INTRODUCTION TO FEMAP API PROGRAMMING

Chris Teague
11-June-2015
Overview

• What is FEMAP API?
• Why develop for FEMAP?
• Developing for FEMAP using other programs
• API programming window
• API Examples
• Program Files and how they compare to API programs
• API Development with Visual Studio
• Object Oriented Programming (OOP)
• Dialog Editor
• Demonstration
What is FEMAP API?

• API stands for Application Program Interface
  • It is a programmatic interface to FEMAP tools, features, and data
• The API allows the user to customize FEMAP to extend the functionality to suit the needs of specific users or projects
• FEMAP includes built-in Visual Basic Integrated Development Environment (IDE) functionality, so you can start programing with the FEMAP API without any extra programs or optional modules
Why Develop for FEMAP?

- Create programs that automate repetitive tasks
- Enhance the functionality of FEMAP by creating add-in applications
  - Create programs that provide “canned” analysis routines
  - Ensure consistency in analysis methods between engineers
- Use FEMAP as a development platform
  - Use FEMAP functionality to develop your own programs
- Automate generation of analysis reports
- Move data generated in other programs like Excel or Matlab into FEMAP or vice versa
API Overview

- Interpreter built-in to FEMAP uses VBA language (Visual Basic for Applications)
  - Identical to writing scripts/macros in MS Office, just different commands
- Create quick, simple user dialogs or full-fledged compiled programs using Visual Studio
- Using type library references, other language options and IDE’s are available outside of FEMAP
  - C#, C++
  - Visual Basic
  - Visual Studio
Developing for FEMAP using other programs

- Component Object Model (COM) Interface allows FEMAP to interface with other programs
  - FEMAP can interact with other programs and other programs can interact with FEMAP (including standalone programs)
  - Ex: use FEMAP to populate Excel, or use Excel to drive FEMAP
- Compatible with .Net platform and Visual Studio
- Direct access to FEMAP tools and database / entity objects
- Leverage FEMAP as a development platform, not just an FEA package
Sending Data from FEMAP to Excel
# Referencing FEMAP from Excel

In this example, we demonstrate how to use Excel to reference FEMAP data. The spreadsheet is named `GroupsForFemap.xslm` and contains the following setup:

<table>
<thead>
<tr>
<th>Column A</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Spar 1</td>
</tr>
</tbody>
</table>

- **Cell A1** contains the label **Spar 1**.
- **Cell B4** contains the value **189**.
- **Cell C4** contains the text **Create FEMAP Group**.

The diagram on the right side of the Excel sheet shows the Model Info panel with the following hierarchy:

- **Coordinate Systems**
- **Geometry**
- **Connections**
- **Aero Model**
- **Model**
- **Analyses**
- **Results**
- **Views**
- **Groups**
- **Layers**
- **1.Spar 1**

This setup allows for easy referencing and manipulation of FEMAP data within Excel.
The FEMAP API Programming window provides a simple and effective way to create programs for FEMAP.

- Built-in IDE within FEMAP
  - Language is based on VBA
  - Syntax highlighting
  - Code completion
  - Object browser
  - GUI editor for simple dialogs
  - Debugging tools
  - Files are pure text (*.BAS)
  - API files are easily modified and transported between different users

Accessed via: Tools -> Programming -> API Programming
FEMAP API Programming Pane Toolbar – 2

- **Step Into**
- **Step Over**
- **Step Out**
- **Run**
- **Step to Cursor**
- **Pause**
- **Stop**
- **Set Breakpoint**
- **Clear Breakpoints**

```
Sub Main
    Dim App As femap.model
    Set App = GetObject(, "femap.model")

    Dim rc As Long

    ' Create Constraint Set 1
    Dim cs As Object
    Set cs = App.feBCSet
```
API Programming Pane: File Tabs

- When multiple files are open, use the tabs to change between files
- Drag the tab area wider if you want to see the filenames
- Single clicking a tab will switch to that tab
- Clicking X on a tab will close that tab
API Example – Complex Program

- RTE Reader
  - Reads and parses output from thermal evaluation program
  - Creates 2D or 3D models with appropriate loads, temperatures and constraints
  - Allows engineer to vary mesh fidelity
- Custom GUI
- Written in VB.Net using Visual Studio 2008
- 1 month development time
- Significant reduction in model creation time
  - 2D models created in seconds rather than 2-3 days
  - 3D models created in minutes rather than 5-6 days
- Reduced the burden on engineers to create models; more time could be spent analyzing results and performing trade studies
API Example – Complex Program
API Example – Complex Program

2D model with thermal contours

Partial 3D model
This API was created as a demo to show how to automate report generation from FEMAP to Word.
API Example – Report Generation

- This is the Word document exported from FEMAP

FEMAP Report

11/29/2012

Contour Plot
The Saratech Advantage menu was developed using an API program so we could easily automate the installation on other computers.
• Repetitive commands can be recorded and played back using FEMAP Program Files (.pro or .prg files).

• The Program File pane controls the recording and playback of Program files.

• Program Files are NOT guaranteed to be work in prior or later releases of the FEMAP version it was recorded in.

• Program Files can be modified to accept user input and can be executed from API scripts.
**PRO/PRG vs. API**

- **PRO/PRG files are a mouse / keystroke emulator**
- **Works similar to the MS Office macro recorder to record, but does not record API calls, it records keystrokes.**

<table>
<thead>
<tr>
<th>Benefits</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Simple to create</td>
<td>• Based on VBA</td>
</tr>
<tr>
<td>• Can emulate all functionality</td>
<td>• Exposes FEMAP “back end”</td>
</tr>
<tr>
<td></td>
<td>• Fast</td>
</tr>
<tr>
<td></td>
<td>• Nearly unlimited potential</td>
</tr>
<tr>
<td></td>
<td>• Easy customization</td>
</tr>
<tr>
<td></td>
<td>• Works between major versions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drawbacks</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Not object oriented</td>
<td>• Steeper learning curve than PRG files</td>
</tr>
<tr>
<td>• Customization is very difficult</td>
<td>• Not all FEMAP functionality is exposed (~95%)</td>
</tr>
<tr>
<td>• Slow</td>
<td>• Simple tasks may be easier with PRG</td>
</tr>
<tr>
<td>• Very limited potential</td>
<td></td>
</tr>
<tr>
<td>• Unlikely to remain compatible between major versions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FEMAP Program File Pane Toolbar

- Save
- Open
- Clear
- Record
- Play
- Pause
- Toggle Breakpoint
- Clear Breakpoint
- Record Comments

Program File Pane:

$ Manage Load Sets
{~1224}
$ ---- Create or Activate Load Set
<A-T>Load Set 1<OK>

$ Manage Constraint Sets
{~1230}
$ ---- Create or Activate Constraint Set
<A-T>Constraint Set 1<OK>
• Found in <FEMAP Root>/pdf/api.pdf
• 2100 page reference manual documents all objects exposed through the type library
• Includes basic examples for some FEMAP API programs
• Class notes are structured in a similar manner to the API reference manual
• Does not cover the API programming window, custom dialogs. These are covered in the WinWrap manual:
  • Help -> Basic Language
• When writing programs for FEMAP, this is your Bible
• Can also be found in FEMAP online help (F1)
  • Help -> Programming
Custom Tools Button/Directory

- Contains programs that are found in the ~/<FEMAP Root>/API directory
- All are uncompiled .bas programs – excellent resource for ideas / tips
- Adding new programs to the Custom Tools button is as simple as copying the program to the directory
- Can contain compiled programs (*.exe), uncompiled programs (*.bas) or PRG files (*.prg/*.pro)
- Creating sub folders in the API directory will create additional sub-menus under the Custom Tools button
Visual Basic Basics

- Visual Basic is one of the easiest programming languages to learn
- What you already know about Visual Basic for Applications (VBA) programming for Office is applicable
  - Similar to Visual Basic .NET (VB.Net)
- Very forgiving
  - Not strongly typed; on-the-fly typecasting
  - Case insensitive
  - Memory management and garbage collection does not have to be the responsibility of the programmer
- Easy to debug
- Easy to learn syntactical requirements
Getting more out of the FEMAP API

- FEMAP API development is not limited to only the API programming window with FEMAP.
- FEMAP API can be referenced through the COM interface.
- Reference the FEMAP libraries and namespaces to provide access to FEMAP objects.
- External programs can be used to drive FEMAP, or FEMAP functions can be embedded within an external program.
Using Visual Studio and FEMAP API

- Visual Studio can be used to write FEMAP programs
- Multiple language choices
- Enhanced forms and controls are available
- Programs are compiled rather than interpreted so performance may increase

Info – The free “Express” versions of Visual Studio are perfectly capable. From a FEMAP standpoint they offer the same capabilities.
What is OOP?

• Object Oriented Programming
• OOP is a programming paradigm that uses “objects” and their interactions with other “objects” as the basis for program design
• Objects can be conceptualized as “building blocks”
• FEMAP API is an object-oriented system
• FEMAP objects are broken down into:
  • Application objects (Geometry, Meshing, LBC’s, Analysis Setup, Post Processing)
  • Tool Objects
    • Data Table
    • Read File
    • FEMAP Selector and FEMAP Sets
  • Results
  • Database Objects
Properties and Methods

- Objects can have properties and methods
- Properties and Methods accessed via the “.” operator
- Properties
  - Describe attributes of the object, such as
    - MyCar.Color
    - MyElement.Nodes()
  - Attributes can be singular or plural (ie. arrays)
- Methods
  - Perform an action on an object such as
    - MyTv.TurnOn()
  - Often require arguments
    - MyCar.SetColor(“green”)
User Dialog Editor

- Simple user dialogs can be created using the API Programming window.
- Use user dialogs to collect information from the user beyond what the simple InputBox can gather.
- Complex dialogs can be created outside of FEMAP using Visual Studio, etc.
- Access the Dialog Editor via the icon ( ) on the toolbar or CTRL-D from the API Programming window.
Embedding FEMAP

- FEMAP may also be embedded within your application
- Use FEMAP capabilities (graphics rendering, etc) within your own application
Live API Demo

- Hello World
- Dialog Box Sample
- Surfaces from CSV file
- Selective Node ID
- Using Excel to create Groups
- CBEAM Results Generator
- Word Report Generator
- Counting TRI3 elements
• Femap API is very powerful, but is still easy to use
• Allows automation and customization to increase your FEA productivity
• Upcoming Femap 301 API Classes (3 Day Class):
  • November 23-25, Mission Viejo, CA