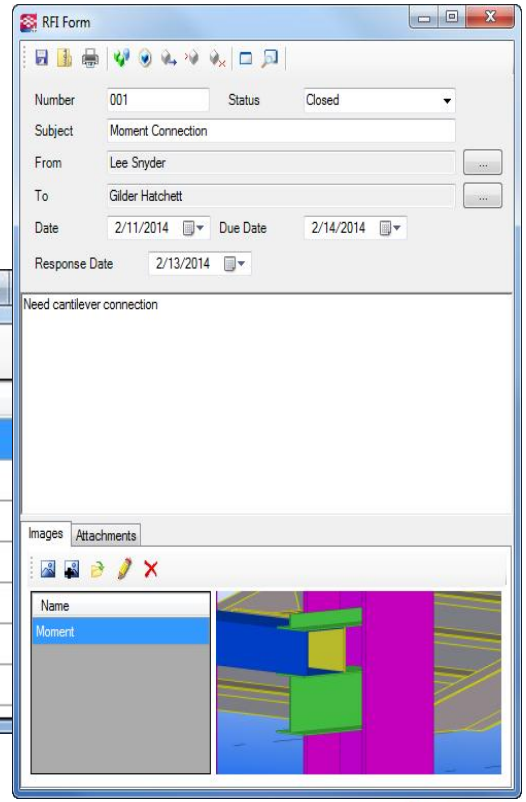




RFI Manager

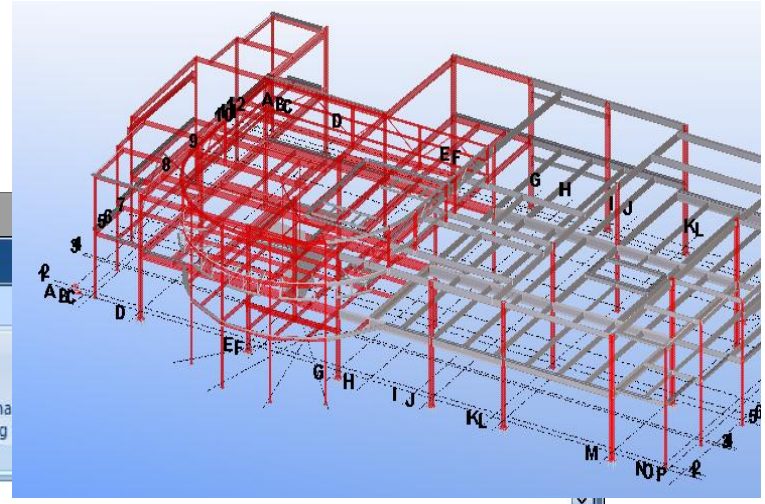
- Manage RFI's and link them with the model.
  - Link to IFC reference model objects as well as native objects
  - Colorize model based of RFI status
  - Automatically check for overdue status
  - Publish to HTML or FTP site or zip up and email





Construction Sequencer

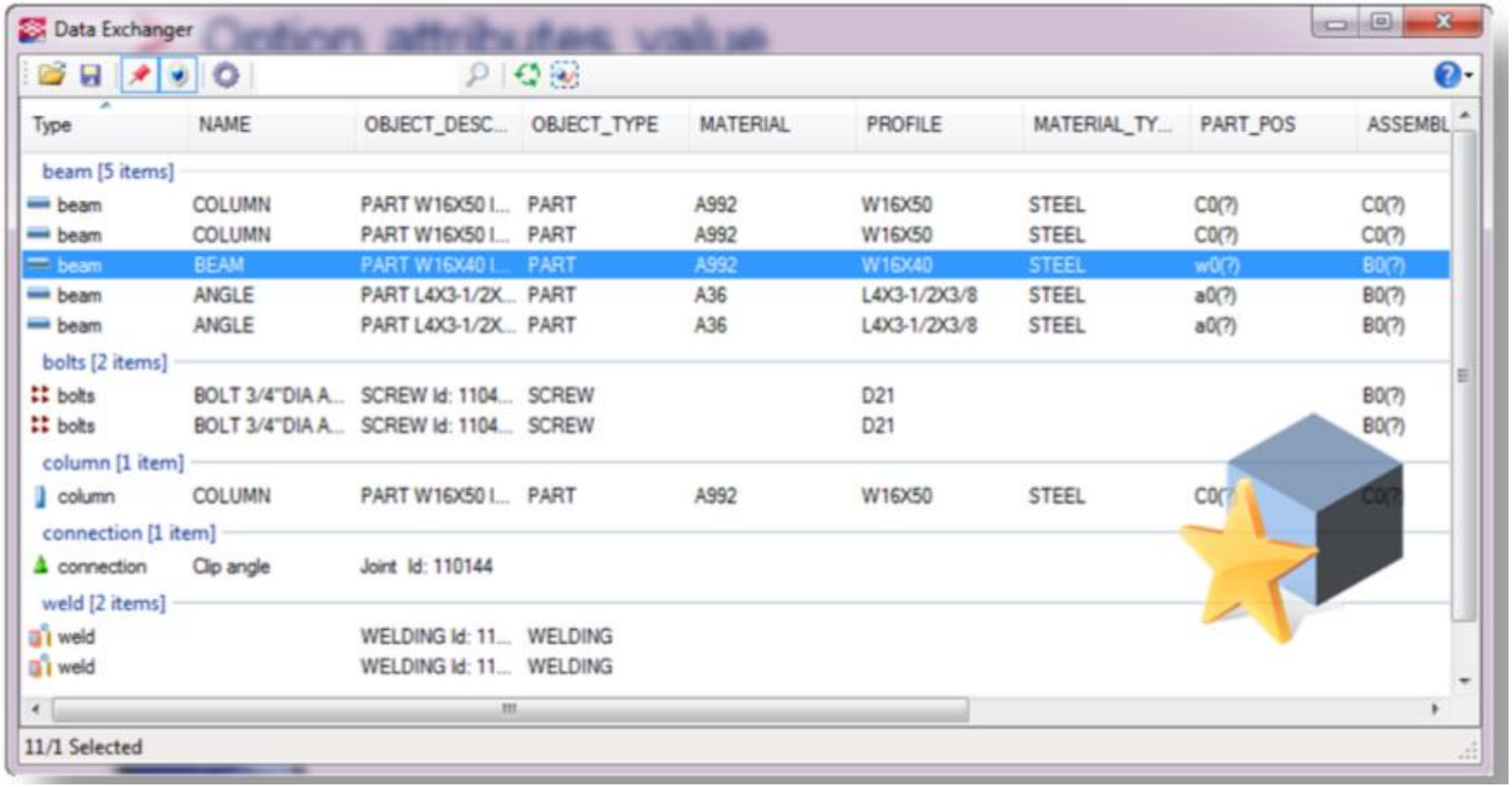
- Define erection order
- Create reports and animations



Excel Report.xlsx - Microsoft Excel

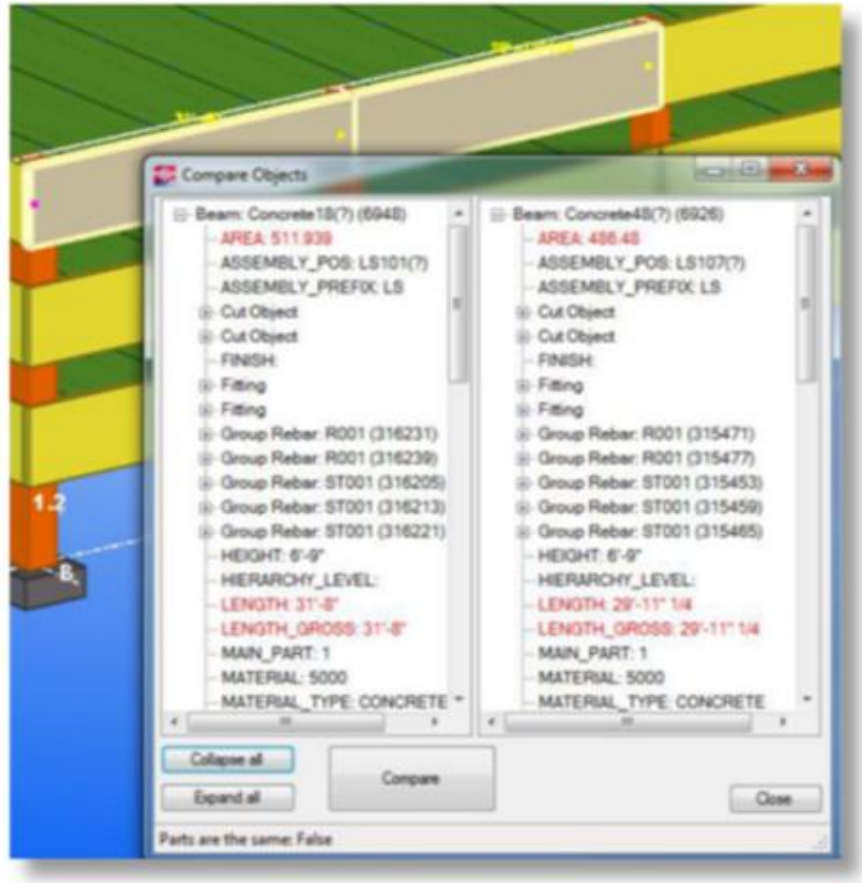
	A	B	C	D	E	F	G	H	I	J
1	Name	Phase	Order	AssemblyMark	Profile	TotalWeight	MaximumOverallL	GlobalCenterOfGravity	ElevationTop	NearestGridLocati
2	COLUMN	41	1	C1	W8X28	936.51	30'5"3/4	(-0"1/4, 30'8"1/16, 13'8"9/16)	29'3"3/4	A/5
3	COLUMN	41	2	C2	W8X28	955.16	31'2"	(-0"1/4, 63'4", 14'0"7/16)	30'0"	A/8
4	COLUMN	41	3	C3	W8X24	767.46	30'2"3/16	(4'0", 16'8", 13'1"1/2)	29'0"3/16	B/-3
5	COLUMN	41	4	C4	W8X24	783.81	30'10"3/8	(4'0", 77'4", 13'5"9/16)	29'8"3/8	B/9
6	COLUMN	41	5	C6	PIPE8STD	874.78	30'6"1/2	(6'0"1/16, 91'11"15/16, 13'7"9/16)	29'4"1/2	C/-
7	COLUMN	41	6	C5	PIPE8STD	856.94	29'10"1/2	(6'0"1/16, 2'0"1/16, 13'3"9/16)	28'8"1/2	C/2
8	COLUMN	41	7	C7	W12X79	2899.31	30'6"	(20'8"1/16, 0', 11'7"3/16)	28'8"	D/1
9	COLUMN	41	8	C8	W12X79	2887.62	30'5"3/4	(20'7"15/16, 30'8", 12'2"13/16)	29'3"3/4	D/5
10	COLUMN	41	9	C9	W12X79	3007.64	31'2"	(20'8", 63'4", 12'9"13/16)	30'0"	D/8
11	COLUMN	41	10	C10	W12X79	2852.67	31'2"	(20'8", 94'0", 11'7"1/2)	29'4"	D/11
12	COLUMN	57	11	C11	HSS8X8X1/4	291.07	9'9"3/8	(31'6"1/2, 10'6"1/4, 33'4"7/16)	38'8"	D-E/2-

Data Exchanger

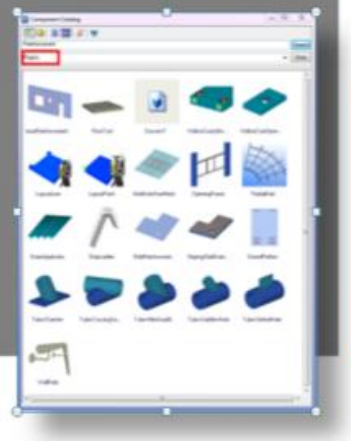


Compare Objects

- Deep object comparison
- Compare and highlight



Use Cases & Scenarios	Main Services & Features	
Detailing, System Components, Connections, Parametric modeling	**Performance** Core level speed, Nesting capability, Protected intellectual property	**Use** Found in Component Catalog, Automatically load on launch
	User Input Parametric properties, Save/Load templates	**Applicability** Serve both modeling and drawing



Plugin API

Tekla Open API – Plug-in

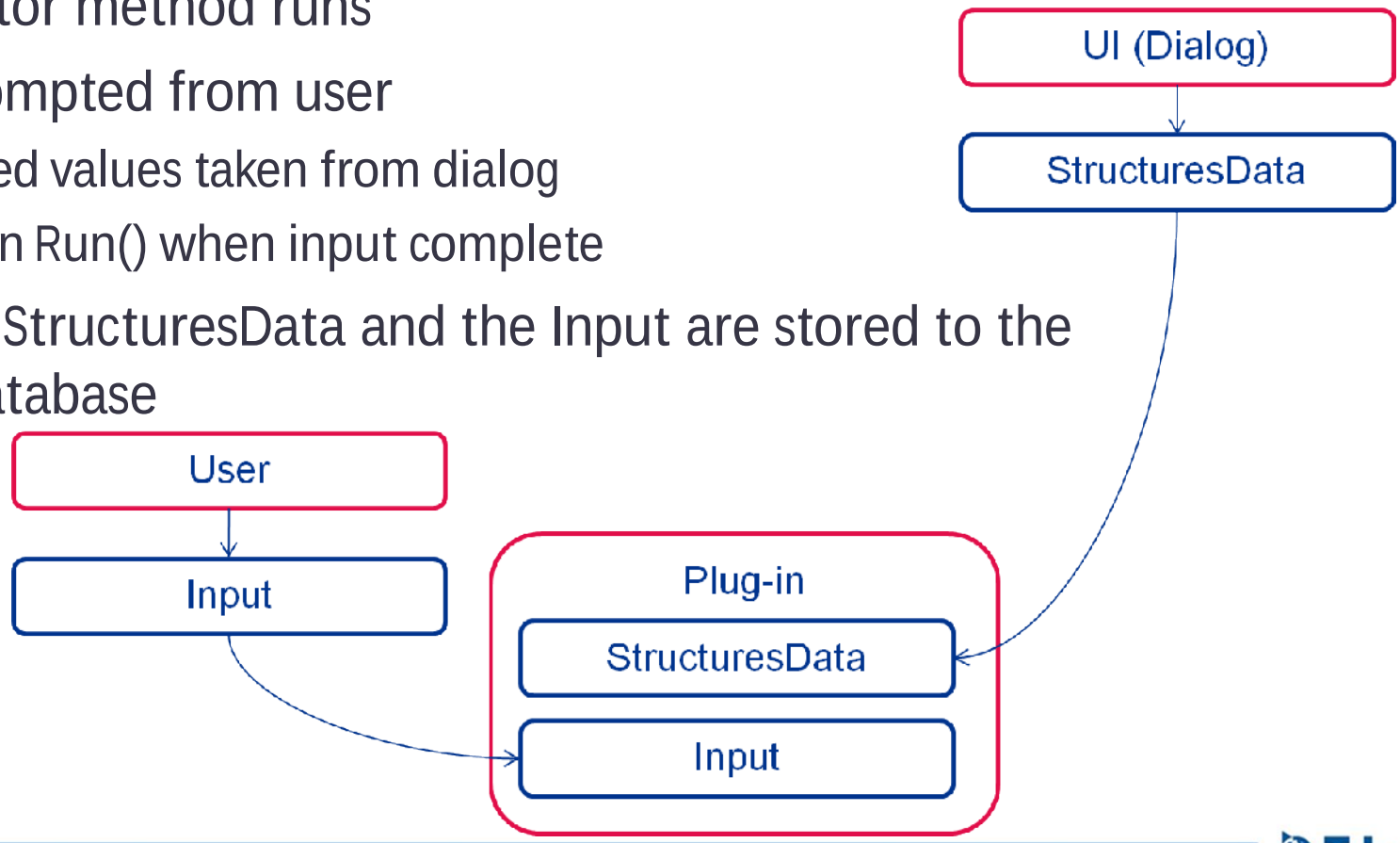
- Plug-in API
  - Can create intelligent objects.
    - Connections and details.
    - Stairs, floor bays, handrails, and more.
  - Automatically loaded at startup.
    - Found in the component catalog.
  - Plugins can be associated to input.
    - Automatically adapt to changes in input objects.
  - Provides Tekla Structures UI.
    - Same INP as custom components.
    - Units, translations, access to catalog.
  - Plug-ins for drawings.

Tekla Open API - Drawing

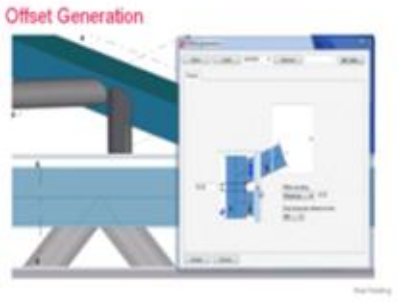
- Drawing API
  - Access the drawing list.
  - Create, modify and delete.
    - GA drawings and views in any drawing.
    - Dimensions, text and other basic objects.
  - Interact with the user.
    - Get currently selected drawing objects.
    - Prompt user to pick objects and locations.
    - Select and highlight objects for the user.
  - Access model objects from the drawing.
    - Select the parts found in a drawing in the model.

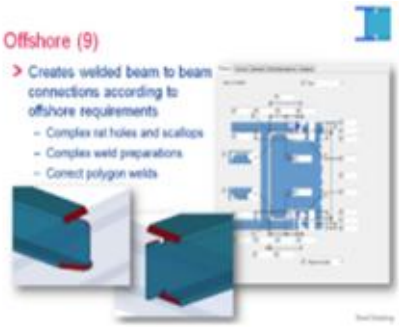
How a Plug-in is Structured

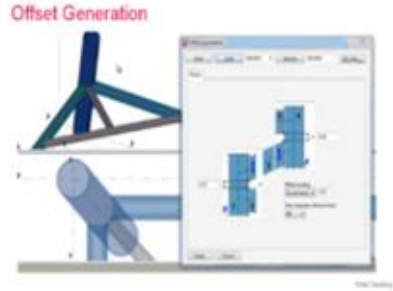
- New Plug-in/connection started
  - Constructor method runs
  - Input prompted from user
    - Applied values taken from dialog
    - Plug-in Run() when input complete
  - Both the StructuresData and the Input are stored to the model database

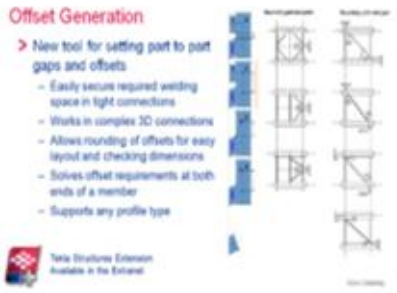


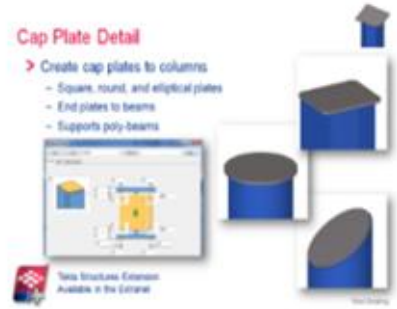
Plug-in API examples

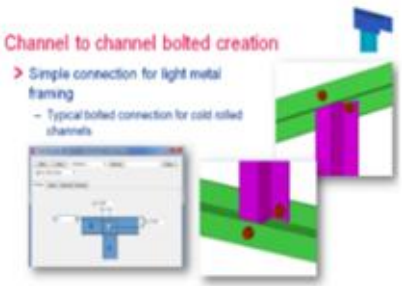


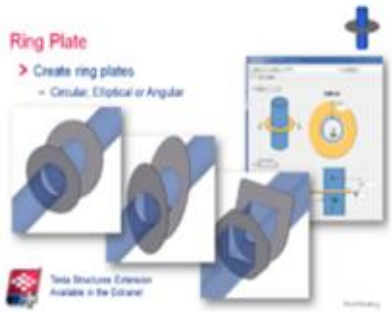


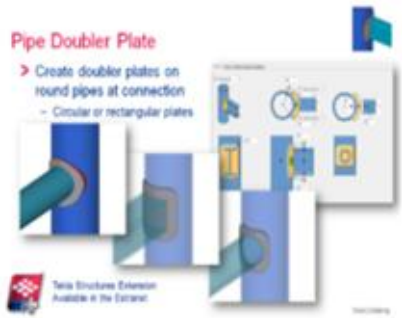




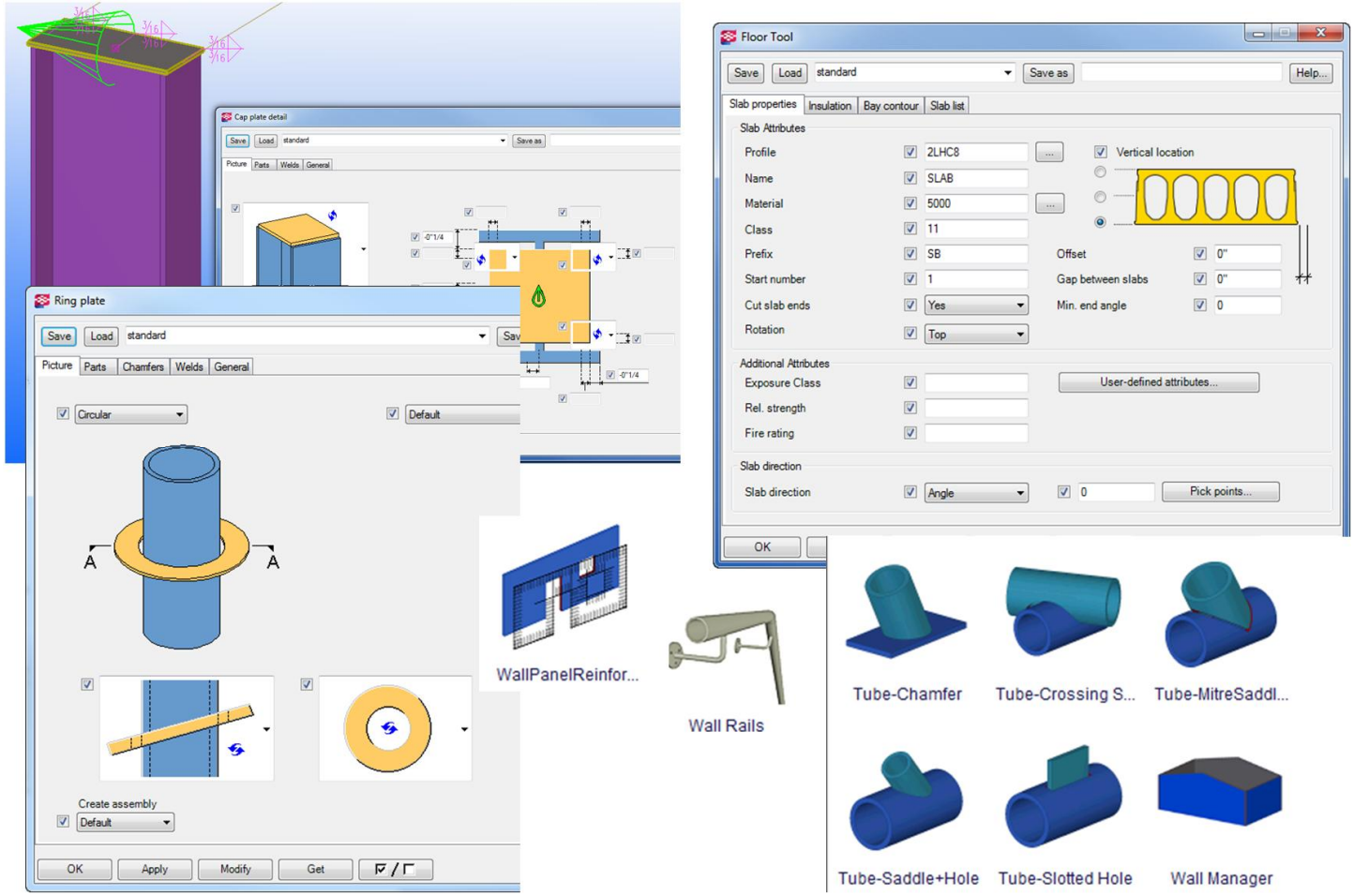








Plug-in API examples



Use Cases & Scenarios	Main Services	
Tailored Drawing Production and management Custom drawing objects , marks, dimensions, and symbols,	**Drawing object** Create, modify, delete, GA drawings, Views in drawing, Dimensions, text, labels, symbols,	**User Interaction** Pick, select, highlight
	Access Model Object Highlight model objects from drawing objects	**Access Drawing List** Iterate through drawings

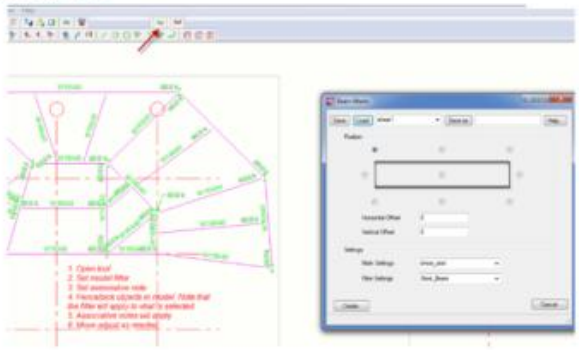
Drawing API

Common Use Cases

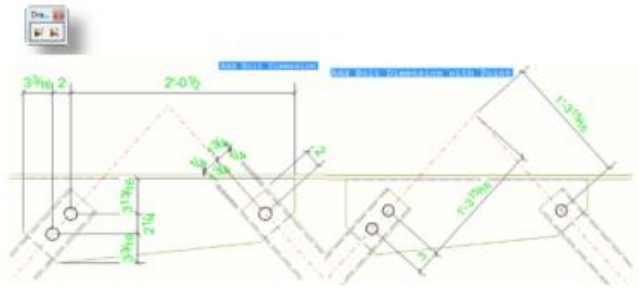
- Drawing API
  - Access the drawing list.
  - Create, modify and delete.
    - GA drawings and views in any drawing.
    - Dimensions, text and other basic objects.
  - Interact with the user.
    - Get currently selected drawing objects.
    - Prompt user to pick objects and locations.
    - Select and highlight objects for the user.
  - Access model objects from the drawing.
    - Select the parts found in a drawing in the model.

Drawing API examples

Beam Marks

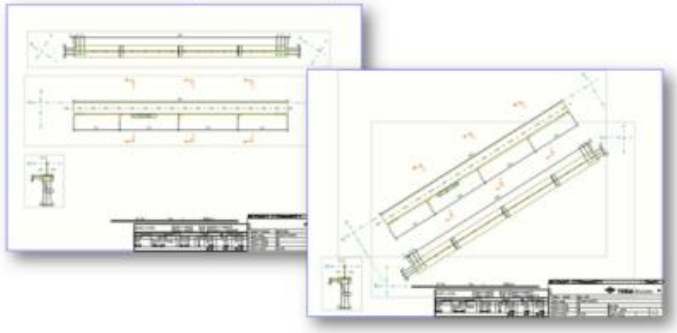


Bolt Dimension Tool



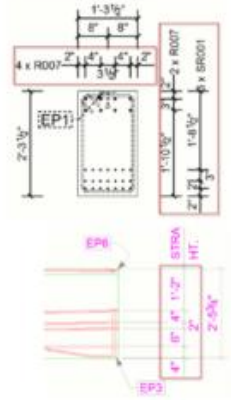
Brace Drawing

> Rotate drawing views to 'in plan' view



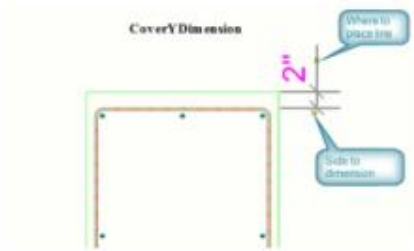
Rebar Dimension Tool

> Group Dimensions
> Automatic dimensioning
> Efficient with less user stes



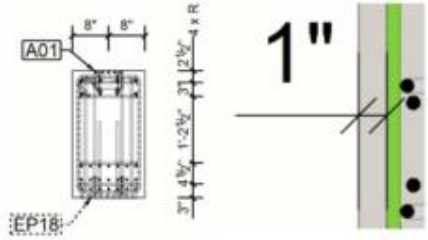
Drawing API examples

Cover Dimensions



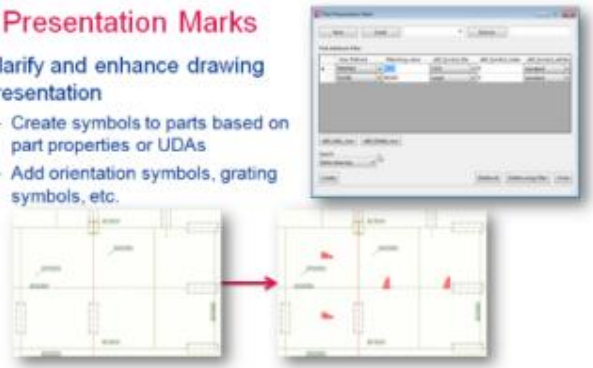
Drawing Dimensioning Tools

- Rebar Dimensioning
- Cover Dimension
- Drawing Detail Manager



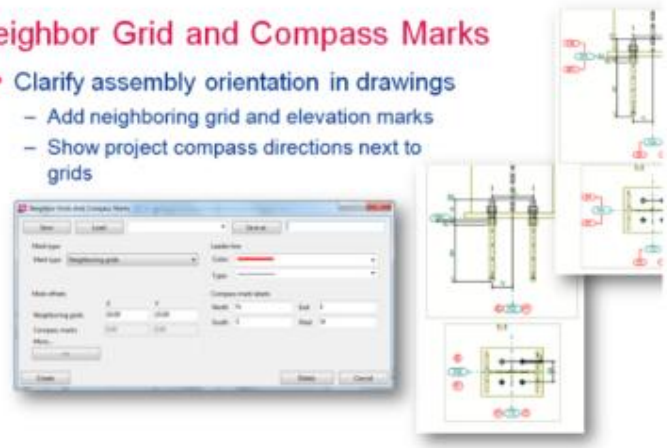
Part Presentation Marks

- Clarify and enhance drawing presentation
  - Create symbols to parts based on part properties or UDAs
  - Add orientation symbols, grating symbols, etc.



Neighbor Grid and Compass Marks

- Clarify assembly orientation in drawings
  - Add neighboring grid and elevation marks
  - Show project compass directions next to grids

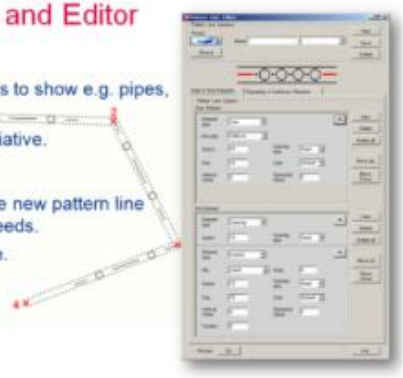


Drawing Plug-in API examples

Drawing Match Line

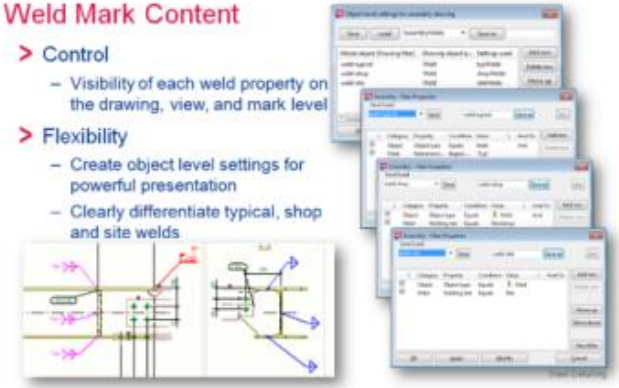
Pattern Line Plug-in and Editor

- Pattern line:
  - Create special line types to show e.g. pipes, ground, …
  - Line handles are associative.
- Editor:
  - Modify existing or create new pattern line types which suit your needs.
  - Graphical user interface.
  - Preview while editing.



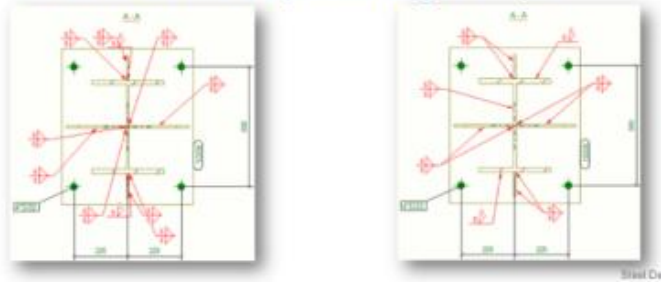
Weld Mark Content

- Control
  - Visibility of each weld property on the drawing, view, and mark level
- Flexibility
  - Create object level settings for powerful presentation
  - Clearly differentiate typical, shop and site welds



Weld Mark Leader Lines

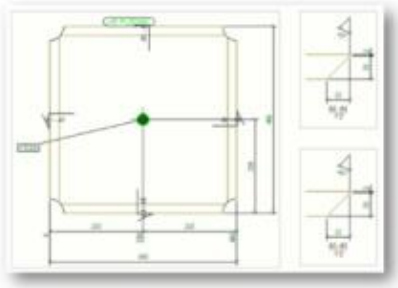
- Position the weld mark leader line
  - Clarify complex welding cases
  - Improve drawing presentation
  - Place the weld mark at any location along the weld path



Drawing Plug-in API examples

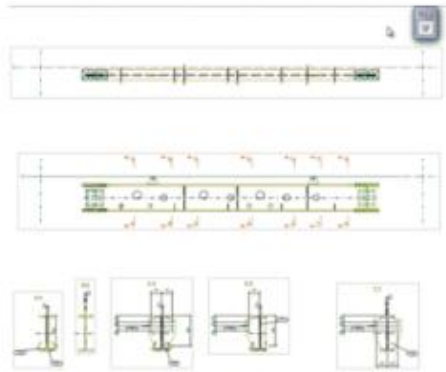
Edge Preparation Details

- Automatically create section views for plates with weld preparations.
- Views created with dimensioned details.
- Views created for workshop drawings.



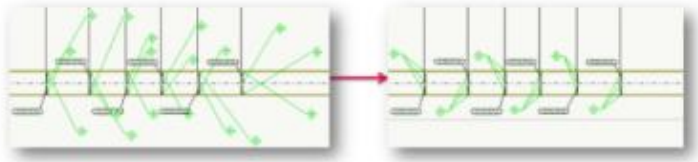
Dimensioning Tool

- New tool for adding dimensions to assembly drawings
- User defined rules for dimensioning



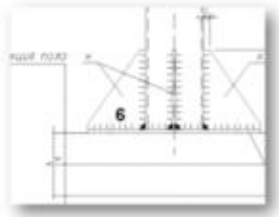
Weld Mark Merge

- Automatically merge identical weld marks
  - Intelligent grouping options
  - Simplify and clarify drawing presentation



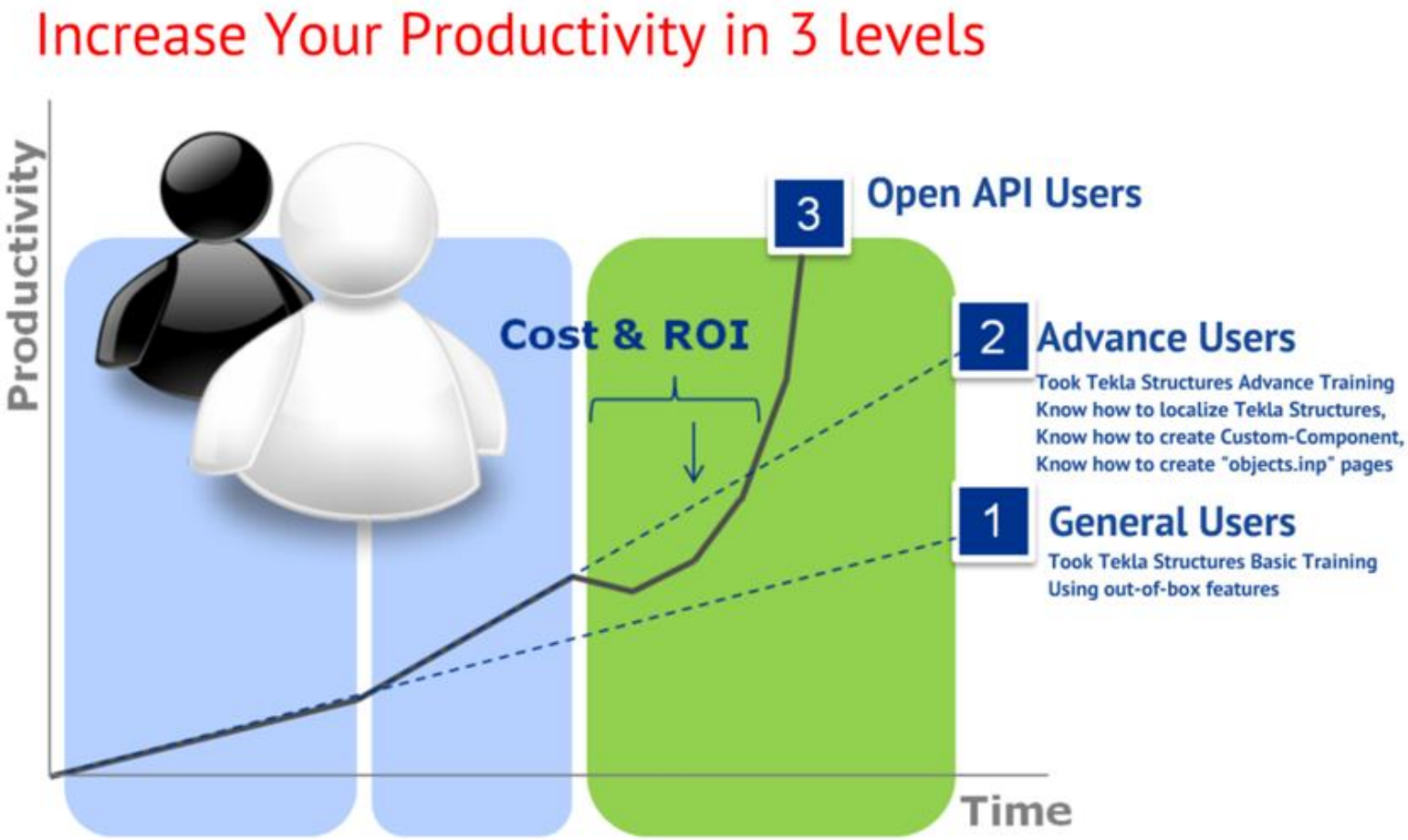
Draw Weld Path

- Clear presentation of weld location in drawings
  - Draw patterned lines to indicate weld paths and weld types



Plan to Utilize the API

Why Utilize the Tekla Open API



Strategy Overview

Formalize your Tekla Open API implementation strategy

> Ingredients

Process
- Know your strength
- Establish your goal

1

People
- Find the right people
- Motivated, passion, and open mind!

2

Technology

3

Tekla Open API comes with Tekla Structures

Tekla Open API is a collection of DLLs (Dynamic Link Library) that are included within your Tekla Structures!

Microsoft Visual Studio Express

Express 2013 for Windows

Visual Studio

Microsoft

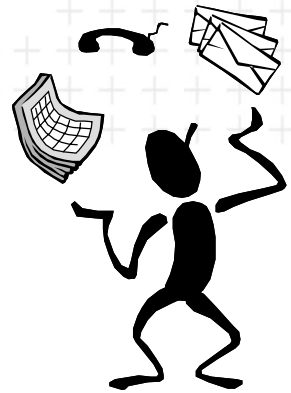
Download

System Requirements:

Supported operating systems

- Windows 7 SP1 (x86 and x64)
- Windows 8 (x86 and x64)
- Windows 8.1 (x86 and x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)
- Windows Server 2012 R2 (x64)

Tekla Open API - Required Skills

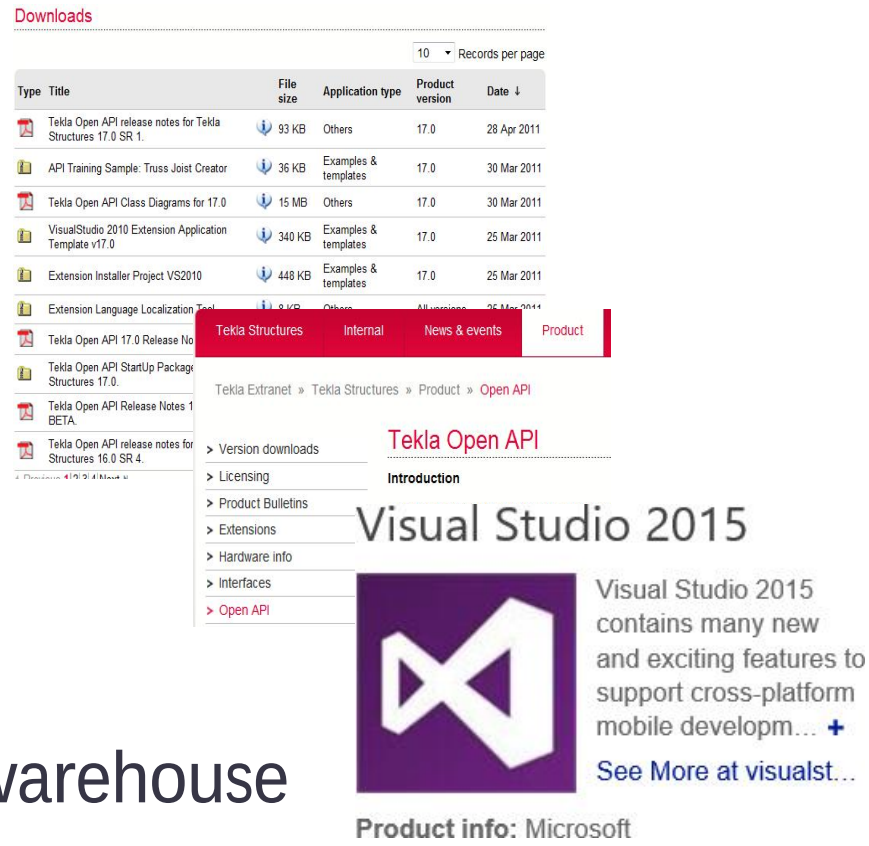


- Basic understanding of Tekla Structures
  - Model creation basics.
  - Part creation, alignment, and operations.
  - Understanding of Tekla Structures objects.

- Understanding/Experience with .NET/COM programming language
  - C#, VB.NET, J# etc.
  - English documentation.

Hardware and Software Needs

- Tekla Structures
- Windows 7, 8.1, 10
- .NET Framework 4.5
- Visual Studio 2015 Professional
- Pick a Language (Suggest C#)
- Tutorial Material
  - Videos, YouTube
  - Books
- Download Startup Package from warehouse
  - Search “Open API”



Hardware and Software Needs (Long Term)

- For debugging and more efficient long term development Visual Studio Professional Edition is recommended
- Packaging software: Windows Installer, Installshield, InstallAware, Advanced Installer, ...
  - Create MSI or exe installers
  - Easy to deploy (unattended and silent)
  - Version control
- Source Code Management System
  - E.g. Git, Stash, Seapine SCM
  - Code history, branch management, automatic builds

Set Goals for Long and Short Terms

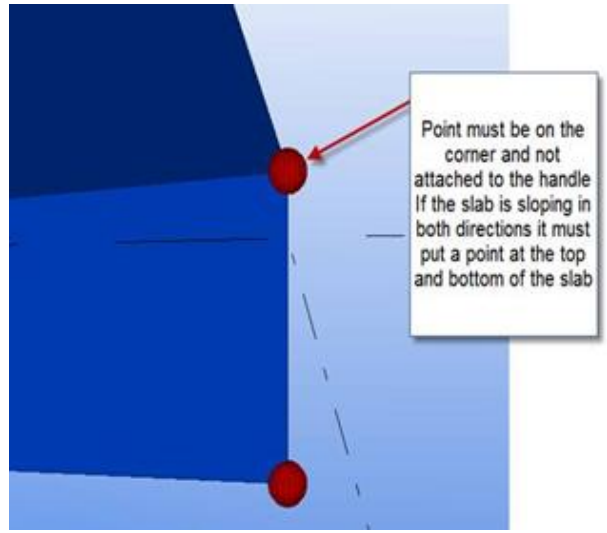
- Support from management
- Realize short term investment and benefits versus the bigger, long term goals that can be reached
- Get support structure for testing, training, and releasing
- Review existing process, asking what can be improved as you go, often workflows can be improved rather than add more automation to existing process
- Make sure to involve peers in documentation and review process
- Document and agree on stages for development

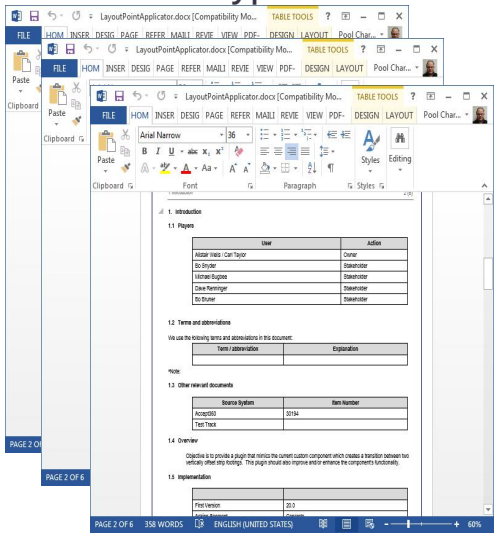
Target Specific Areas For Development

- Work with peers and identify key bottle neck areas in current process where software automation can help in productivity.
  - In some cases this can be simple macros to automate a set of tasks...
  - Direct export to ERP solution or in-house database?
  - Drawing dimensioning tools?
  - Process management tools?
- Document what you want to accomplish on paper, include peer review discussing risks and rewards.
- Check to be sure there is not already an existing solution on extranet.

Create Basic Documentation

- Why do this?
  - Current problem / need / gap: More accurate and faster field layout on slab boundaries
  - How does it solve or partially address the pain?: Quicker way to place layout points for slab edges and openings
- For whom do we do this?
  - Applicable customer: Segments – Concrete Contractors, SPGC
  - User types or roles: Survey guys, Field Engineers





- **Slab Plug-in** – applies layout points along grid boundaries
  - Set settings in plug-in UI
  - Pick Slab
  - Apply points along slab perimeter and openings
  - Options:
    - Top and bottom
    - Target spacing
    - Always a point at a corner

Setup

- Incorporate good coding techniques
- Use good MsBuild structure for version control
- Adopt a source code management system
- Reuse code in company toolkits
- Involve users in development through communication and suggestion channels
- Keep track of feature requests and bugs from users

Thank You