PTC® Live Global

PTC Creo Customization APIs - Overview

Bipin Kochar
PTC Creo Product Management

PTC Creo Customization APIs - Agenda

Agenda

- PTC Creo Parametric Customization
  - PTC Creo Parametric TOOLKIT
  - Other APIs
- The Creo API Strategy
- PTC Creo Object TOOLKIT
- PTC Creo 3.0 What's Exciting…
- PTC Creo 3.0 - Key Technical Changes …

PTC Creo Parametric TOOLKIT

A quick overview …

PTC Creo Parametric TOOLKIT - Overview

Prior to Creo, Pro/TOOLKIT was the primary customization and automation tool with access to PTC Creo Parametric.

With the launch of Creo, Pro/TOOLKIT is now re-branded as PTC Creo Parametric TOOLKIT.

PTC Creo Parametric TOOLKIT allows:

- Automating modeling for derived or single-use models driven by geometric or parametric constraints
- Monitoring the state of user's interactive session to enforce company rules or offer design advice
- Integrating external applications requiring access to PTC Creo Parametric data or operations
- Extending the PTC Creo Parametric User Interface with custom processes seamlessly embedded into the interface
- Automating tedious or difficult PTC Creo Parametric operations with applications designed to save design time and prevent mistakes
- Creating processes to run automatically without user interface on demand or at designated times

PTC Creo Parametric TOOLKIT - Overview

- Characteristics
  - C library programming interface to PTC Creo Parametric
  - Over 5000 functions
  - 2 modes of communication – DLL and IPC
  - Bi-directional data transfer between applications and PTC Creo Parametric

- User Requirements
  - C programming skills, knowledge of pointers and data structures
  - Development license to develop applications; applications may be unlocked and distributed within the enterprise without the PTC Creo Parametric TOOLKIT development license [i.e. no runtime license]
  - Advanced functionality licensed separately

PTC Creo Parametric TOOLKIT – Types of Applications

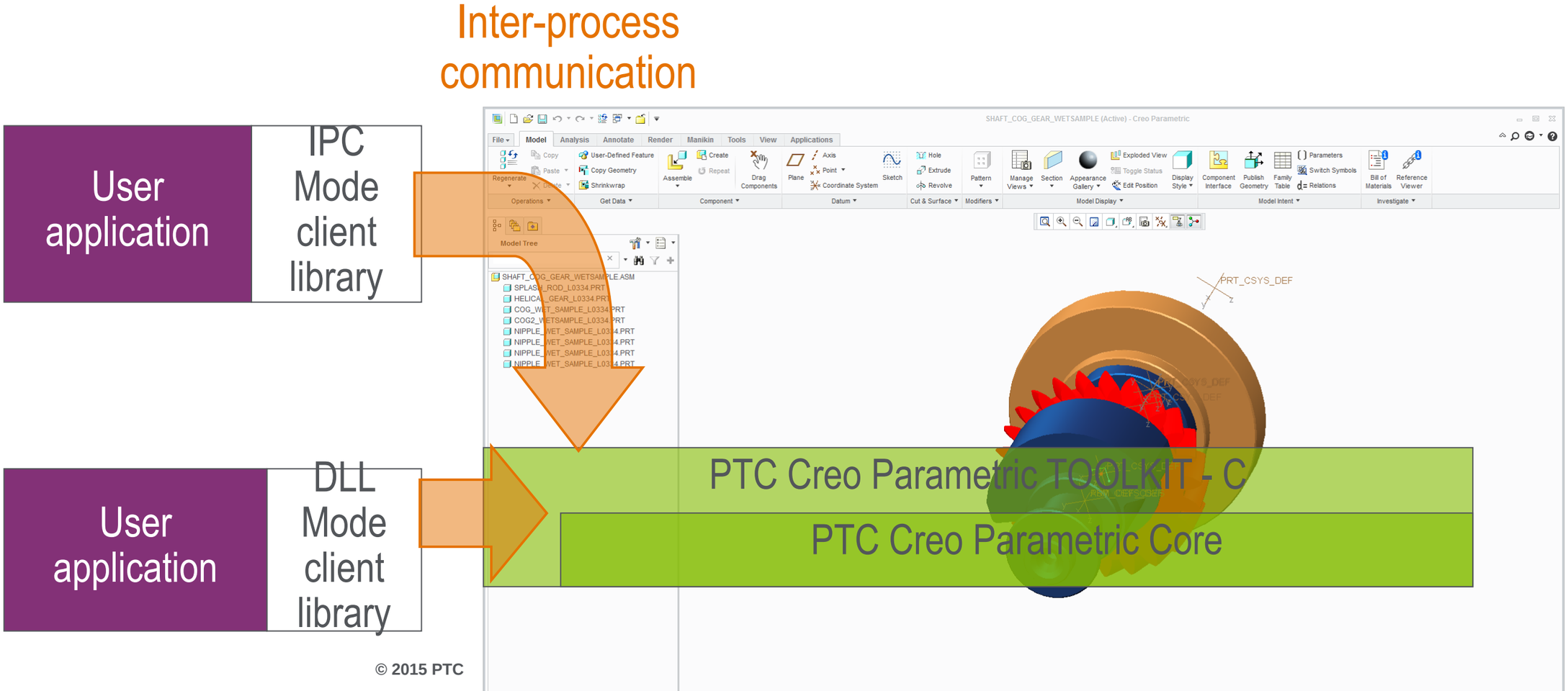
PTC Creo Parametric TOOLKIT supports two types of applications:

- Synchronous
  - Applications invoked from commands within a running session of PTC Creo Parametric
  - Mode of communication may be:
    - ***DLL - fastest method***
    - *spawn (multi-process) - slower, but offers more flexible link options [IPC Mode]*
- Asynchronous [IPC mode]
  - Applications running outside of PTC Creo Parametric that start or connect to one or more PTC Creo Parametric sessions
  - Communication is multi-process

PTC Creo Parametric TOOLKIT Architecture

PTC® Live Global

- DLL applications loaded directly into the PTC Creo Parametric process space
- IPC applications run as a separate process with inter-process communications passing parameters between the applications



Creating a PTC Creo Parametric TOOLKIT Application

- Compose the C or C++ Application.
  - Optional: use any one of several third party IDE's
- Compile the Application
  - Visual Studio 2010 [for PTC Creo 1.0 and 2]
  - See http://www.ptc.com/partners/hardware/support.html
- Configure the PTC Creo Parametric Environment
  - Provide application executable / DLL & text directory locations in registry file "creotk.dat"
- Start up PTC Creo Parametric
  - PTC Creo Parametric reads registry file to activate application
- Once development is complete, **unlock** the application so that others may use it without the PTC Creo Parametric TOOLKIT license

Sample creotk.dat file

```
name Product1
startup dll  (for dll mode apps)
exec_file /home/app1/<machine>/obj/frnpgm1.dll
text_dir /home/app1
end

name Product2
startup spawn (for spawn mode apps)
exec_file /home/app2/<machine>/obj/frnpgm2
text_dir /home/app2
end
```

PTC Creo Parametric TOOLKIT Applications – Launching …

PTC Creo Parametric TOOLKIT application code can be invoked by

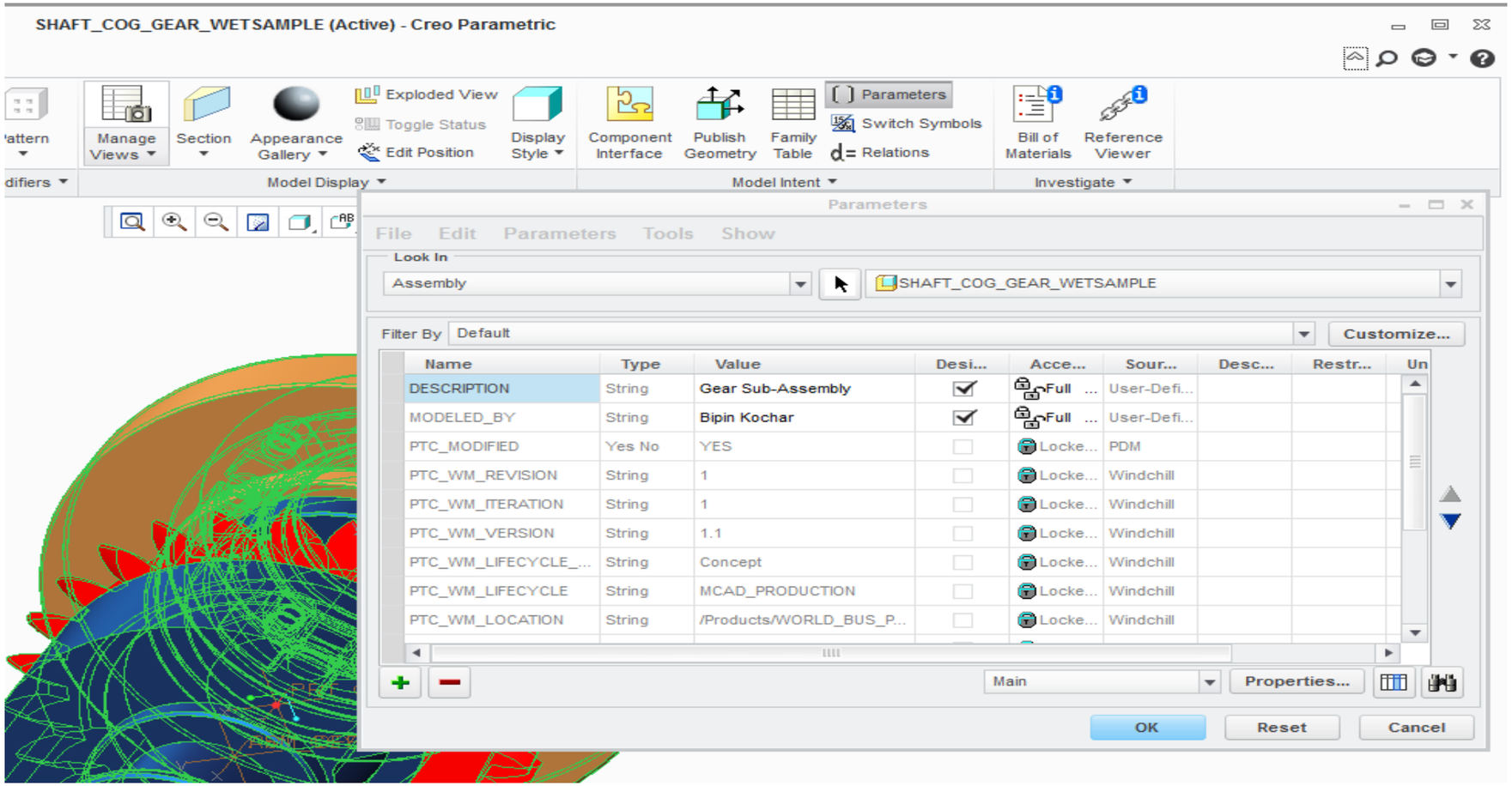
- PTC Creo Parametric startup
  - launched immediately upon registration of the PTC Creo Parametric TOOLKIT application
- User interface driven
  - invoked by user choosing application-created menu buttons and user interface components
- Event driven
  - invoked when certain events happen within PTC Creo Parametric
- Task driven
  - invoked by function calls from other external foreign applications

Example - PTC Creo Parametric TOOLKIT applications

PTC® Live Global

Customize PTC Creo Parametric TOOLKIT UI to automate repetitive or critical tasks

- model creation from templates
- geometry creation
- drawing creation



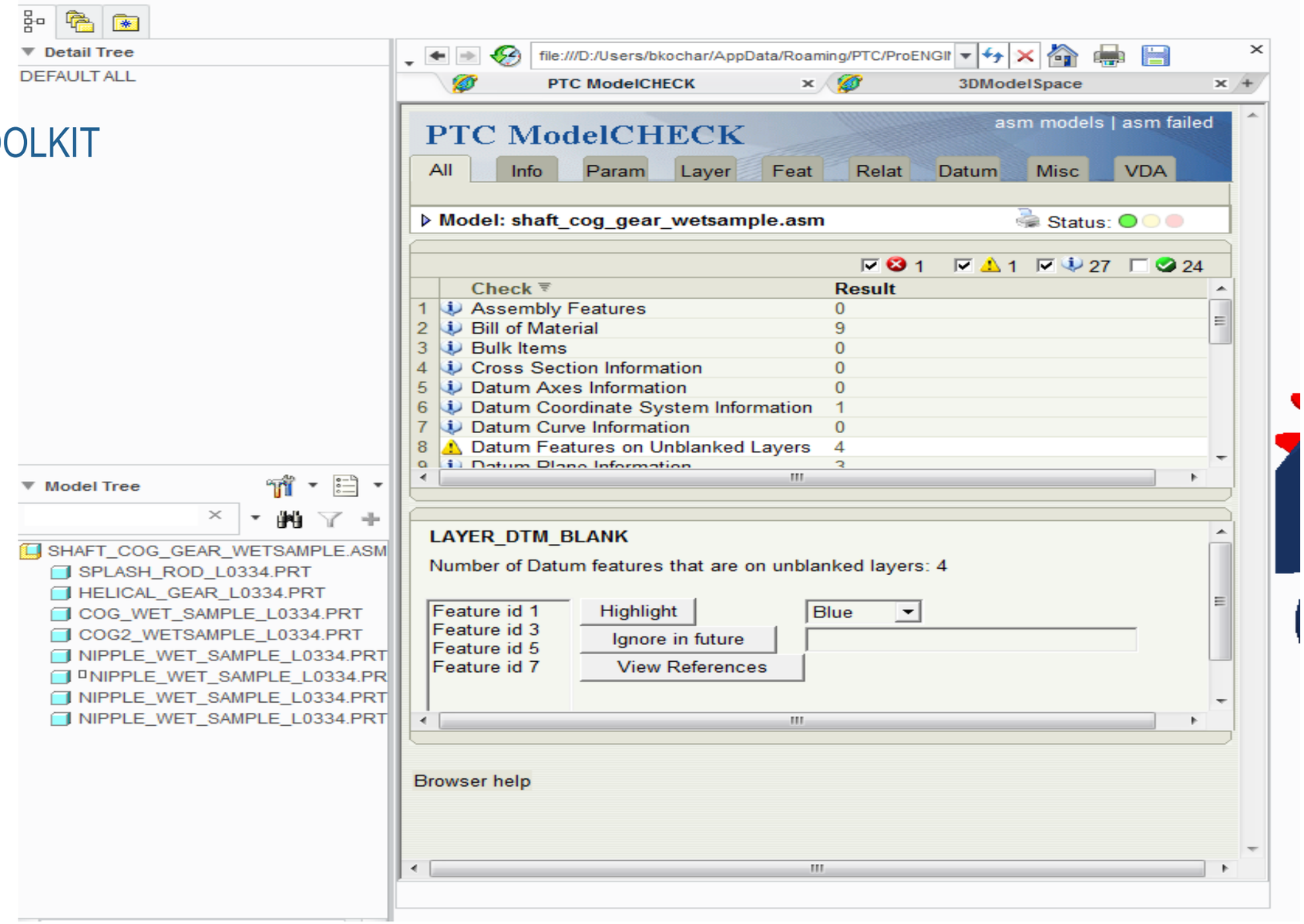
Example - PTC Creo Parametric TOOLKIT Applications

PTC® Live Global

PTC's ModelCheck:

Leveraging PTC Creo Parametric TOOLKIT functions to:-

- Enforce standards
- Verify models
- Assist repairs
- Validate user modeling



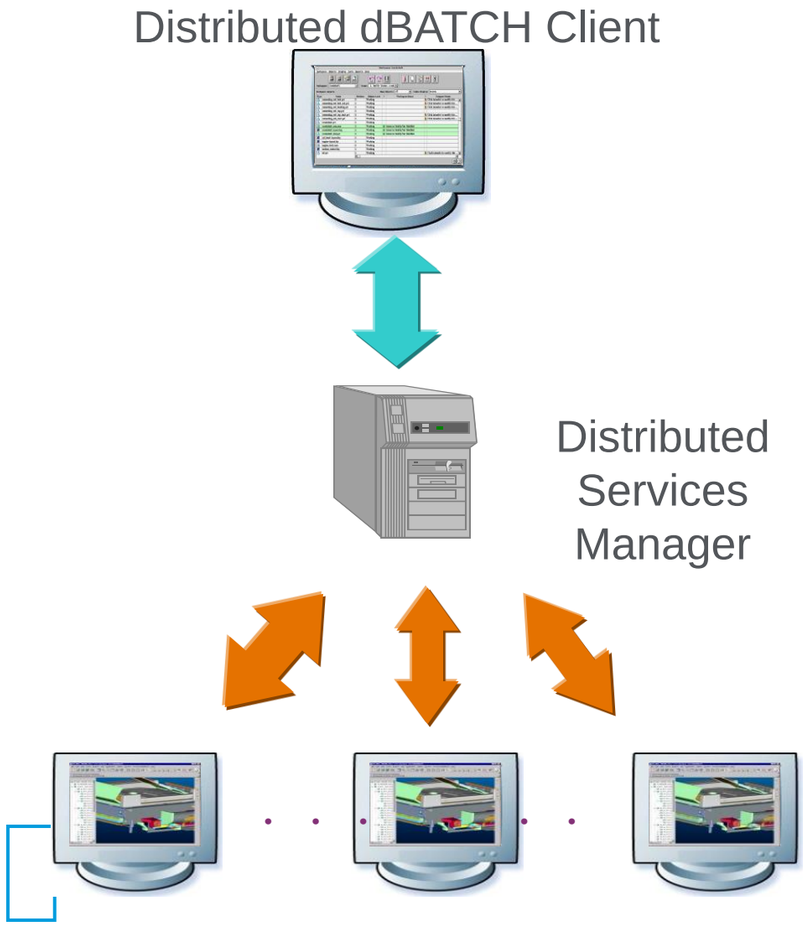
Example - PTC Creo Parametric TOOLKIT Applications

PTC's Distributed dBATCH:

- Includes PTC Creo Parametric TOOLKIT asynchronous applications as services that:
  - Generate an extensive set of EXPORT and IMPORT formats
  - Execute PTC ModelCheck
  - Plot and print models
  - Save models with display for preview in PTC Creo View Express
  - Number of other automation tasks …
  - What more …

Custom Tasks can be created …

… using PTC Creo Parametric TOOLKIT



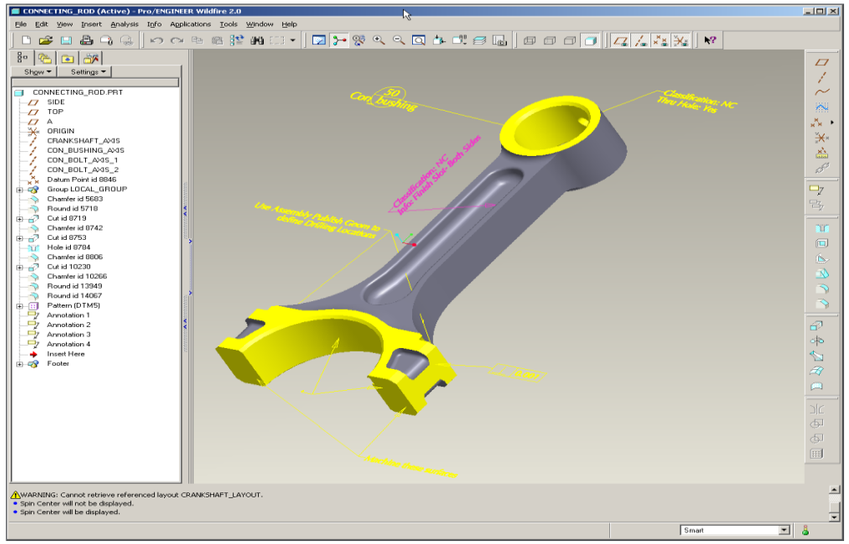
PTC Creo Parametric TOOLKIT – Advanced Extensions

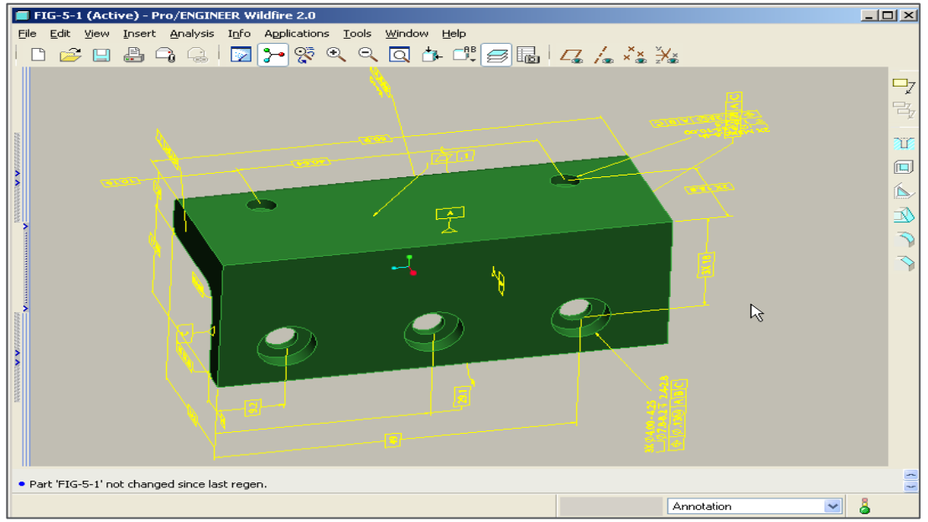
PTC® Live Global

Advanced TOOLKIT Extension for 3D Drawings

Provides Advanced Access to 3D drawings content

- Annotation features & annotation elements: Access and locate
- Gtols: New functionality related to annotation features
- Dimensions: Access driven and reference dims
- Symbols: Access 3D symbols
- Surface finish: Access symbol-based surface finish.





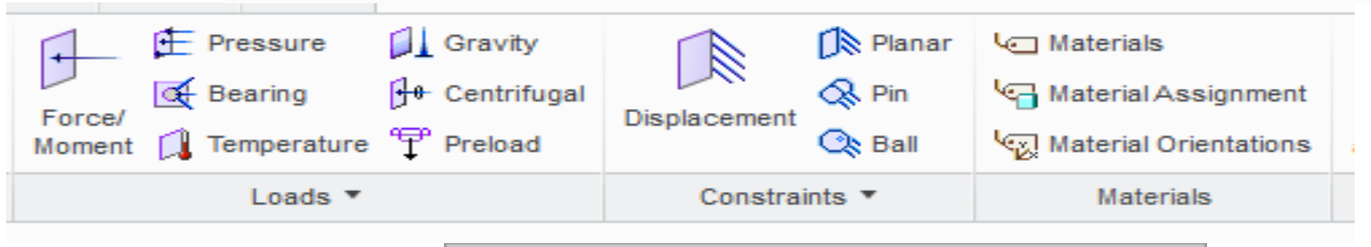
Note - This is a special royalty-bearing extension to core PTC Creo Parametric TOOLKIT, administered through the PTC Partner Advantage Program

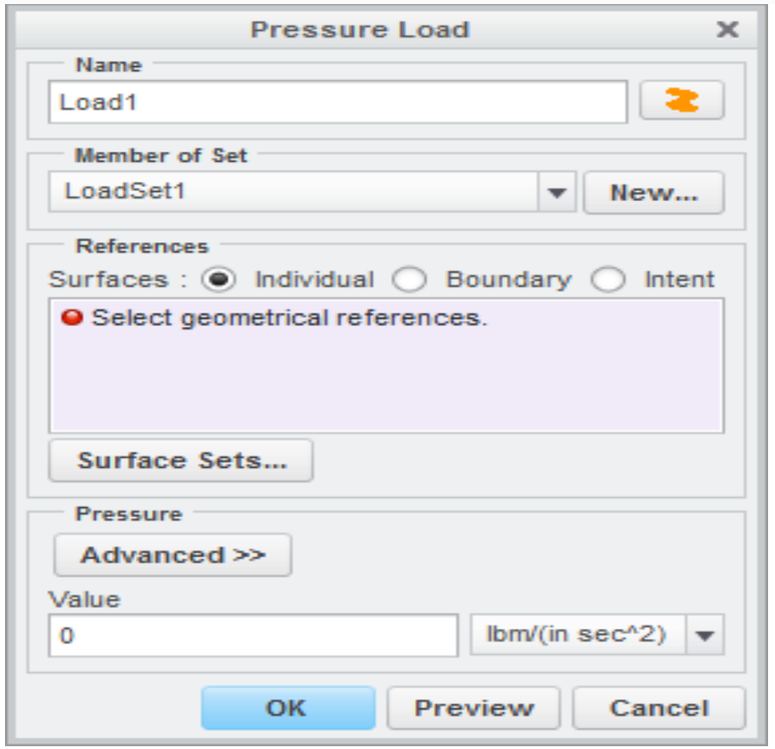
PTC Creo Parametric TOOLKIT – Advanced Extensions

PTC® Live Global

Advanced TOOLKIT Extension for Mechanica

- Read access to Simulation objects stored in the model
  - Loads/constraints/sets
  - Beams/sections/orientations/releases
  - Contacts/welds/interfaces
  - Shells/shell pairs/properties
  - Material orientations/mesh controls
  - Springs/spring properties
  - Simulation geometry (non-manifold geometry generated for analysis purposes)
- Most functionality requires the Mechanica run-time license
- Over 1000+ functions …





Note - This is a special royalty-bearing extension to core PTC Creo Parametric TOOLKIT, administered through the Partner Advantage Program

PTC Creo PFC APIs

A quick overview …

Creo PFC APIs

- In addition to the C APIs, PTC Creo Parametric also has APIs with other language bindings, namely:-
  - Java – J-Link
  - Javascript – Web.Link
  - Visual Basic - VB API for Creo Parametric

  These are popularly, know as PFC APIs.
- PFC APIs are free with your PTC Creo Parametric seat …

  … i.e. do NOT require a development or run-time license.
- PFC APIs provide access to a limited set of functionalities in PTC Creo.
- Targeted for simple applications …
- Are inter-operable with PTC Creo Parametric TOOLKIT –

  - enables you to more build complex applications mixing these appropriately.

J-Link APIs – Capabilities

J-Link is a Java API.

- The functionality provide by J-Link includes:-
  - Core Creo part, assembly, and drawing manipulation
  - Windchill server access
  - Data exchange
  - Creo User Interface customization (limited)
  - Applications triggered by events occurring in the Creo Parametric session
- Designed for connecting both synchronously and asynchronously to a Creo Parametric session.
- Requires no license beyond the license to run PTC Creo Parametric.
- Requires Java 7 Update 25
- Technical documentation in form of User Guide and an APIWizard.

J-Link and PTC Creo Parametric TOOLKIT can be used in one application …

… provide flexibility of utilizing both technologies in the most effective manner.

Creo Web.Link – Capabilities …

Web.Link is a Javascript API.

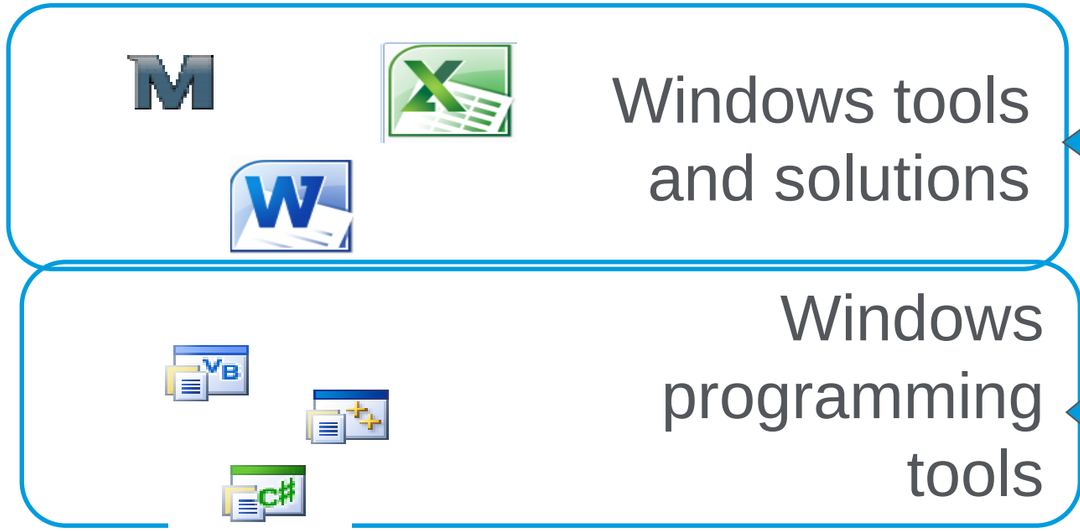
- Enables users interact with Creo Parametric through a browser.
- With Web.Link and a basic knowledge of HTML and JavaScript, users can create customized Web pages, which can be used to automate and streamline aspects of their engineering process.
- Web.Link capabilities include:-
  - Starting up a new local Creo session or connecting to an already running one
  - Performing file management operations such as retrieving, renaming, or copying design information
  - Simplifying assembly representations, performing assembly operations such as component replacement or control of explode states
  - Assigning materials, calculating mass properties, and changing dimensions and parameters
  - Performing interactive selection and regeneration
  - Annotating 3D models
  - PTC Windchill Server Access
- Requires no license beyond the license to run Creo Parametric.
- Technical documentation in form of User Guide and an APIWizard.
- Web.Link is qualified with the embedded browser in Creo Parametric and with Internet Explorer

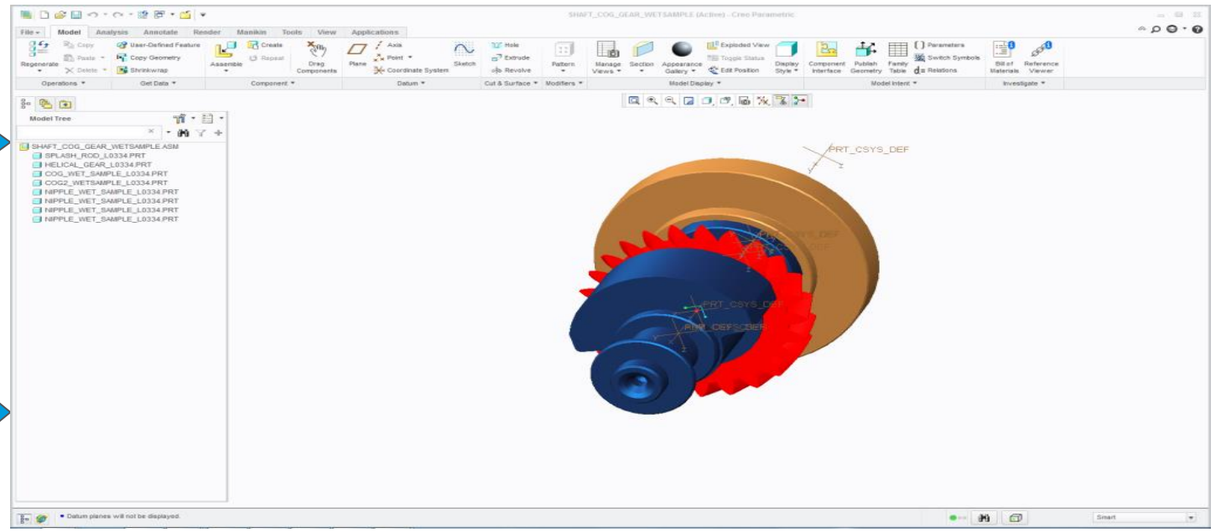
Visual Basic for PTC Creo Parametric

VB APIs provide the same capabilities as asynchronous J-Link, such as:-

- Core Creo part, assembly, and drawing manipulation
- Windchill server access
- Data exchange
- Applications triggered by events occurring in the Creo Parametric session

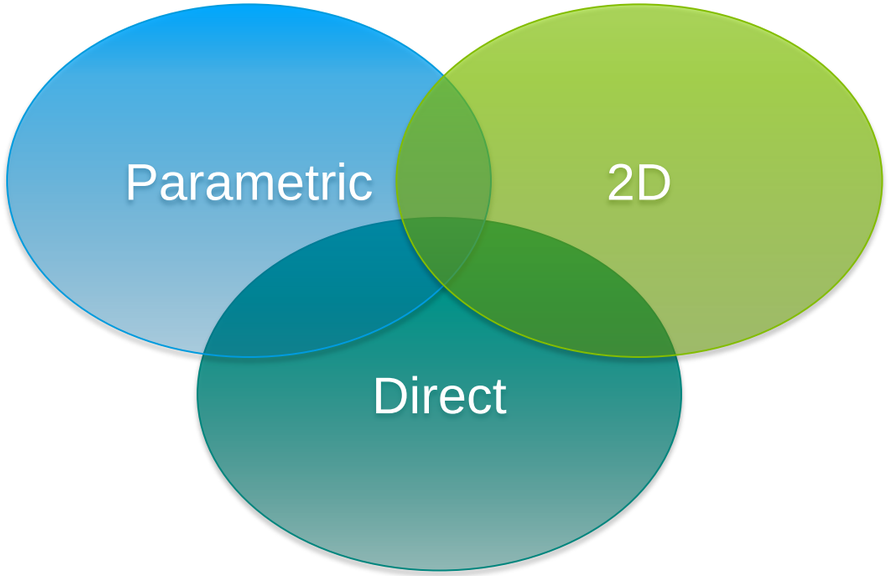
- Compatible with Visual Basic .NET 2010 and Visual Basic for Applications
- Designed for connecting asynchronously to a Creo Parametric session.
- Requires no license beyond the license to run Creo Parametric.
- Technical documentation in form of User Guide and an APIWizard.

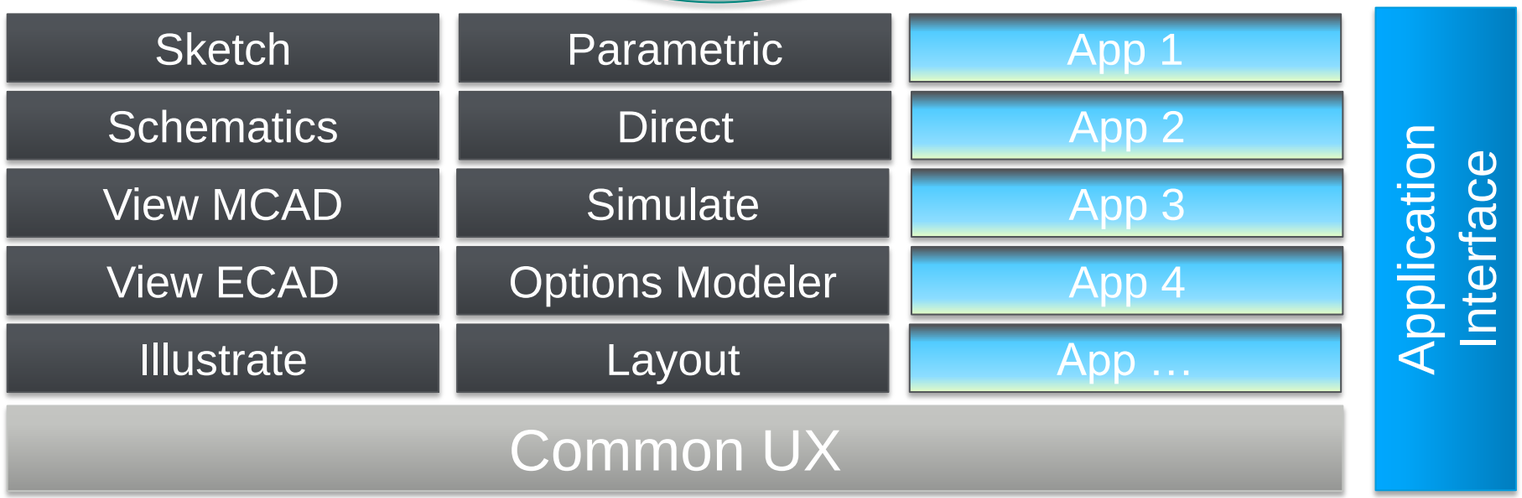




Creo API Strategy

PTC Creo Product Strategy – where we are headed…





- Many creation paradigms
- Many applications
  - Carving off existing capabilities
  - New organic development
- Access to the PTC Creo database
  - Openness
  - Long term archiving
  - 3rd party application development

PTC Creo API Strategy Drivers …

Key Drivers for the Creo API Strategy are:-

- Provide a rich API for partners to build new Creo Apps.
- Support Customization of new PTC Creo Apps like PTC Creo Direct …
- Modernization of Technology
- Improve Developer Productivity
- Enable Code re-use across Creo Apps
- Provide multiple bindings for the APIs

We hence introduced the Creo Object TOOLKIT C++ in Creo 1.0

Why use Object TOOLKIT C++?

Some reasons are:-

- Designed to be the common API across Creo authoring apps
- Code is Object Oriented
- Improved Data Integrity
- Data type conversions are automatic – e.g. no need to convert from String to Wstring or vice versa – makes the code more *readable and easier* to develop.
- Uses Smart Pointers. This reduces burden on the developer to manage the object persistence when developing an application.
- Uses Factory APIs. Code is more efficient, and reduces risk of improper use.
- Inter-operable with PTC Creo Parametric TOOLKIT and other toolkits.
- Designed to minimize transition from Creo Parametric TOOLKIT

Enables re-use of Customizations across apps.

Improved development productivity and quality of customizations

Enables smooth transition to new technology …

*What more, Object TOOLKIT C++ is **free** with your PTC Creo Parametric TOOLKIT license*

Sample snippet of code for Feature Creation using element tree …

C TOOLKIT

To fill the first 3 elements of the tree, following is the ProTOOLKIT code:

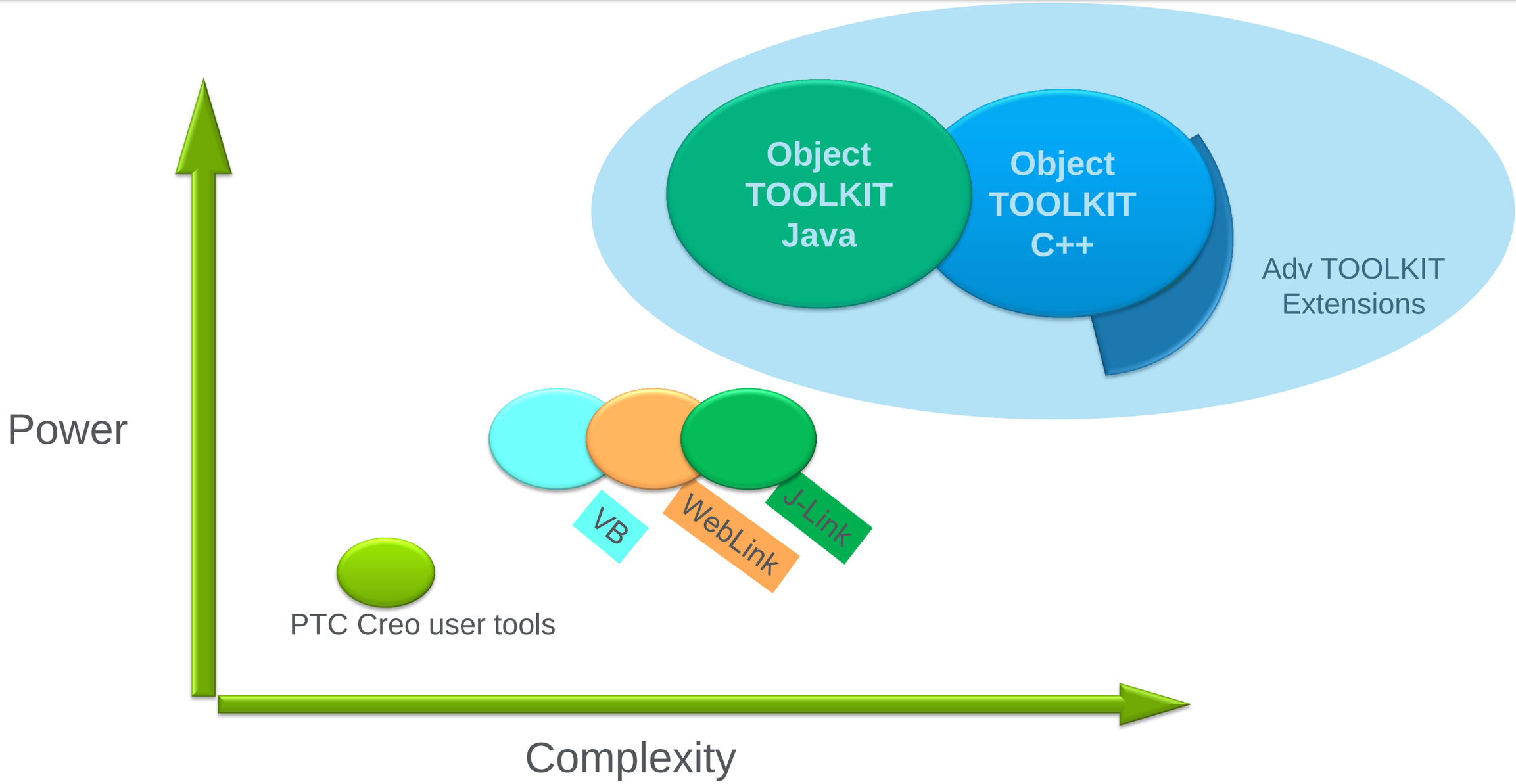
```
/*PRO_E_FEATURE_TREE */
        status = ProElementAlloc(PRO_E_FEATURE_TREE,&elem_0_0);
/*PRO_E_FEATURE_FORM*/
        status = ProElementAlloc(PRO_E_FEATURE_FORM,&elem_1_1);
        status = ProElementIntegerSet(elem_1_1,4);
        status = ProElemtreeElementAdd(elem_0_0,NULL,elem_1_1);
 /*PRO_E_SWEEP_TYPE*/
        status = ProElementAlloc(PRO_E_SWEEP_TYPE,&elem_1_2);
        status = ProElementIntegerSet(elem_1_2,PRO_SWEEP_TYPE_MULTI_TRAJ);
        status = ProElemtreeElementAdd(elem_0_0,NULL,elem_1_2);
```

Object TOOLKIT code is much cleaner and simpler …

Object TOOLKIT C++

```
//PRO_E_FEATURE_TREE
elements->append(wfcElement::Create(PRO_E_FEATURE_TREE, 0, 0););
//PRO_E_FEATURE_FORM
elements->append(wfcElement::Create(PRO_E_FEATURE_FORM, pfcCreateIntArgValue(4), 1));
//PRO_E_SWEEP_TYPE
elements->append(wfcElement::Create(PRO_E_SWEEP_TYPE, pfcCreateIntArgValue(PRO_SWEEP_TYPE_MULTI_TRAJ), 1));
```

PTC Creo Parametric Automation Trends



PTC Creo API Strategy …

- PTC Creo Parametric TOOLKIT will continue to see enhancements
  - Expansion of functional coverage
  - Updates to the data model changes
- PFC APIs will continue to be supported but not see any enhancements or updates.
- Focus will be on enhancing the Object TOOLKITs
  - Expanding coverage to all the functionality available in Creo Parametric TOOLKIT
  - Adding new functionality not available in Creo Parametric TOOLKIT
  - Expanding the coverage to other Creo apps like PTC Creo Direct and PTC Creo Layout
  - Newer language bindings will be introduced [Java etc…] to support migration from PFC APIs

Thus enable customers to smoothly transition to the Object TOOLKIT interfaces …

PTC Creo 3.0 – What's New …

PTC Creo UI Editor

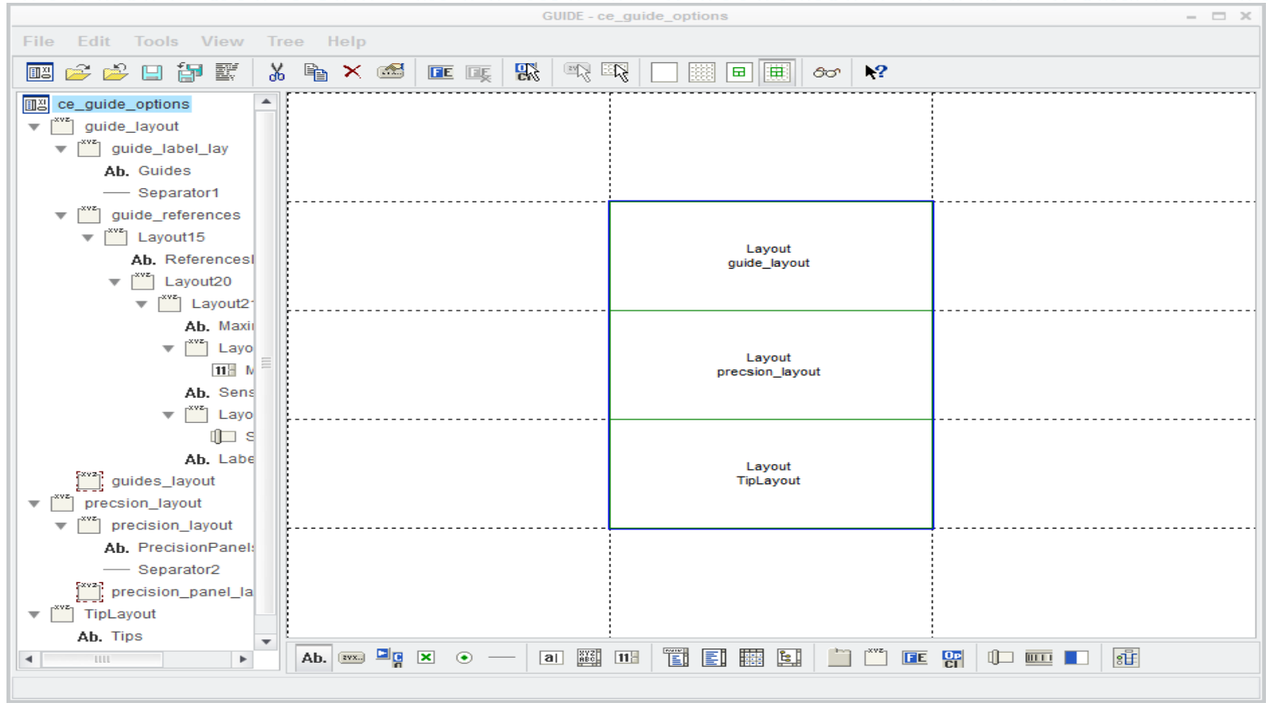
PTC® Live Global

The PTC Creo UI Editor enables you to design resource files graphically and preview the UI.

The PTC Creo UI Editor is divided into two main areas – a tree and a work area.

- The tree contains a hierarchical list of the components in the dialog.
- The work area contains a grid representation with each cell labeled according to its contents. The PTC Creo UI Editor also has a menu bar and a tool bar.

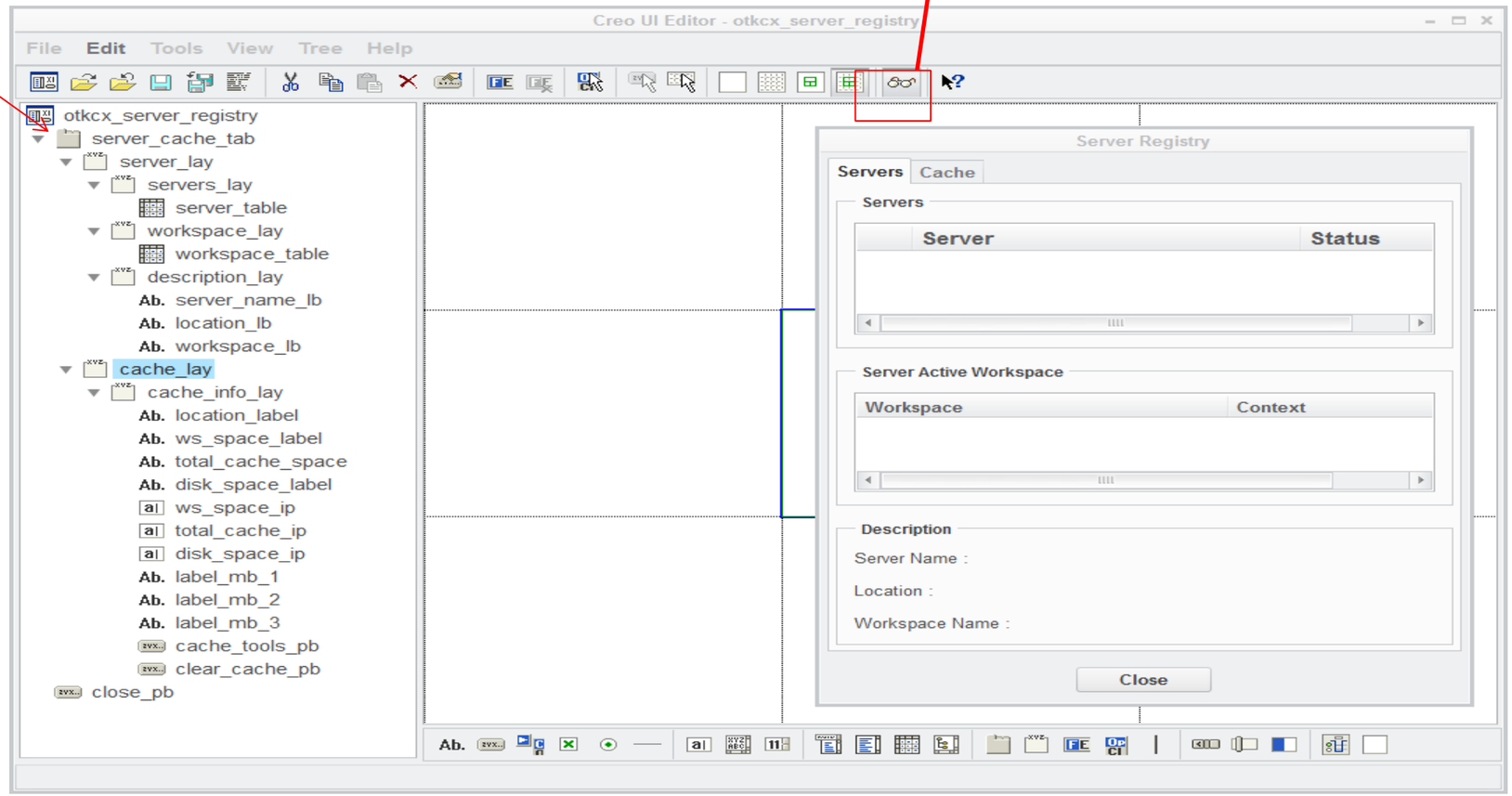
Creates the Resource file & Generates Code Stubs – either in Object TOOLKIT Java or C++



Included (free) with your PTC Creo Parametric TOOLKIT license.

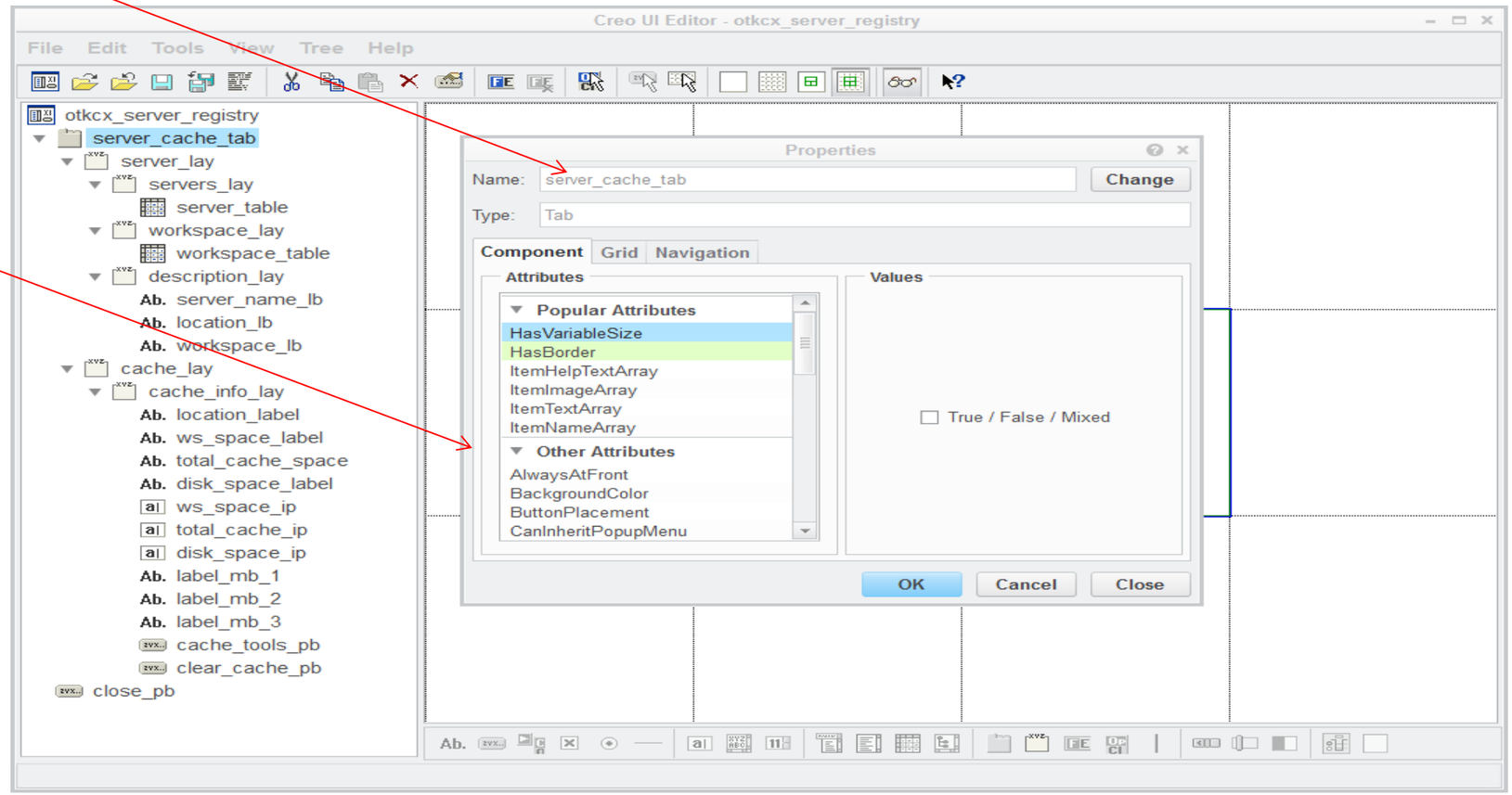
Design the layout of your dialog …

```
(Dialog otkcx_server_registry
   (Components
      (Tab                server_cache_tab
                          server_lay
                          cache_lay)
      (PushButton         close_pb)
   )
   (Resources
      (server_cache_tab.Decorated     True)
      (server_cache_tab.TopOffset     5)
      (server_cache_tab.BottomOffset  5)
      (server_cache_tab.LeftOffset    5)
      (server_cache_tab.RightOffset   5)
      (close_pb.Label                 "&Close")
      (close_pb.TopOffset             10)
      (close_pb.BottomOffset          10)
      (.Label                 "Server Registry")
      (.Rows                  32)
      (.Columns               38)
      (.Focus                 "server_table")
      (.BackgroundColor       3)
      (.CloseButton           False)
```

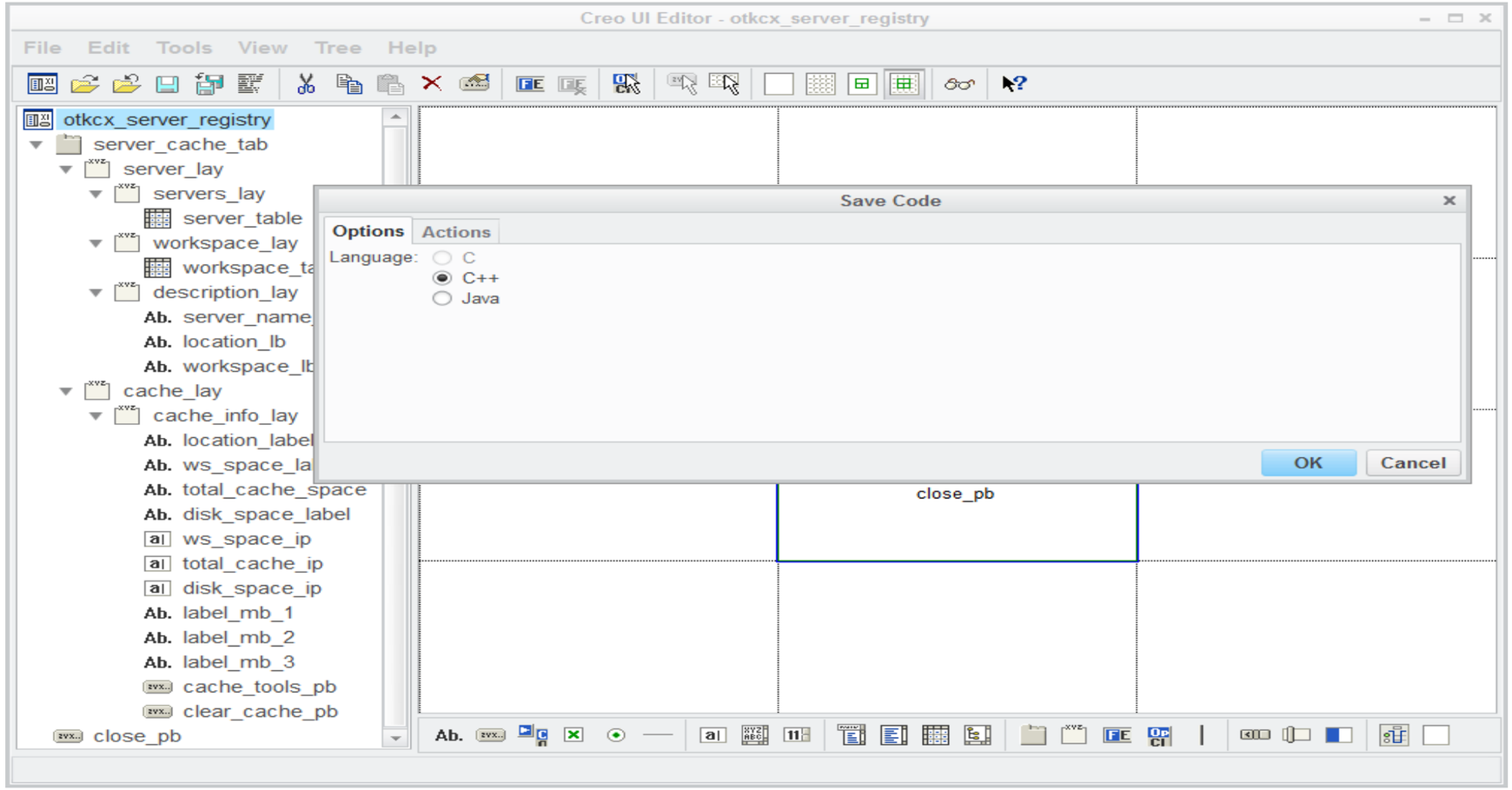


Customize the properties …

```
(Dialog otkcx_server_registry
   (Components
      (Tab                 server_cache_tab
                           server_lay
                           cache_lay)
      (PushButton          close_pb)
   )
   (Resources
      (server_cache_tab.Decorated     True)
      (server_cache_tab.TopOffset     5)
      (server_cache_tab.BottomOffset  5)
      (server_cache_tab.LeftOffset    5)
      (server_cache_tab.RightOffset   5)
      (close_pb.Label                 "&Close")
      (close_pb.TopOffset             10)
      (close_pb.BottomOffset          10)
      (.Label                         "Server Registry")
      (.Rows                          32)
      (.Columns                       38)
      (.Focus                         "server_table")
      (.BackgroundColor               3)
      (.CloseButton                   False)
      (.Layout
```



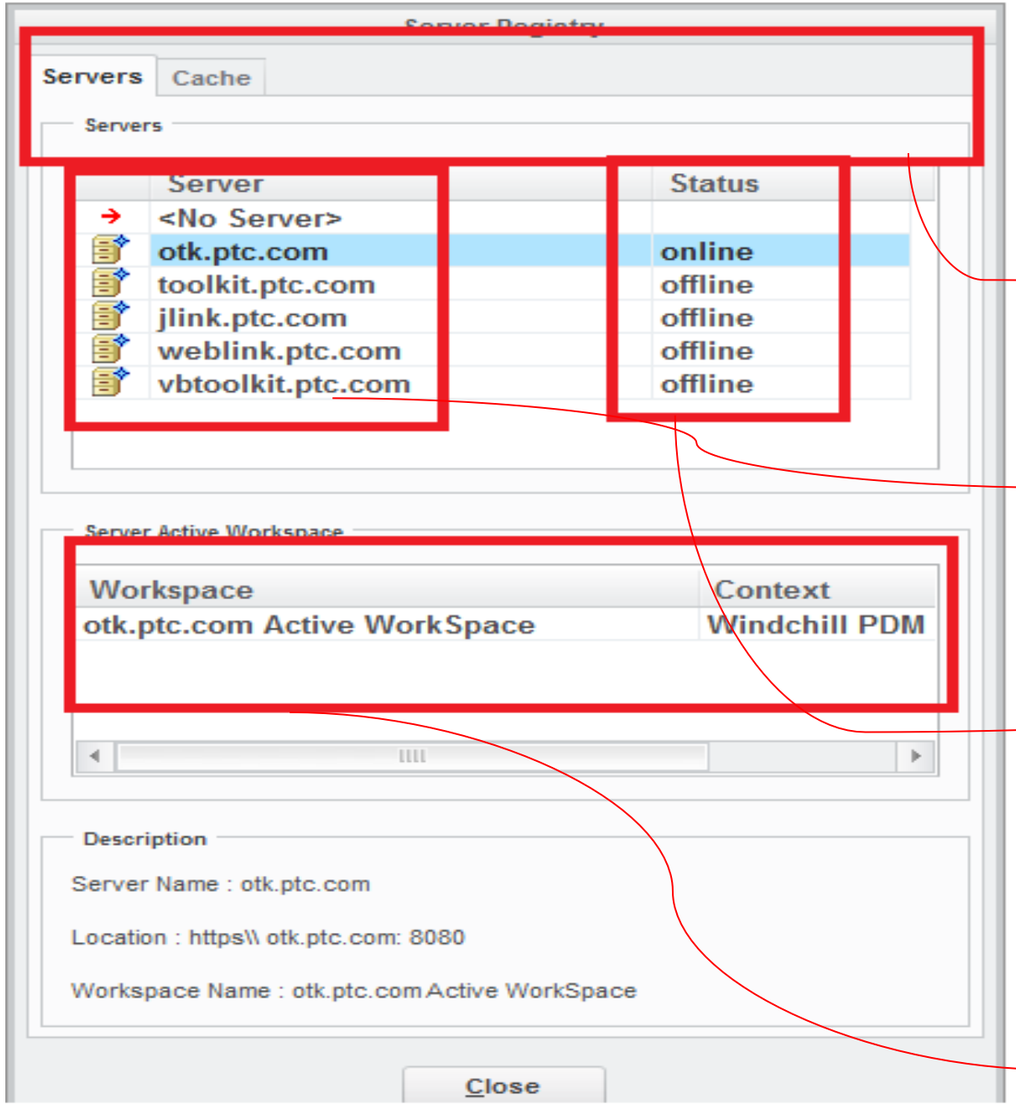
Generate code by clicking File ➔Save Code



Code can be generated with either C++ or Java bindings

for use with your Object TOOLKIT applications …

Java Object TOOLKIT – Implementing Callbacks …



Server-Cache Tab (code to switch between tabs)

```
If (handle.GetSelectedItemNameArray().get(0).matches("server_lay"))
pfcGlobal.GetProESession().UIDisplayMessage("TabSwitch.txt","Server Tab",null);

 else if (handle.GetSelectedItemNameArray().get(0).matches("cache_lay"))
pfcGlobal.GetProESession().UIDisplayMessage("TabSwitch.txt","Cache Tab",null);
```

Populating Server List in a Table with name of the server

```
TableCell serverCell = uifcTable.TableCellFind(serverTable.GetDialog(),
serverTable.GetComponent(), serverNames[ii], "server");
serverCell.SetText(serverNames[ii]);
```

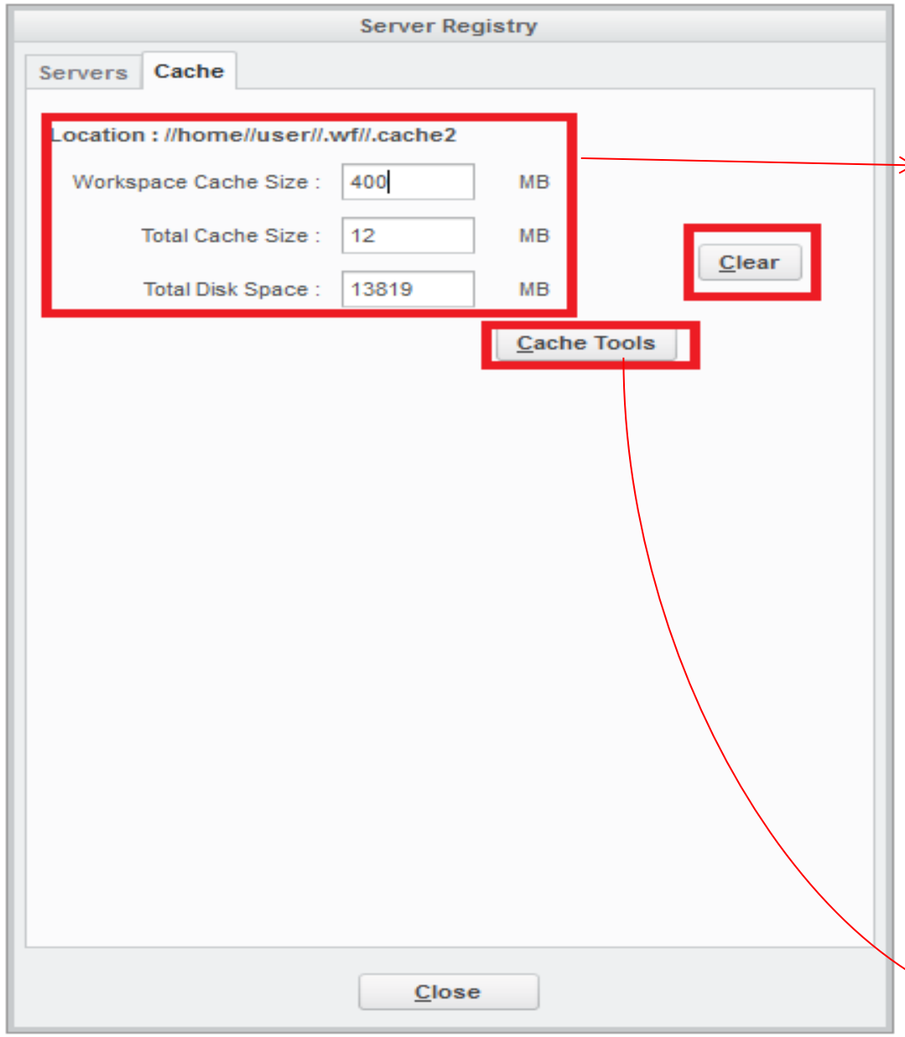
Populate server list in a Table with status of the server

```
TableCell statusCell = uifcTable.TableCellFind(serverTable.GetDialog(),
serverTable.GetComponent(), serverNames[ii], "status");
statusCell.SetText(serverStatus[ii]);
```

Fill workspace table with workspace name and context

```
TableCell contextCell = uifcTable.TableCellFind(workspaceTable.GetDialog(),
workspaceTable.GetComponent(),"workspace_row", "context");
contextCell.SetText("Windchill PDM");
```

Java Object TOOLKIT Implementing Callbacks …




```
(Set values into label - Server Location , Cache Space , Disk Space

Label cacheLoc = uifcLabel.LabelFind ( OTKServerRegistryDialog.OTK_SERVER_DIALOG,
"location_label");

cacheLoc.SetText("Location : //home//user//.wf//.cache2");

InputPanel wsIp = uifcInputPanel.InputPanelFind (
OTKServerRegistryDialog.OTK_SERVER_DIALOG, "ws_space_ip“ );

wsIp.SetIntegerValue(400);

InputPanel cacheIp = uifcInputPanel.InputPanelFind (
OTKServerRegistryDialog.OTK_SERVER_DIALOG, "total_cache_ip“ );

cacheIp.SetIntegerValue(12);

InputPanel diskIP = uifcInputPanel.InputPanelFind (
OTKServerRegistryDialog.OTK_SERVER_DIALOG, "disk_space_ip");

diskIP.SetIntegerValue(13819);
```

```
Action Listener for PushButton "Cache Tools"

OTKCacheTools listener = new OTKCacheTools();

PushButton clearPB = uifcPushButton.PushButtonFind (
OTKServerRegistryDialog.OTK_SERVER_DIALOG, "cache_tools_pb");
```

New Products – Java Object TOOLKIT

PTC Creo Object TOOLKIT Java – Why?

Why Object TOOLKIT Java?

- Many customers would like to do their customization in Java.
- J-Link provides only very limited capability – primarily only read capability – and hence these users are unable to leverage full benefit from automation.

Object TOOLKIT Java addresses this gap …

- Provides access to the full functionality available with Object TOOLKIT C++.
- Not just read APIs – but APIs to change the model contents, create features and do a host of things never possible with J-Link.
- Includes the PTC Creo UI Editor and supporting customizing other PTC Creo creation apps.
- Unlike J-Link, this is a licensed product …

PTC Creo Object TOOLKIT Java – Key Capabilities …

PTC® Live Global

- ***UI Customization***
  - *Create new dialogs, Add commands to the Creo ribbon, Add custom to code to standard Creo commands, Add context sensitive commands to the RMB menus*
- ***Session***
  - *Menus, Messages, Session level listeners, View and Window manipulation, Interactive Selection, Custom Graphics and text, Mouse input, Command line arguments*
- ***Models***
  - *Operations & Information, Import / Export 2D Formats, 3D Formats, Export Plot Files, Access External Data, Regeneration, Model Level Listeners, Cross Sections, Analysis, Layers, Family Tables, Relations, Parameters, Units*
- ***Part***
  - *Mass properties, Feature creation.*
- ***Assemblies***
  - *Component placement and constraints, Simplified representations, Exploded states, Interference / clearance calculations*
- ***Drawings***
  - *Views, Tables, Sheets, Notes, Entities, Groups, Symbols …*
- ***PTC Windchill Interaction***

PTC Creo Object TOOLKIT Java – Licensing

Key goals:

- Keep this API affordable for SMB customers.
- Provide upgrade path to J-Link users who need advanced functionality.

Hence, Object TOOLKIT Java is licensed as a Development License and a Runtime License.

- Development License will much lower than that for PTC Creo Parametric TOOLKIT.
- Runtime licensing can be license locked or floating
- Customers owning PTC Creo TOOLKIT licenses will be able to buy the Development license as an extension to their existing TOOLKIT licenses at a significant discount.

Questions?

PTC® Live Global

PTC® PRODUCT & SERVICE ADVANTAGE®

PTC Creo 3.0 – Key Technical Changes

Agenda

Key Technical Changes:-

- Compiling and Linking Applications
- Migration Tools
- Pro/DEVELOP Retirement
- Support for PTC Creo Unite technology
- Other key technical changes
  - New Layout Model Type
  - Registry File related changes
  - RMB Shortcut related changes
- Other Considerations
  - Multi-threading APIs
  - Digital Rights Management

Key Technical Changes - Compiling and Linking

All PTC Creo Parametric TOOLKIT applications on 32-bit and 64-bit Windows platforms must be built using the Microsoft Visual Studio 2012 compiler.

- In Visual Studio, set the configuration property Platform Toolset as Visual Studio 2012 - Windows XP (v110_xp).
- To use the makefile based solution, you must add the flag -D_USING_V110_SDK71_ to the compile line of each object.
- These changes have been incorporated in the sample makefiles.

Standard Libraries

- new libraries protk_dll_NU.lib, ucore.lib, and udata.lib have been added.
- The library protk_dll.lib has been deprecated and may not be supported in future releases.

PTC recommends rebuilding your PTC Creo Parametric TOOLKIT
applications with the new libraries.

Key Technical Changes – Migration Tools

The migration tool ptk_revtool has been retired, and will not be supported in future releases.

A new perl script mark_deprecated.pl is being provided to assist in updating applications from older versions to the current release. This script will:-

- Insert comments into your Toolkit Application code for all the superseded APIs used in your application code.
- In case a replacement API is available, the name of that API would be included in the comment.
- The list of superseded APIs are available in file [Toolkit Load Point]\scripts\protkmap.txt

To check whether which release the API was deprecated, please refer to the API Wizard.

Pro/DEVELOP Retirement

Key Change:-

- PTC will NO longer support Pro/Develop APIs from PTC Creo 3.0
- Pro/Develop headers, binaries, examples, makefiles will NO longer be available on PTC Creo 3.0

Action Required:-

- All apps being built on Creo 3.0 need to be updated to use equivalent Creo TOOLKIT [or C++ Object TOOLKIT] APIs
- Apps built on Creo 2.0 and earlier versions using Pro/Develop APIs may NOT work correctly – it is strongly recommended that these be updated to use TOOLKIT APIs for PTC Creo 3.0.

Please inform Tech Support for any Pro/Develop API for which you do not find an equivalent Toolkit API.

Key Technical Change – Support for PTC Creo Unite

PTC Creo Unite enables you to open non PTC Creo parts and assemblies in PTC Creo Parametric without creating separate PTC Creo models.

PTC Creo Unite functionality will be available in a maintenance release of PTC Creo 3.0.

The PTC Creo Parametric TOOLKIT interfaces that support multi-CAD assemblies are available in the PTC Creo Parametric TOOLKIT 3.0 F000 release.

Key Changes:-

Existing size of ProName is insufficient to accommodate names of non-native models. Hence, all existing APIs and notifications using ModelNames will be deprecated and new APIs are introduced.

- PTC will continue to support these existing APIs with purely native documents for Creo 3.0
- *However, these APIs will NOT be supported with documents with non-native content [due to above technical limitation*].
- These deprecated APIs will be obsoleted in Creo 4.0.

Note: Creo Object TOOLKIT interfaces dealing with part / assembly modelnames remain the same - hence, minimal to no changes will be required to update Object TOOLKIT applications.

Update your apps now – to benefit from the PTC Creo Unite technology

Key Technical Changes – Registry File

Object TOOLKIT (C++ and Java) are being extended to support other Creo apps – starting with PTC Creo Direct.

Following new fields are introduced in the registry file:-

- **rbn_path**: Allows you to specify the name and path of the ribbon file, which must be loaded when you open PTC Creo.
- **toolkit**: protoolkit (default)/object. Use protoolkit for applications based on PTC Creo Parametric TOOLKIT or object for applications based on PTC Creo Object TOOLKIT C++ or Java.
- **creo_type**: parametric (default)/direct. Valid only for Object TOOLKIT Applications. Specifies the Creo App type in which this application will be run. Use direct for PTC Creo Direct.

Switch to Object TOOLKIT C++ or Java to benefit from application re-use across Creo apps.

New ModelType for Layout Models

Key Changes:-

- A new ModelType has been introduced for Layout models [PRO_MDL_CE_SOLID].
- This change is necessitated to support the new functionality being added to Layout models
- An Assembly can now consist of 3 types of models - Part, Assembly, Layout.

Apps which do specific processing based on ModelType should be updated to factor in this change.

Key Technical Changes - RMB Shortcut Menus

Key Changes:-

- In Creo 3.0, RMB shortcut menus are being made object aware.
- End-users will be able to customize these thru' the Ribbon Customizer→ShortCuts
- Current TOOLKIT RMB related APIs [ProPopupmenuButtonAdd etc…] will however continue to be supported with legacy behavior.
  - *However, these would NOT be customizable by the end-user thru' the Ribbon Customizer→ShortCuts*

Action Required:-

- NONE

Key Technical Changes – Other Considerations

Multi-threading Support

- From PTC Creo Parametric 3.0 onward, Multithreading is now always supported in PTC Creo Parametric TOOLKIT applications.
- The function ProEngineerMultithreadModeEnable() has hence been deprecated.

Digital Rights Management

From PTC Creo Parametric 3.0 onward, Digital Rights Management (DRM) is no longer supported. Applications that check the DRM permissions will have to be updated.

Customized Ribbon User Interface

- Customized Ribbon User Interface components built in Creo Parametric 1.0 F000 or M010 may not work correctly in certain situations
- Please rebuild these ribbon customizations again using Creo Parametric 1.0 M020 and higher releases or send these to PTC for review and clean-up.

- Your feedback is valuable
- Don’t miss out on the chance to provide your feedback
- Gain a chance to win an instant prize!
- Complete your session evaluation now

PTC® Live Global

PTC® PRODUCT & SERVICE ADVANTAGE®