An OpenTD Tour

Presented at TFAWS 2022

by Matt Garrett

www.crtech.com

C&R TECHNOLOGIES, INC.
Phone 303.971.0292
Fax 303.971.0035

Contents

- What is OpenTD?
- OpenTD Project Hub
- Thermal Desktop Bonus Apps
- Tips and Tricks
- Selected New Features in 6.3

C&R TECHNOLOGIES
www.crtech.com

What is OpenTD?

- Application Programming Interface (API)
  - ✓ Allows you to write programs to control TD with and without human interaction.
- Installed with TD 6.0 and above
- Accessible with any language that can talk to .NET
  - ✓ C#, MATLAB, Python, Powershell, VB.NET, F#, etc.
  - ✓ We support C#.
- Version-controlled
  - ✓ Your 6.3 programs won't break when we release 6.4.
- Documented
  - ✓ Getting Started guide and Class Reference installed with TD
  - ✓ Additional examples on CRTech forums
  - ✓ Videos on the CRTech website

C&R TECHNOLOGIES
www.crtech.com

What can you do with OpenTD?

- Connect your program to TD
  - ✓ Attach to already-running instance or start new TD
  - ✓ Connect to one or many instances of TD simultaneously
- Create an add-in that runs within TD
- Create, query, edit, delete most TD entities
  - ✓ Nodes, conductors, heat loads, etc.
  - ✓ FD surfaces and solids
  - ✓ FE meshes
  - ✓ Lumps, paths, ties, etc.
  - ✓ Case sets, orbits
  - ✓ Thermophysical and optical properties
  - ✓ Much more!
- Work with units, symbols, and expressions
- Interact with the GUI to get or set AutoCAD selection sets
- Run cases, and control them during the run
- Explore, compare and plot results, including as-solved model topology

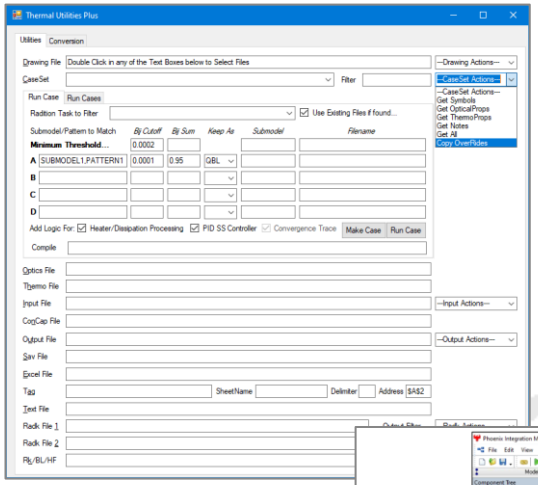
C&R TECHNOLOGIES
www.crtech.com

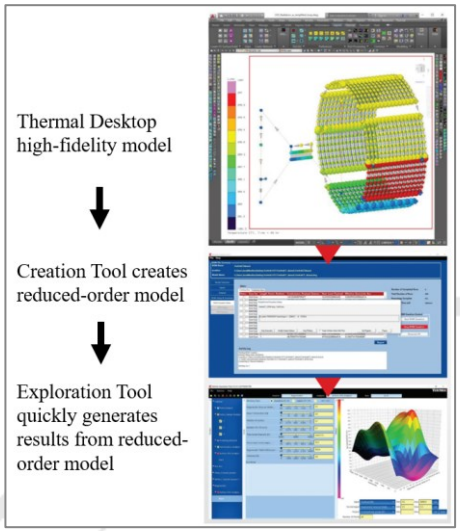
OpenTD Project Hub

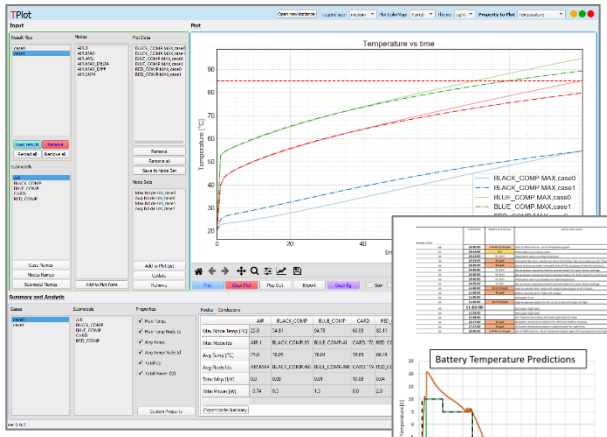
C&R TECHNOLOGIES
www.crtech.com

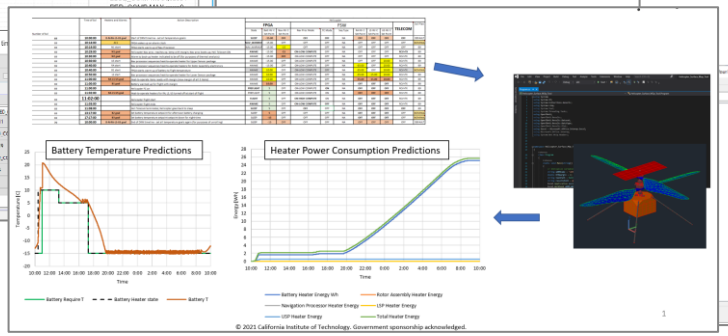
OpenTD Project Hub

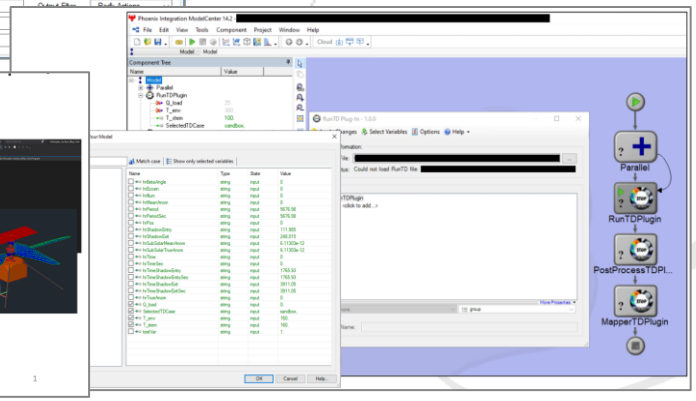
- For TFAWS last year, we asked for “Real-World OpenTD Applications”
  - ✓ Overwhelmed by response











C&R TECHNOLOGIES
www.crtech.com

OpenTD Project Hub

- Several customer requests for a place to post OpenTD binaries, projects, code snippets, success stories, etc.
- We created the OpenTD Project Hub as a place to share useful OpenTD tools
- Build on each other's knowledge and experience
  - ✓ Discover new techniques

C&R TECHNOLOGIES
www.crtech.com

OpenTD Project Hub

Forums » Product-specific Discussions » OpenTD

OpenTD Project Hub

This is a place to share useful tools you've created using OpenTD, from projects with full source code, to binaries or code snippets., to real-world success stories. You can also share real-world success stories even if you can't provide any code or specifics. Our community is full of creative, smart people; let's work together to build even more interesting tools.

POST A NEW TOPIC

SUBSCRIBE TO FORUM

TOPICS	STATISTICS	LAST POST
Real-World OpenTD Applications by **MattGarrett** » Fri Jun 03, 2022 @ 3:18pm	Replies: **1**	by **VictorS** Mon Jul 25, 2022 @ 1:28pm
Template Launcher: Consistently set layers, submodels, rad groups, and more by **Mark Schmidt** » Tue Jun 07, 2022 @ 4:55pm	Replies: **0**	by **Mark Schmidt** Tue Jun 07, 2022 @ 4:55pm
CRTest: write scripts to run models and compare results by **MattGarrett** » Fri Jun 03, 2022 @ 3:49pm	Replies: **0**	by **MattGarrett** Fri Jun 03, 2022 @ 3:49pm
Dataset Viewer: peek inside save files or CSR's by **MattGarrett** » Fri Jun 03, 2022 @ 3:05pm	Replies: **0**	by **MattGarrett** Fri Jun 03, 2022 @ 3:05pm

Your OpenTD tool here?

https://crtech.com/forum/1830

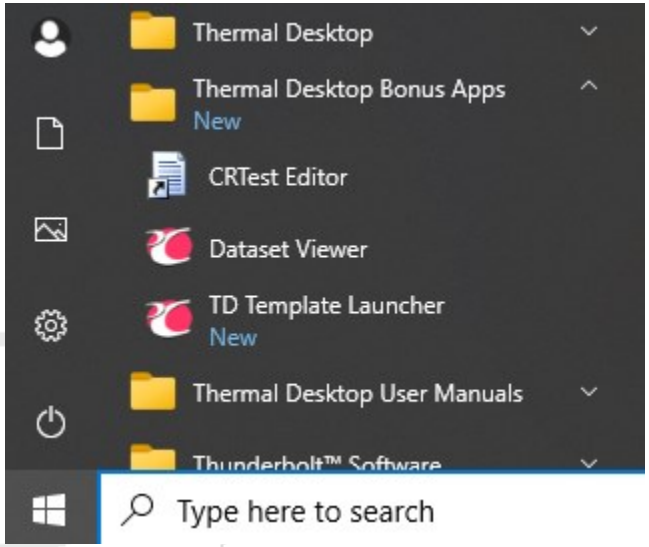
C&R TECHNOLOGIES
www.crtech.com

Thermal Desktop Bonus Apps

C&R TECHNOLOGIES
www.crtech.com

Thermal Desktop Bonus Apps

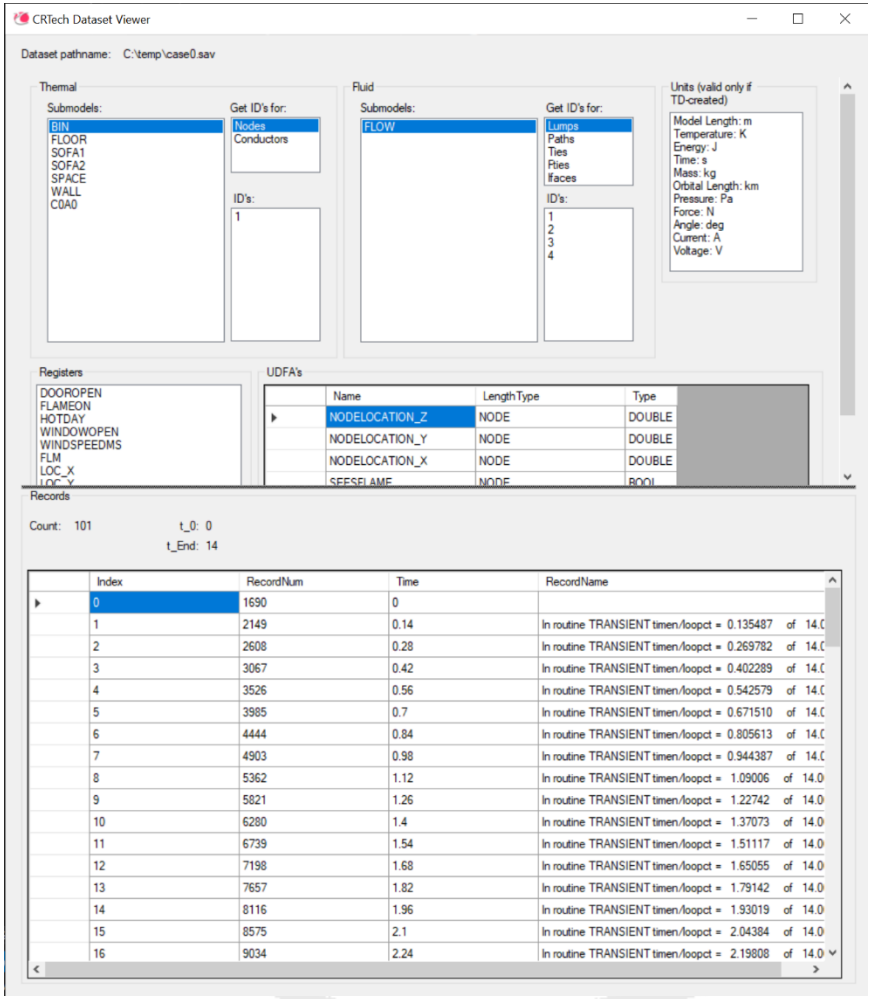
- CRTech posted internal utilities to OpenTD Project Hub
  - ✓ Dataset Viewer
    - ✓ Peek into save files or CSR's
  - ✓ Template Launcher
    - ✓ Quickly start TD with standard templates
  - ✓ CRTest
    - ✓ Write scripts to run models and compare results using a simple imperative language



C&R TECHNOLOGIES
www.crtech.com

Dataset Viewer

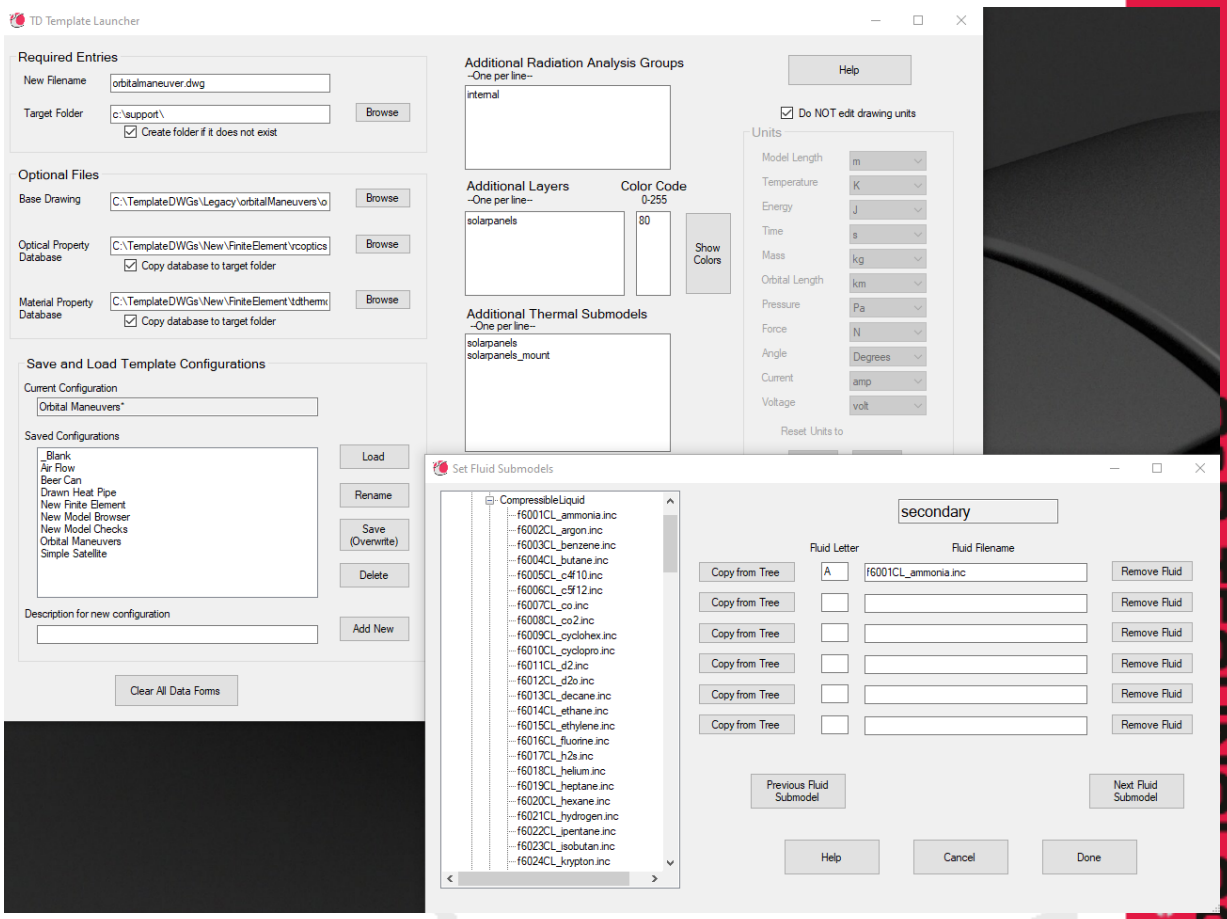
- Quick peek into save files or CSR's
- No plotting, just basic info



C&R TECHNOLOGIES
www.crtech.com

TD Template Launcher

- Collects commonly-used dwg setup tasks in one place.
- Start from a template dwg if desired
- Copy or link to property db's
- Save/reload useful configs

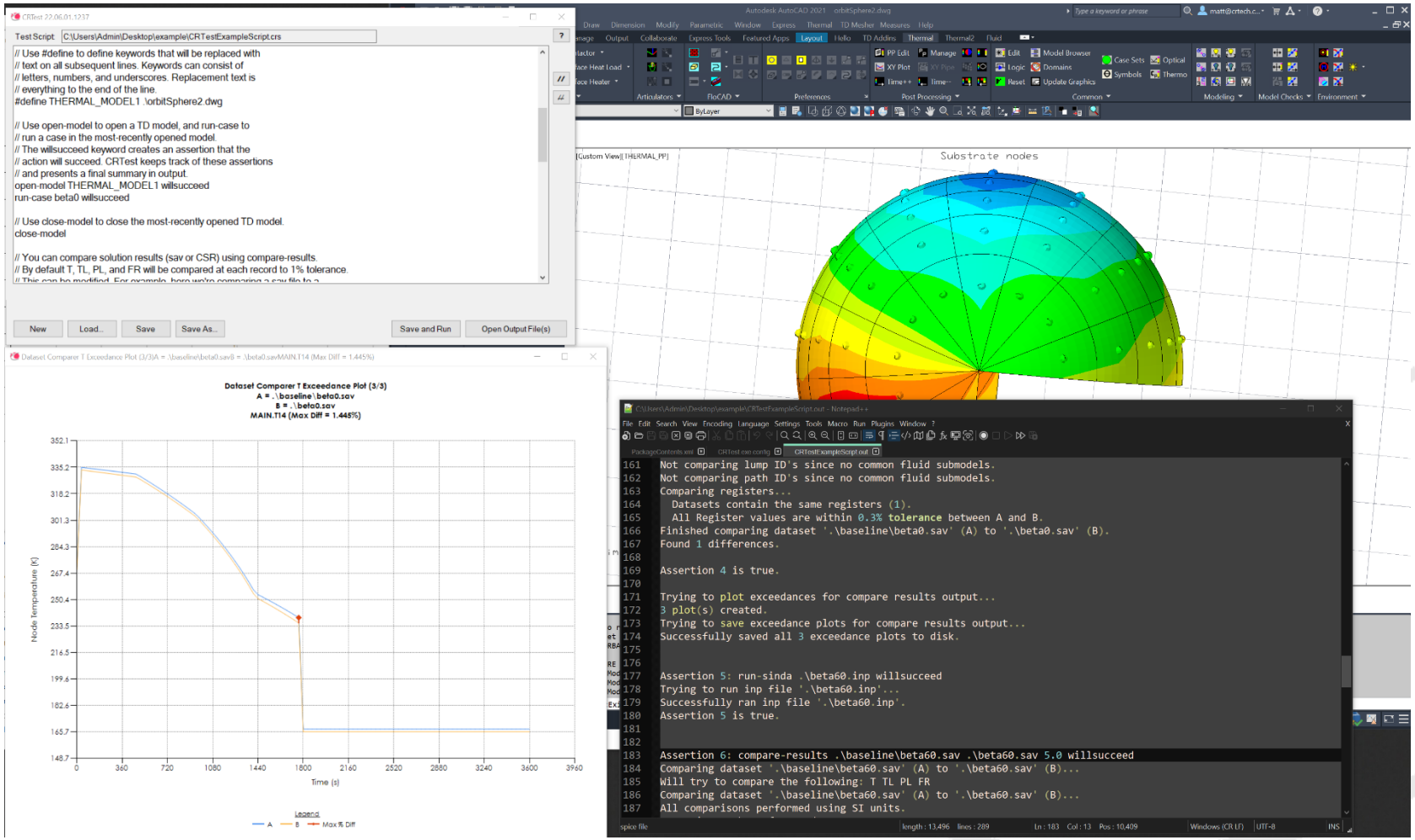


CRTest

- CRTech's QA group uses CRTest to perform acceptance testing of new versions of TD and S/F
- Imperative syntax, not object-oriented
  - ✓ Open and close dwg's
  - ✓ Run cases
  - ✓ Run S/F inp files
  - ✓ Compare results
  - ✓ Manipulate files and directories
  - ✓ System commands
- Subset of OpenTD, using imperative syntax that is familiar to many engineers

C&R TECHNOLOGIES
www.crtech.com

CRTest



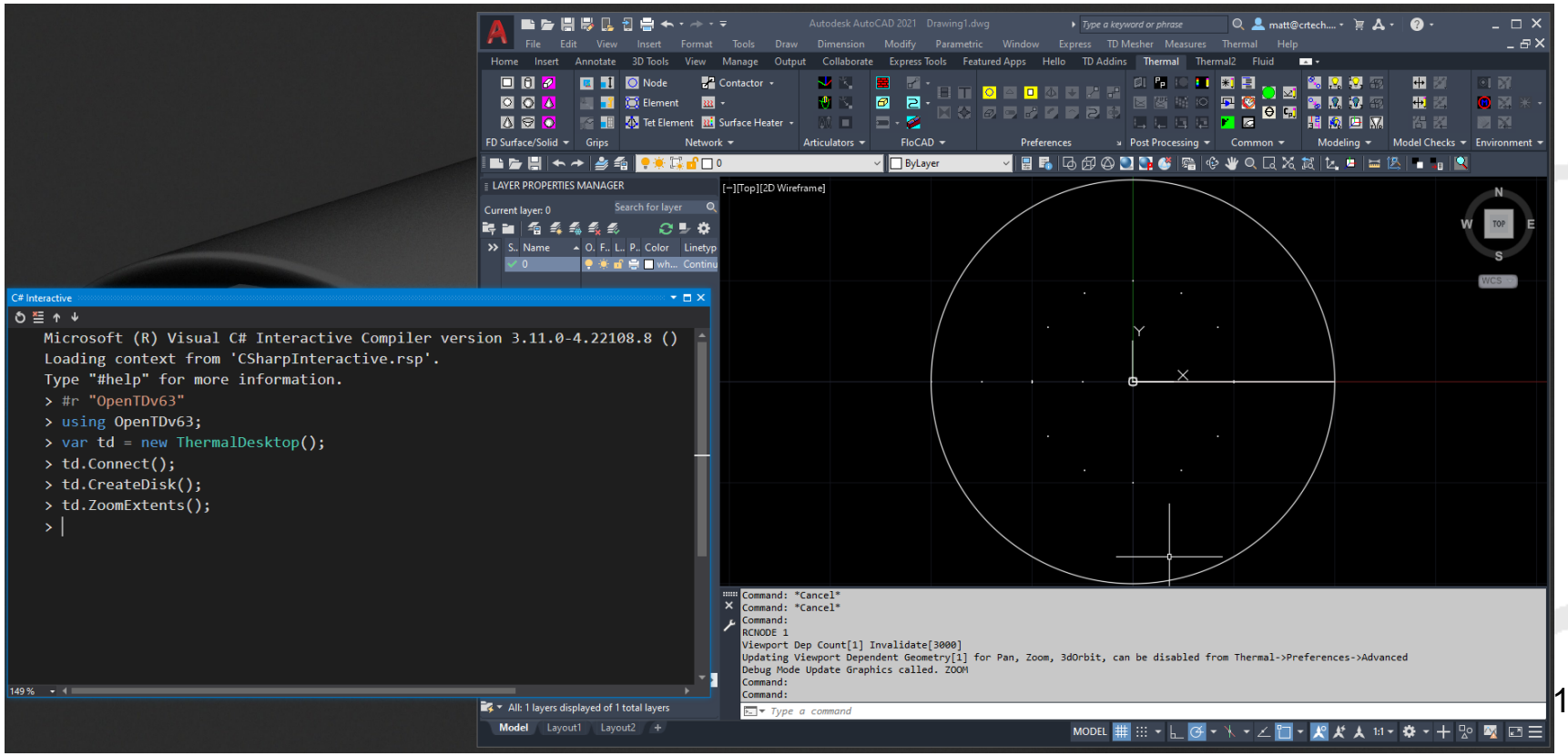
C&R TECHNOLOGIES
www.crtech.com

Tips and Tricks

C&R TECHNOLOGIES
www.crtech.com

Use C# Interactive Compiler

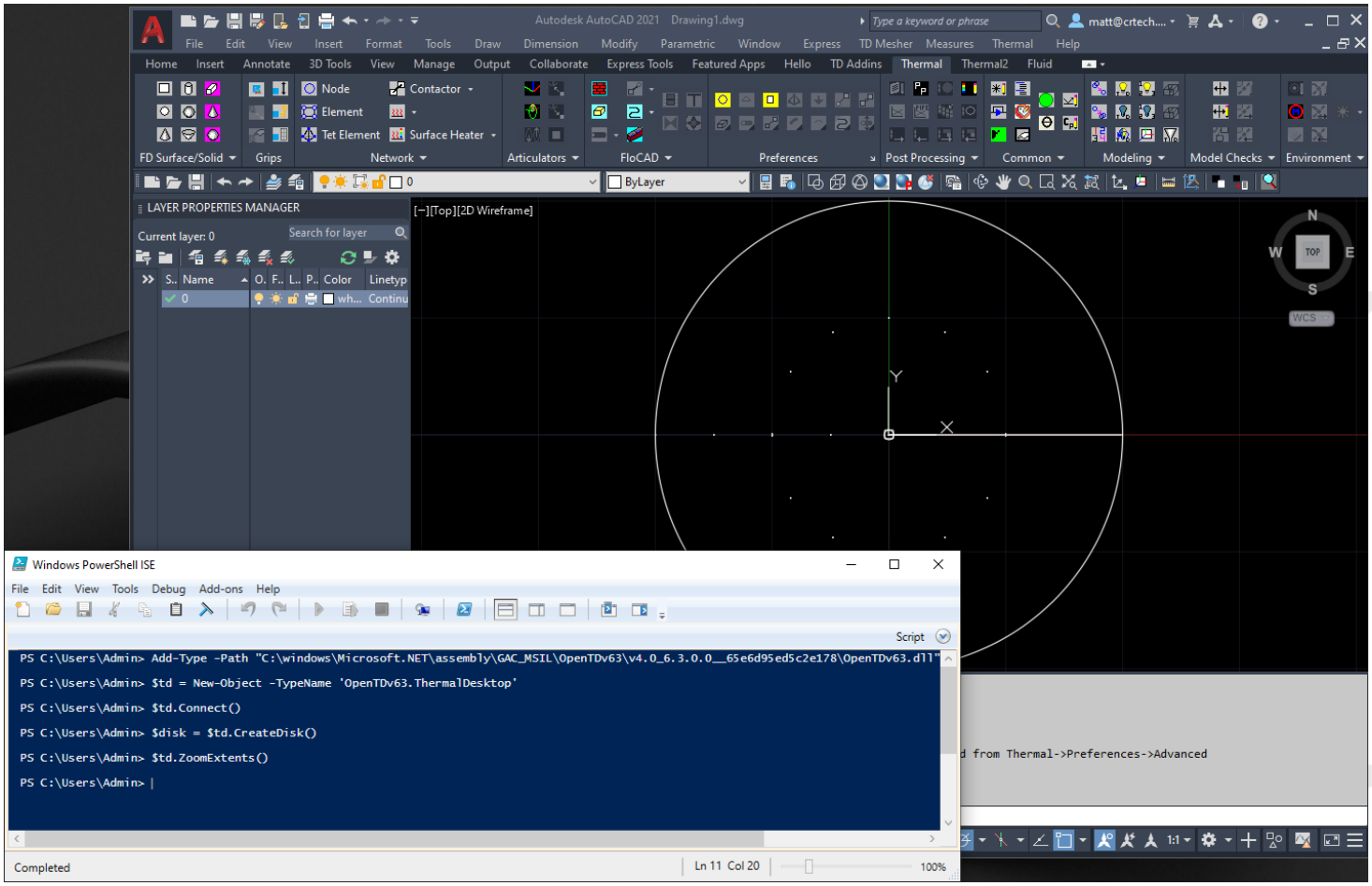
- You can run OpenTD code without compiling in Visual Studio
- View->Other Windows->C# Interactive
- Use *#r "OpenTDv63"* to load a dll, then it's business as usual



C&R TECHNOLOGIES
www.crtech.com

Use Powershell

- Powershell is a free alternative to Visual Studio for interacting with OpenTD
- https://crtech.com/forum/topic/use-opentd-powershell



C&R TECHNOLOGIES
www.crtech.com

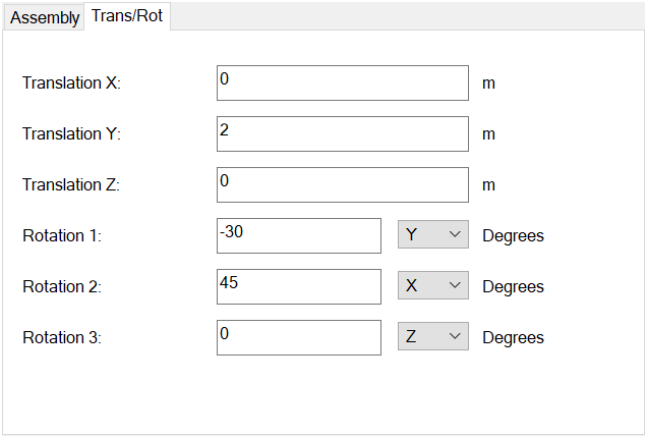
BaseTrans and LocalTrans

- Used to position geometry
- BaseTrans is a *Matrix3d* equivalent to positioning geometry using AutoCAD commands
- LocalTrans is a *Transformation* equivalent to positioning geometry using the “Trans/Rot” tab in the GUI
- OpenTD positions entities at the WCS origin, then transforms them by BaseTrans, followed by LocalTrans
- https://crtech.com/forum/topic/using-basetrans-and-localtrans

C&R TECHNOLOGIES
www.crtech.com

Using LocalTrans

- LocalTrans rotations are *intrinsic*, they are about the entity's current axes, not fixed axes
- Example:
  - ✓ Translate 2 m along the entity's Y axis
  - ✓ Rotate -30 deg about the entity's Y axis
  - ✓ Rotate 45 deg about the entity's new X axis




```
a.LocalTrans.Ty = 2;
a.LocalTrans.Axis1 = 1; // Y
a.LocalTrans.Rot1 = -30;
a.LocalTrans.Axis2 = 0; // X
a.LocalTrans.Rot2 = 45;
a.Update();
```


C&R TECHNOLOGIES
www.crtech.com

Using BaseTrans

- Don't try to chain rotations using a single Matrix3d and multiple calls to SetToRot* methods
  - ✓ Each cancels the last
- Instead, use matrix multiplication
- Example (with intrinsic rotations):
  - ✓ 10 deg rotation about an entity's Z axis
  - ✓ 2 m translation along its current X axis
  - ✓ 50 deg rotation about its current X axis

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
var A = new Matrix3d().SetToRotZ(10);
var T = new Matrix3d();
T.SetOrigin(new Point3d(2, 0, 0));
var B = new Matrix3d().SetToRotX(50);
a.BaseTrans = A * T * B;
a.Update();
```


C&R TECHNOLOGIES
www.crtech.com

Connections and Domains

- Connection consists of Handle and Marker
  - ✓ Handle identifies an entity
  - ✓ Marker identifies how the entity is connected. Example, marker values for FD Brick:

Decimal	Binary	Applied Surfaces					
1	0b000001						XMIN
42	0b101010	ZMAX		YMAX		XMAX	
63	0b111111	ZMAX	ZMIN	YMAX	YMIN	XMAX	XMIN

- Special case: if Marker == -999, then Connection refers to a Domain and the Handle string is the domain name

C&R TECHNOLOGIES
www.crtech.com

Connections and Domains Example

```
// apply heat load to XMIN and YMIN of an FD Brick:
var brick = td.CreateSolidBrick();
var heatLoad = td.CreateHeatLoad(brick);
heatLoad.AppliedType = RcHeatLoadData.AppliedTypeBoundaryConds.SURFACE;
heatLoad.ApplyConnections[0].Marker = 0b000101;
heatLoad.Name = "q applied to brick";
heatLoad.Update();

// apply a heat load to a domain:
var domainConnections = new List<Connection>();
// ...
// add connections to domainConnections
// ...
td.GetDomainManager().CreateDomain
    ("HEATED", DomainType.NODESET, domainConnections);
var rectHeatLoad = td.CreateHeatLoad(
    new Connection("HEATED", -999));
rectHeatLoad.Name = "q applied to HEATED domain";
rectHeatLoad.Update();
```


C&R TECHNOLOGIES
www.crtech.com

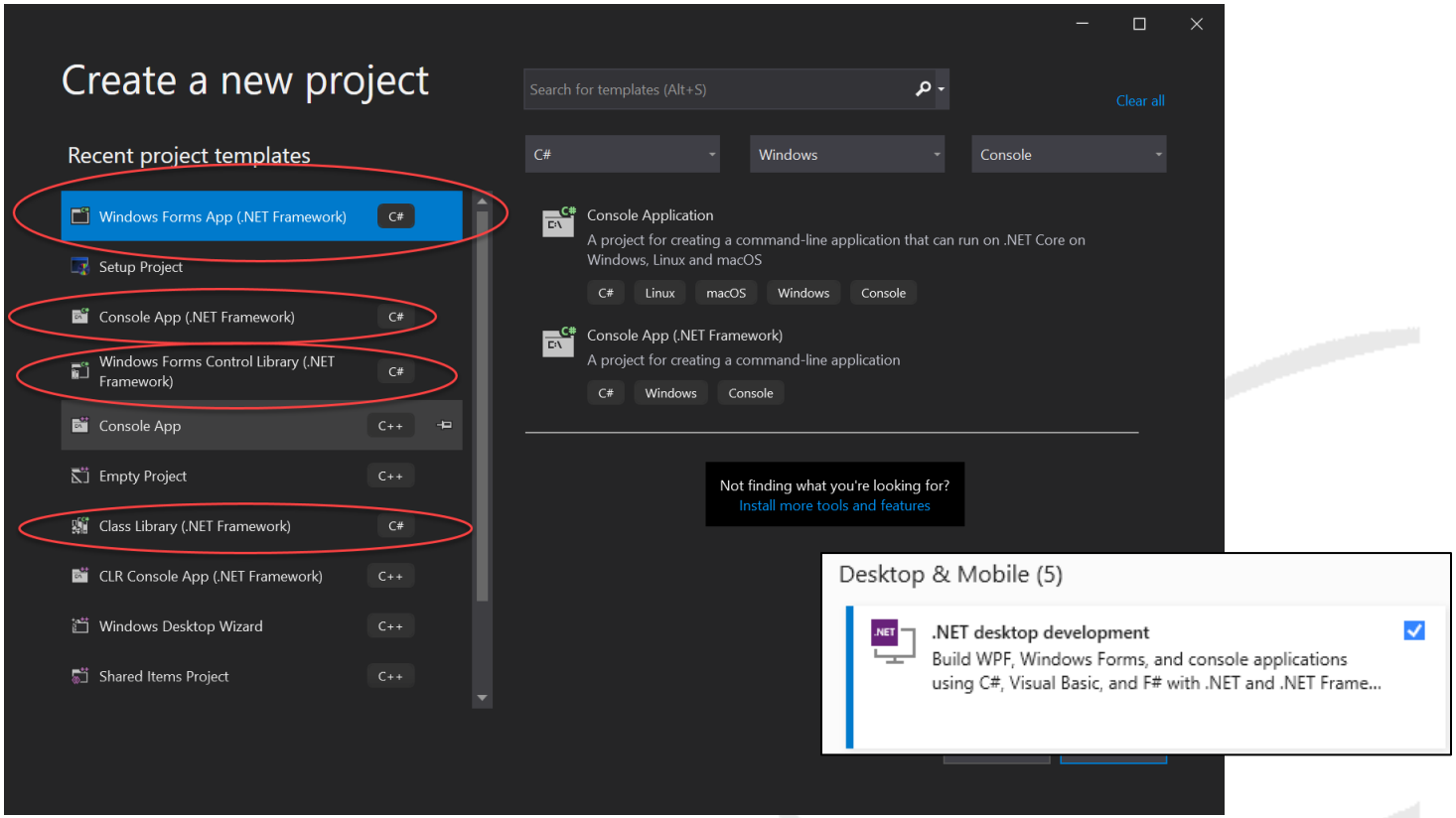
Use Select or Linq Query Syntax to Transform Lists

```
// To get a list of node handles in a model, you might do this:
{
    var nodes = td.GetNodes();
    var nodeHandles = new List<string>();
    foreach (var n in nodes)
        nodeHandles.Add(n.Handle);
}

// Using Select you can do it in one line:
{
    var nodeHandles = td.GetNodes().Select(n => n.Handle);
}

// Construct queries using Linq Query Syntax:
{
    var nodeHandlesRightSide = from n in td.GetNodes()
                               where n.Origin.X > Xmid
                               orderby n.Origin.X
                               select n.Handle;
}
```


C&R TECHNOLOGIES
www.crtech.com

Use .NET Framework (not Core or Standard)

- Example project templates:



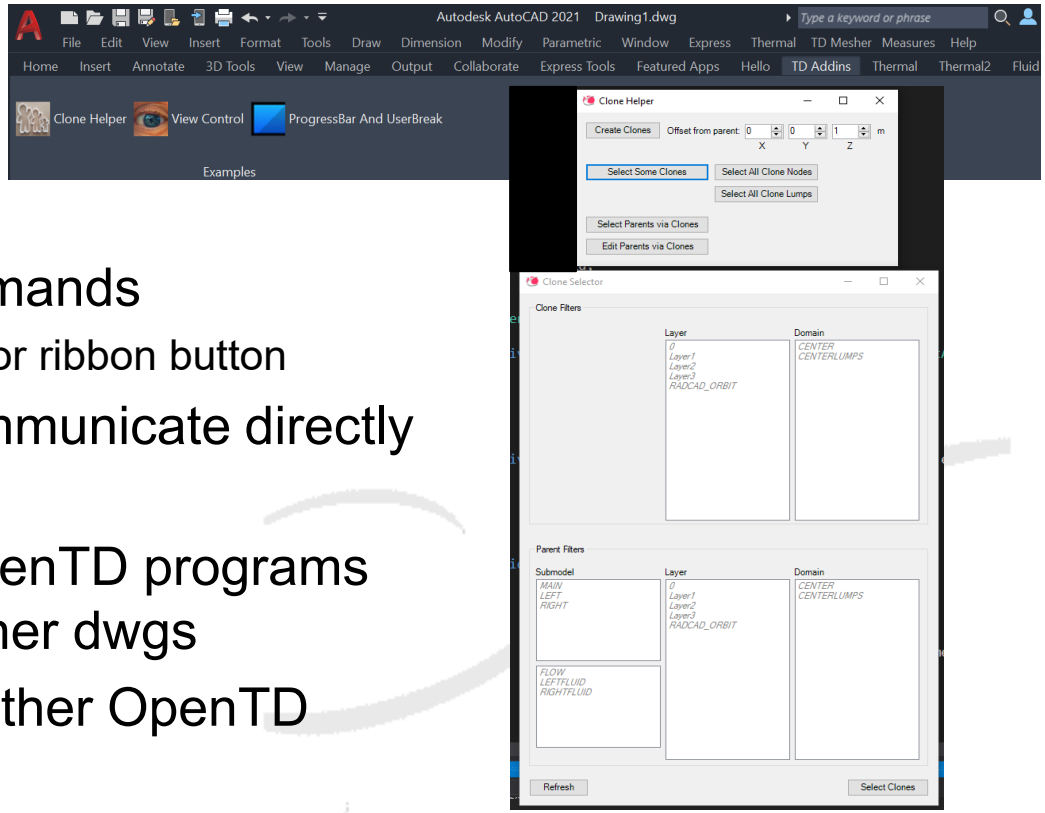
- https://crtech.com/forum/topic/finding-net-framework-visual-studio-2019

C&R TECHNOLOGIES
www.crtech.com

Selected New Features in 6.3

C&R TECHNOLOGIES
www.crtech.com

Addins



- Create your own TD commands
  - ✓ Launch with command line or ribbon button
- Addins run within and communicate directly with TD, not inter-process
- Can do anything other OpenTD programs can do, including open other dwgs
- Use the same syntax as other OpenTD programs
- https://www.crtech.com/forum/topic/create-your-own-td-add-version-63

C&R TECHNOLOGIES
www.crtech.com

Working with End Users

- OpenTD 6.3 contains several features for interacting with end users
  - ✓ Especially useful for Addins, but can be used in any OpenTD program
- New Features:
  - ✓ Get/Set/ClearSelection (partially added mid-release in 6.2)
    - ✓ Control the TD selection set
    - ✓ Prompt end users to select entities on-screen or from the Model Browser
  - ✓ ProgressBar
    - ✓ Control the same progress bar used by RadCAD during calculations
  - ✓ IUserBreak (for Addins only)
    - ✓ Allow the end user to stop long operations by pressing ESC
  - ✓ UCS
    - ✓ Get or set the TD user coordinate system
  - ✓ Show/HideModelBrowser
  - ✓ Enable/DisableUndo
  - ✓ GetMainWindow
    - ✓ returns an object representing the TD window; use it as the owner for Addin dialogs

C&R TECHNOLOGIES
www.crtech.com

Logging

- Internally, Thermal Desktop Suite uses a flexible logging system to write messages to log files at runtime
- This system is now available in the OpenTDv63.Logging namespace

1. Create a logger for your class:

```
private static readonly Logger log
    = LoggerFactory.GetLogger(typeof(MyClass).ToString());
```

2. Send a message from the logger:

```
log.LogWarn($"Operation number {opNum} did not complete: {warnMsg}");
```

3. At runtime a config file determines routing:

```
<?xml version="1.0" encoding="utf-8"?>
<CRLogging>
  <!-- Use this file to configure where OpenTD will send logging messages.
       To make OpenTD use it, copy it to %localappdata%\ThermalDesktop
       and restart OpenTD. -->
  <Loggers>
    <!-- Loggers dispatch messages to their listeners. Logger names correspond to names used
         within OpenTD to categorize messages. Use the RootLogger (below) to configure all loggers
```

4. And the logging system writes to log files according to config:

```
20220419 15:13:59 (1/21664) - Warning: MyLibrary.MyClass: Operation number 145 did not complete: matrix2 was null
```

C&R TECHNOLOGIES
www.crtech.com

CoSolver now more Flexible

- CoSolver namespace introduced in 6.2
  - ✓ Could run a TD case and connect to the SINDA/FLUINT run
- For 6.3, added two new classes:
  - ✓ SF_Launcher can run SINDA/FLUINT models directly from the inp file
  - ✓ SF_CoSolver runs inp files and connects to the SINDA/FLUINT run

```
// If you'd just like to run a SINDA/FLUINT model, use an SF_Launcher:
string justRunPath = Path.Combine(workingDir, "justRun.inp");
var justRunLauncher = new SF_Launcher(justRunPath);

// SF_Launcher.Run() creates a new process to run the model. Use WaitForExit()
// if you'd like to wait for it to finish before proceeding:
Console.WriteLine("Running S/F model '" + justRunPath + "'...");
var justRunProcess = justRunLauncher.Run();
justRunProcess.WaitForExit();
Console.WriteLine("Done.");
```

C&R TECHNOLOGIES
www.crtech.com

Access Legacy TD COM Interface

- Before OpenTD, there was a limited API using COM
- For completeness the old API can now be accessed with SendLegacyCommand
- Getting starting guide contains full list of commands

Sample commands:

- **postprocess filename [optional delay in ms]**
  Creates or updates a TEXT-type dataset from a file. New applications should use PostProcessing.DatasetManager instead.
- **ppnexttime**
  Steps forward to the next record in the current dataset. New applications should use PostProcessing.DatasetManager instead.
- **ppsavefile savOrCsrPathname**
  Creates or updates a SF-type dataset from a save file or CSR directory. New applications should use PostProcessing.DatasetManager instead.
- **ppsettime index**
  Sets the record for the current post-processing dataset. Uses 0-indexed record array index (0, 1, 2, etc) not record number. New applications should use PostProcessing.DatasetManager instead.

C&R TECHNOLOGIES
www.crtech.com

All New Features

- **Major New Features**
- added Addin capability
- added support for Measures
- added full support for global dwg preferences
- added support for Post Processing Data Mappers
- added support for legacy TD COM interface via ThermalDesktop.SendLegacyComCommand method
- added SF_Launcher for running text-based input models in SINDA/FLUINT
- added SF_CoSolver for running and controlling text-based input models in SINDA/FLUINT
- added access to Logging namespace, allowing client programs to use CRTech's rich logging features
- added OpenTDMixedInterface.dll, providing access to some OpenTD features via C/C++/Fortran

- **Other New Features**
- added GetSymbolValue method to return an evaluated symbol value from TD
- added Show/HideModelBrowser methods
- added support for getting and setting the UCS
- added ProgressBar class to control RadCAD progress bar
- added IUserBreak interface to allow end users to cancel an Addin operation
- added several AutoCAD commands for controlling OpenTD servers
- updated all public methods that accept List to accept IEnumerable instead, for LINQ and general flexibility
- updated LogicObjectRoutineTypes values for clarity
- ThermalDesktop.Quit() now defaults to a gentle quit, but can be forced to kill the acad process
- added DomainManager.GetDomain method for getting the entities in any type of domain
- added DataArray constructor that just accepts a list of doubles
- added Enable/Disable Undo methods -- disabling undo can speed up operations
- added overloaded ThermalDesktop.SetView methods, which wrap the Restore*View methods
- added ThermalDesktop.ClearSelection method
- added ThermalDesktop.GetMainWindow method to get the AutoCAD window to use as the owner of OpenTD dialogs
- added Notes.HasPassword list to identify notes that are password-protected
- added Comparer.SaveExceedancePlots method
- Comparer now works with registers
- added Comparer.CompareJustFinalRecord member
- added Get/SetInstanceData methods to allow storage of general data with a TD instance

- **6.3 New Features Included in 6.2**
- added support for 3d polylines
- added ThermalDesktop.Get- and SetNodeCorrespondenceMap methods
- added GetSelection method to prompt the user to select entities manually in TD
- added SetSelection method to highlight objects in the TD graphics window
- added GetEntityTypes method to get entity types from handles
- test applied to objects for consistency before creating or updating heatload
- added GetXREFpathnames method
- added DomainManager.GetAnysetDomain method for getting the entities in an ANYSET domain
- added SS_CONSTANT_TEMP_APPLIED_NODES as an allowable value for Heater.SSMethod

https://crtech.com/videos/whats-new-opentd-v63

C&R TECHNOLOGIES
www.crtech.com

Getting Started with OpenTD

These videos give an overview of OpenTD:
https://www.crtech.com/videos/introducing-opentd-api-thermal-desktop
https://www.crtech.com/videos/whats-new-opentd-62
https://crtech.com/videos/whats-new-opentd-v63

And here's a detailed list of features: https://www.crtech.com/forum/viewtopic.php?f=53&t=2084

If you're new to OpenTD, you should start by reading "Getting Started with OpenTD 63.pdf", which can be found in your Thermal Desktop 6.3 installation directory under "Manual". Once you've mastered the concepts presented in the Getting Started guide, visit the user forum at www.crtech.com for additional OpenTD demos and examples, including how to use OpenTD with MATLAB or Python.

If you've read the Getting Started guide and are having trouble getting OpenTD to work, please let us know. We'll be happy to help.

OpenTD requires at least TD 6.1 and AutoCAD 2017. A preliminary version of the API called "TD API V1" was released with TD 6.0.

C&R TECHNOLOGIES
www.crtech.com

Acknowledgements

- OpenTD Project Hub Slide
  - ✓ James Etchells / ESA: STEP-TAS/TD Converters
  - ✓ Hume Peabody / NASA GSFC: Various TD Utilities
  - ✓ Victor Freeman / RAFAEL: Tplot
  - ✓ Derek Hengeveld / Redwire: Veritrek
  - ✓ Daniel Bae / NASA GSFC: TD Integration into Multiphysics Workflow
  - ✓ Stefano Cappucci / NASA JPL: Ingenuity Mars Helicopter Mission Operations

C&R TECHNOLOGIES
www.crtech.com