Strategies for MATLAB and Simulink Upgrades

Judy Wohletz
Principal Application Engineer
MathWorks Automotive Conference 2016
Upgrade Constraints

Windows

Compilers

32 bit

64 bit

MathWorks

Co-Simulation

Linux

Partner

NO ENGINEERS®
Upgrade Challenges

- Short time between projects to upgrade
- Encountered a show-stopping issue after an upgrade
- Limited knowledge of new features and their impact on the process
- Took multiple years to upgrade to a new version
- Custom tools don’t work in the new version
- Maintaining hundreds to thousands of models
- Will have to revalidate the models and code
- Users want to stay in the old version
Upgrade Workflow

Assess → Plan → Migrate → Test → Release → Support
ASSESS

- **Choose a Target Version**
  - Review [Release Notes](#) and [Bug Reports](#)
  - Contact MathWorks partners to verify which versions they support
  - Choose and change your target version if necessary

- **Initial Testing**
  - Run Upgrade Advisor on a model

- **Regression Testing**
  - Run regression tests on tasks that users commonly perform
  - Better to find the issues now than find them later
PLAN: Create a Business Case

- New features that will improve your workflows
- Upgrading to a new operating system
- Third party software will no longer support your version
PLAN: Define Your Goals Upfront

- Limit the scope of the upgrade process
  - Don’t try to do everything
- Upgrade existing models to a new version
  - Upgrade models without introducing new features
  - Validate models in the new version
- Upgrade custom tools to a new version
  - Replace custom tools with built-in Simulink functionality
MIGRATE: Initial Migration

- Select a large model to test the upgrade process
  - Complies with your modeling style guidelines
- Run Upgrade Advisor on the model
- Test your typical workflows
- Document warning and error messages
- Expand testing to more models
  - Different modeling styles
- Resolve the issues
MIGRATE: Automated Migration

- Automate the conversion process
- Create one MATLAB script that calls everything
- Recommend that the model expert convert his/her models
  - Aware of production deadlines
  - Will need to validate the model
TEST

- Custom Tools
  - Verify the custom tools perform the intended behavior in the new version
- Third Party Tools
  - Test third party tools with the new version
- Beta Testing
  - Start using the new MATLAB version for everyday work
RELEASE

- Training
  - New Features
  - Custom Tools

- Timing
  - Needs to be flexible
  - Don’t want to impact a production schedule
  - New models should use the new version
  - Set a deadline for the rest of the models
Continuous Upgrade Philosophy

- **Prerelease Testing**
  - Test your models and custom tools for each prerelease
  - Send your feedback to the MathWorks
  - Don’t wait until your next upgrade to start testing

- **Industry Model Testing**
  - Submit models to MathWorks Industry Model Testing
  - MathWorks will run tests on your models prior to the prerelease
  - Dramatically reduces the chance of release incompatibilities

- **Seminars**
  - To keep up to date with MathWorks new features
  - Will help you decide what version to upgrade to
Benefits

- Surprises are minimized.
- You don’t have to wait to use new features.
- The new version is released in a timely manner.
- MathWorks may resolve your issues before your next upgrade.
- Your future upgrades will be easier.
TOOLS

Upgrade Advisor

Performance Advisor
Simulink Test

Simulink Test™ provides tools for authoring, managing, and executing systematic, simulation-based tests of models, generated code, and simulated or physical hardware. It includes a Test Sequence block that lets you construct complex test sequences and assessments, and a test manager for managing and executing tests. Simulink Test enables functional, baseline, equivalence, path-tracing, back-to-back testing, including software in the loop (SIL), processor in the loop (PIL), and real-time hardware-in-the-loop (HIL). You can apply pass and fail criteria that include absolute and relative tolerances, limits, logical checks, and temporal conditions. Setup and cleanup scripts help you automate or customize test execution.

You can create non-intrusive test harnesses to test components in the system modal or in a separate test model. You can store test cases and their results, creating a repository for reviewing and investigating failures. You can generate reports, archive, and revise test results, rerun failed tests, and debug the component or system under test.

With Simulink Test and Simulink Verification and Validation™, you can link test cases to requirements captured in Microsoft Word, EVM® Rational® DOORS®, and other documents.

Support for industry standards is available through EEC Certification Kit (for EC 61508 and ISO 26262) and DO Qualification Kit (for DO-178).
RESOURCES

Upgrade Paper

MATLAB and Simulink Version Upgrades for Large Organizations
By Judy Webster and Fred buddy

Upgrade Center

Why Upgrade?
Get more out of MATLAB® and Simulink® by upgrading to the latest versions of your products. Most users can download the latest release and get started using the new features right away.

Users who rely on critical applications, and models built in MATLAB and Simulink see the benefits of upgrading regularly

- Maximize the value of your MATLAB and Simulink software
- Take advantage of the features you requested
- Increase workflow efficiency and build confidence in your solutions
- Save time by upgrading regularly

- How do I upgrade?
- Contact support for help upgrading
- Upgrade now

Maximize the value of your MATLAB and Simulink software

Get the most out of MathWorks Software Maintenance Service. Every year, MathWorks delivers a steady stream of new technology, customer-requested product enhancements, and quality improvements. Join other companies in taking advantage of the features that you are helping to improve.
SUPPORT

Technical Support

Consulting Services
MATLAB and Simulink Upgrade Experience

Chris DeBoer
Eaton Corporation – Vehicle Group
Upgrade Process Overview

- Assess the Current Situation
- Plan Business Case and Decision Gates
- Migration Activities
- Testing the Upgraded Toolchain
- Release Process
Assess the Current Situation

- Initial Investigation
  - Revealed Enterprise move to new OS
  - Discovered issues with Custom Storage Classes

- Consultation with MathWorks
  - Changed Target version from R2014b to R2015b
  - Received guidance to resolve Custom Storage Class issues
Create Business Case and Decision Gates

- **Business Considerations**
  - Cost: Time, Resources
  - Program Timing
  - Risks

- **Decision Gates**
  - Business Case
  - Migration Progress
  - Testing
  - Pre-Release
Migration Activities and Testing

• Upgrade Custom Storage Classes
  • Format Change from R2011b to R2012a

• Resolve New Errors and Warnings
  • Heavy use of “Update Diagram”
  • Limited use of Upgrade Advisor

• Update TLC files

• Update Build process
  • Legacy Build process to Standard Build process
Testing

- Simulation Testing
- HiL Testing
- Test Cell Testing
- Vehicle Testing
Release

- User Installation
  - Installation Information distributed prior to Release
  - Required confirmation of simulation and build
- User Training
  - User Interface
  - New build process
- User Testing
  - Beta Testers verify updated toolchain
- Full Release