MATLAB APPS & TOOLBOXES AT JAGUAR LAND ROVER
MATLAB EXPO – 5TH OCTOBER 2016
David Barry
Simulation Strategy & Tools | Simulation Group | Vehicle Engineering
Agenda

• Virtual Engineering Introduction
• MATLAB at Jaguar Land Rover
• How MATLAB tools are distributed throughout the organisation as apps and toolboxes using an in-house store
• Best practices for MATLAB tool development and how they are shared and encouraged
Agenda

• Virtual Engineering Introduction

• MATLAB at Jaguar Land Rover

• How MATLAB tools are distributed throughout the organisation as apps and toolboxes using an in-house store

• Best practices for MATLAB tool development and how they are shared and encouraged
11 vehicle lines.

3 UK vehicle assembly plants, with 2 UK design and engineering sites.

Nearly 40,000 people globally – headcount has almost doubled over the last five years.

Plants in China, India and Brazil.

Employs over 9,000 engineers and designers.

Sales network in 154 countries.

Jaguar Land Rover is the largest automotive employer in the UK.

150 awards won in 2015/16.
The Challenge for Engineering

- 5 Functions: Body, Chassis, Powertrain, Electrical, Vehicle
- 18 Customer attributes: e.g. Ride & Handling, Performance & Economy
- Massively increased Systems Complexity
- 10,000 Requirements
- 100,000 Test Cases
- 154 Global Markets
- 100 ECUs
- >100 Millions of lines of code
- 4500 Data Signals Refreshed 100*/Sec
- 1000s New Parts
- 55 Major Systems
- 1,200 features
- 100s New Leading Edge Technologies
- 1,200 features
- "More Great Product Faster"
Agenda

• Virtual Engineering Introduction
• MATLAB at Jaguar Land Rover
• How MATLAB tools are distributed throughout the organisation as apps and toolboxes using an in-house store
• Best practices for MATLAB tool development and how they are shared and encouraged
MATLAB at Jaguar Land Rover

• Best in class tool – professionally developed & supported
• Highly productive and integrated environment for engineers to do their job
• Toolboxes provide out of the box capability for specialist engineering tasks
• Framework to build our own tools
The Code Sharing Challenge

- Strategic partnership with MathWorks helped to develop an internal MATLAB user community
- Recognised opportunities for code sharing and consolidation
  - No consistent way to share code across the business
  - Non-value add overhead for developers to maintain custom code sharing mechanisms
  - Difficult and time consuming for new users to get started
  - Difficult to know what is available - duplication of effort across teams
- MathWorks Consulting helped us to develop a solution – JLR MATLAB App Store
Many departments manage all of their automation routines through centralised toolboxes.
JLR Apps & Toolboxes
- Examples

- Vehicle diagnostics analysis – 2 similar apps on the store – common code now part of central toolbox
Toolbox for centralised development of data structures & file handlers
JLR Apps & Toolboxes
- Examples

- Apps for managing and sharing models & engineering data
JLR Apps & Toolboxes - Examples

- Apps for testing Simulink models
JLR App Store
- Examples

- Event extraction & metric calculation
Agenda

- Virtual Engineering Introduction
- MATLAB at Jaguar Land Rover
- How MATLAB tools are distributed throughout the organisation as apps and toolboxes using an in-house store
- Best practices for MATLAB tool development and how they are shared and encouraged
MATLAB Apps & Toolboxes
MATLAB Apps & Toolbox Packaging

- MATLAB Apps & Toolbox technology provide the mechanisms for packaging our code
JLR App Store Features
- Web based store

MATLAB Store

Archive

Vehicle Feedback XML Analyzer Application
by Simulation Group
This application is intended to import Vehicle Diagnostic Data (DTCS/DDLDe) from various XML file formats from around the business.

Engineering Data Toolbox
by Simulation Group
Toolbox providing importers for JLR engineering data files and a set of common data preparation methods. Type editDoc to get started.

App Manager
by Networks
Tool for managing installed apps and toolboxes

The App Manager allows you to:
- Connect to a MATLAB Store, either local or in a SharePoint site
- Check for new versions of installed apps and toolboxes
- Check for app and toolbox recommendations
- Update and download apps and toolboxes to a specified version

The App Manager uses MATLAB release compatibility information when suggesting and displaying add-on versions, if this information is provided by the store.

Download
Download the app or toolbox to your local MATLAB.

Share
Share the app or toolbox with others.

Owner

Permission

Compatibility

Remove

Recommended as 'Latest Version' to:
1. JLRINTESA/shared users

App Manager

Compatibility

Recommended versions:
- R2015b
- R2015a
- R2015b
- R2015a
- R2014b
- R2014a

Version 3.4

This is the latest version of the App Manager which supports the new compatibility feature recently added to the store. Please upgrade to this version. If you are using MATLAB R2015b then you should download the App Management Toolbox rather than this app.
JLR App Store Features
- App Manager
JLR App Store
- Usage > Number of submissions

[Graph showing the number of submissions over time from May 2015 to October 2016]
JLR App Store
- Usage > Number of users

![App Store Users Chart]

- Total Users: 1200
- Date Range: May 2015 to Oct 2016

The chart shows the growth in the number of users over time, with a significant increase from 2015 to 2016.
JLR App Store
- Usage > Number of downloads

![App Store Downloads Graph]
### JLR App Store
- Usage > Number of downloads

<table>
<thead>
<tr>
<th>App Name</th>
<th>Total Downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering Data Toolkit</td>
<td>704</td>
</tr>
<tr>
<td>JLR Layout Toolkit</td>
<td>695</td>
</tr>
<tr>
<td>App Manager</td>
<td>684</td>
</tr>
<tr>
<td>License Borrowing</td>
<td>673</td>
</tr>
<tr>
<td>BCS Toolkit</td>
<td>662</td>
</tr>
<tr>
<td>App Management Toolkit</td>
<td>651</td>
</tr>
<tr>
<td>JUA Test Framework</td>
<td>640</td>
</tr>
<tr>
<td>Spinnies</td>
<td>629</td>
</tr>
<tr>
<td>Jera Wizarda Toolkit</td>
<td>618</td>
</tr>
<tr>
<td>ORC Linear Model</td>
<td>607</td>
</tr>
<tr>
<td>GUI Controls Toolkit</td>
<td>596</td>
</tr>
<tr>
<td>VGT Testkit</td>
<td>585</td>
</tr>
<tr>
<td>Vehicle Feedback Toolkit</td>
<td>574</td>
</tr>
<tr>
<td>VGT Compare</td>
<td>563</td>
</tr>
<tr>
<td>Model &amp; Software Development Toolkit</td>
<td>552</td>
</tr>
<tr>
<td>HydroMount Tuner</td>
<td>541</td>
</tr>
<tr>
<td>PDiTools ReadBack 1.2</td>
<td>530</td>
</tr>
<tr>
<td>Road Builder</td>
<td>519</td>
</tr>
<tr>
<td>PDiTools Q8 Tool</td>
<td>508</td>
</tr>
<tr>
<td>Charge Balance Data Analyser</td>
<td>497</td>
</tr>
<tr>
<td>ORC WTB Simplifier</td>
<td>486</td>
</tr>
<tr>
<td>Damper Module Toolkit</td>
<td>475</td>
</tr>
<tr>
<td>Vehicle Dynamics Toolkit</td>
<td>464</td>
</tr>
<tr>
<td>DCM Compare Tool</td>
<td>453</td>
</tr>
<tr>
<td>Matrix Toolkit</td>
<td>442</td>
</tr>
<tr>
<td>Model Repository Browser</td>
<td>431</td>
</tr>
<tr>
<td>VGT Channel Filter</td>
<td>420</td>
</tr>
<tr>
<td>Glazed</td>
<td>409</td>
</tr>
<tr>
<td>Log Viewer</td>
<td>398</td>
</tr>
<tr>
<td>Kinz Test Runner</td>
<td>387</td>
</tr>
<tr>
<td>Automated Datasheet Creator</td>
<td>376</td>
</tr>
<tr>
<td>GuidedDiagnostic</td>
<td>365</td>
</tr>
<tr>
<td>Video Character Recognition</td>
<td>354</td>
</tr>
<tr>
<td>PMU Cheddar</td>
<td>343</td>
</tr>
<tr>
<td>ORC XSensor Data Processor</td>
<td>332</td>
</tr>
<tr>
<td>DAV Toolkit</td>
<td>321</td>
</tr>
<tr>
<td>MW License Tracker</td>
<td>310</td>
</tr>
<tr>
<td>SSB Sensitivity Tool</td>
<td>309</td>
</tr>
<tr>
<td>OGS Log Viewer</td>
<td>298</td>
</tr>
<tr>
<td>Talklab Heatmaps</td>
<td>287</td>
</tr>
<tr>
<td>Powertrain Sound Quality ORC Tuning</td>
<td>276</td>
</tr>
<tr>
<td>TLS</td>
<td>265</td>
</tr>
<tr>
<td>Vehicle Mass Analysis Tool</td>
<td>254</td>
</tr>
<tr>
<td>Crown Speed</td>
<td>243</td>
</tr>
<tr>
<td>PDiTools TakeAway pnt</td>
<td>232</td>
</tr>
<tr>
<td>Powertrain Sound Quality ORC Tuning</td>
<td>221</td>
</tr>
<tr>
<td>SET MSD Database Management</td>
<td>210</td>
</tr>
<tr>
<td>AID-on Approval Tool</td>
<td>209</td>
</tr>
<tr>
<td>PDiTools FilmRepair pnt</td>
<td>198</td>
</tr>
<tr>
<td>AEV Comparator</td>
<td>187</td>
</tr>
<tr>
<td>CRS Connector</td>
<td>176</td>
</tr>
<tr>
<td>DRC not to .org Surface Profile Connector</td>
<td>165</td>
</tr>
<tr>
<td>S-Function Connector</td>
<td>154</td>
</tr>
<tr>
<td>vibration Analysis Tool</td>
<td>143</td>
</tr>
<tr>
<td>Nanovibe Performance Analysens</td>
<td>132</td>
</tr>
<tr>
<td>Toolmate CSV Postprocess</td>
<td>121</td>
</tr>
<tr>
<td>FuzzySearchFinder</td>
<td>110</td>
</tr>
<tr>
<td>Materials LCF File Analysis</td>
<td>99</td>
</tr>
<tr>
<td>FT Wav Exhaust Assessment Report</td>
<td>88</td>
</tr>
<tr>
<td>Temperature Monitor</td>
<td>77</td>
</tr>
<tr>
<td>DTU Analyser</td>
<td>66</td>
</tr>
</tbody>
</table>

*App Store - Total Number of Downloads*
*1st May 2015 - 1st October 2016*
JLR App Store
- Benefits

• Accelerate adoption of existing tools by new users
• Reduce duplication
• Improve quality & usability
• Reduce time spent on non-value-add tasks
• Encourage and recognise sharing of high-impact tools
• Educate developers
Approved Apps & Toolboxes – check for JLR coding standards and best practices

Integrate the store with automated testing environment

Jenkins
App Store Future Development Plans

- Continue to engage with MathWorks developers for enhancements to MATLAB
- Add-On Explorer integration with JLR store
- Manage custom toolbox dependencies
- Improvements to App Designer
Agenda

• Virtual Engineering Introduction
• MATLAB at Jaguar Land Rover
• How MATLAB tools are distributed throughout the organisation as apps and toolboxes using an in-house store
• Best practices for MATLAB tool development and how they are shared and encouraged
Writing MATLAB Tools
- JLR best practices

- Apps are hand-written using GUI Layout Toolbox for positioning & resizing
- Apps should follow the Model View Controller (MVC) design pattern
- Use a consistent look and feel for user interfaces
- Make use of GUI mockup and architecture/class diagram tools for large projects
Writing MATLAB Tools
- JLR best practices

- Use a source control and bug tracking tool
- Use a consistent naming convention
- Know your end-user – casual vs developer interfaces
- Use MATLAB Unit Testing Framework
Writing MATLAB Tools
- JLR best practices

- Develop a standard toolbox to group together data structures and file handlers
- Document all of your code as you write it
- Use MATLAB custom documentation to provide help and examples for your own toolboxes

```matlab
function c = addme(a, b)
% ADDME: Add two values together.
% C = ADDME(A) adds A to itself.
% C = ADDME(A, B) adds A and B together.
% See also SUM, PLUS.

switch nargin
    case 2
        c = a + b;
    case 1
        c = a + a;
    otherwise
        c = 0;
end
```
Writing MATLAB Tools
- Sharing best practices

- User guides & how-to videos on the App Store
- On-site MathWorks training including JLR custom courses for OOP & app building
- MathWorks consulting projects
- Internal go-to experts
- Monthly lunch & learn events
- Internal WebEx sessions
- Website & blog for internal MATLAB user community
THANK YOU

David Barry
Simulation Strategy & Tools | Simulation Group | Vehicle Engineering
dbarry1@jaguarlandrover.com