MATLAB App and Toolbox Development

David Sampson
MathWorks Consulting
Agenda

- Frequently used design patterns
- Calling external code from MATLAB
- Packaging and distributing code
- Learning more
Frequently used design patterns
Frequently used design patterns

- Data access layer
- Componentized apps
- Graphics components
Data access layer

**Presentation tier**
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

**Logic tier**
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

**Data tier**
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Demonstration:

Data access layer
Data access layers in MATLAB

- Design pattern: adapter
- Define a data access API (a set of abstract methods)
- Implement a class for each actual data source or sink type
- Applications interact with data sources or sinks via these data access objects
Componentized apps
Demonstration:
Data visualization app
Event notification
Property access, method call
Componentized apps in MATLAB

- Design pattern: model-view-controller
- Decompose the app into a model and a series of views and controllers
- Model owns shared data and state
- Model is a handle class whose API includes properties, methods and events
- Views react to model events and update their graphics objects
- Controllers respond to user actions and set model properties and/or call model methods
Graphics components
Demonstrations:

Password entry box
Web map
Graphics components in MATLAB

- Design pattern: composite
- Handle class (matlab.mixin.SetGet)
- Create graphics in constructor, store in private properties
- Expose specific properties
- Expose standard properties: Parent, Units, Position, Visible
- Delete object when graphics is deleted, and vice versa
Calling external code
Calling external code from MATLAB

- Java
- .NET
- Python
- WSDL
- C/C++/Fortran MEX
- C shared libraries

Why?
- Use algorithms
- Use utilities and services
- Interface to systems
Demonstration:

Calling external code
Packaging and distributing code
MATLAB Toolbox files (.mltbx)

- Available from R2014b
- Packs multiple files into a single file to be distributed
- Includes metadata and product dependencies
- Single step to install (or update)
- Configures MATLAB path
- Registers apps and documentation

Toolbox artifacts

- Code
- Apps
- Examples
- Documentation
- Tests
- Utilities
Suggested toolbox structure

/my.prj                        /tbx/my
/util.m                        /Contents.m
/tbx                           /fcn.m
/releases                     /Class.m
/tests

where tbx is what is released, and contains...
Toolbox packaging dialog
Toolbox automation APIs

- `matlab.addons.toolbox.packageToolbox`
- `toolboxVersion`
- `installToolbox`
- `uninstallToolbox`
- `installedToolboxes`
Demonstration:

Toolbox UIs and automation APIs
Learn more

- Slides and some demos will be available
- “Building, Scaling, and Implementing Risk Model and Stress Test Framework” by Bet Herrera @ 1520
- Training
  - Building interactive applications in MATLAB
  - Object-oriented programming in MATLAB
- Workshops by MathWorks Consulting
  - App development
  - Testing
- Upcoming posts on Developer Zone, blogs.mathworks.com/developer
Thank you.

Questions?