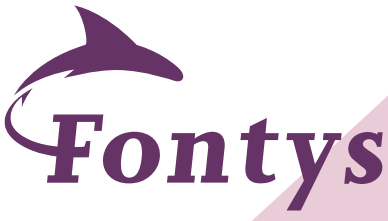
# Analysis and Considerations for Using MATLAB Standalone Archives

Tomasz Olejarczuk




22-04-2024

# Contents

# Analysis and Considerations for Using MATLAB Standalone Archives

## 1. Summary


### Challenges with File Exchange and Licensing

The initial implementation of the project relied on file exchange for data transfer between various components. This approach presented several limitations. Firstly, file exchange can be difficult to manage and is a bottleneck for the entire system, since the algorithm scans for documents on the same directory over and over. This leads to potential performance issues and is not scalable, especially as the volume of data increases. Additionally, the server-based MATLAB application creates a scalability constraint due to both the file-exchange mechanism's limitations and the licensing model, which would require licensing each instance separately. First approach was to address licensing problem and pack the application into a docker image that would not require a separate license.


## 2. Introduction

MATLAB Standalone Archives provide a mechanism to package MATLAB code and its dependencies into self-contained executable applications. These executables can run on compatible systems without requiring a full MATLAB installation. This functionality is beneficial for several use cases:

- **Distributing MATLAB Algorithms:** Researchers, engineers, or analysts can share their MATLAB-based calculations and tools with a wider audience who may not have MATLAB licenses.
- **Embedding in Other Applications:** Standalone Archives facilitate the integration of MATLAB computations into other software systems or potentially even hardware devices.
- **Simplified Deployment:** Users can run the executable as they would any other program, eliminating the need for complex MATLAB setup on their machines.

However, MATLAB Standalone Archives do have limitations. Their executables can be tied to specific operating systems, and they might become large in size. They are less suitable for applications requiring significant interactivity, dynamic data input, complex user interfaces, or seamless web integration. In such scenarios, alternative approaches should be considered, such as MATLAB Production Server, MATLAB Web Apps, or cloud-based MATLAB platforms. File system constraint and ease of deployment can be fixed by packaging the Matlab Standalone Archive into a docker image.

# 3. Methodology

## 3.1. Technical Overview

The process of creating a MATLAB Standalone Archive involves compiling MATLAB code and functions into a packaged application. This includes any necessary dependencies and the streamlined MATLAB Runtime, which enables execution without a full MATLAB installation. The resulting executable format depends on the target operating system (Windows, Linux, macOS). I created a Matlab build script to automate the process:

```
function build()

res = compiler.build.standaloneApplication('teamiumalgorithm.m', '
    SupportPackages', 'autodetect');
opts = compiler.package.DockerOptions(res,'ImageName','matlab-algorithm
    -app');
compiler.package.docker(res,'Options',opts);

end
```

### 3.1.1. Build script code breakdown

1. **Building the standalone application**

   `compiler.build.standaloneApplication` - this command builds the standalone application

- **Parameters:**
  - `'SupportPackages', 'autodetect'` - Matlab takes those parameter as a name of the parameter and it's value, so it autodetects which `SupportPackages` are needed automatically
  - `'teamiumalgorithm.m'` - entrypoint file

```
res = compiler.build.standaloneApplication('teamiumalgorithm.m', '
    SupportPackages', 'autodetect');
```

2. **Setting Docker build options** `compiler.package.DockerOptions` - this command sets parameters for building docker image

- **Parameters:**
  - `res` - standalone application being passed as a parameter to docker build
  - `'ImageName','matlab-algorithm-app'` - name of the docker image

```
opts = compiler.package.DockerOptions(res,'ImageName','matlab-algorithm
    -app');
```

3. **Building the docker image** `compiler.package.docker` - builds the docker image

- **Parameters:**
  - `res` - standalone application being passed as a parameter to docker build
  - `'Options',opts` - docker build options from the previous line are passed here

```
compiler.package.docker(res,'Options',opts);
```

### 3.2. Licensing Advantage

In a Dockerized setup, MATLAB Standalone Archives eliminate the need for individual licenses on the server. By packaging the code as an executable within a Docker image, you create a self-contained environment for running your MATLAB computations. Additionally, the archiving process helps to partially obscure the underlying source code, offering a degree of intellectual property protection.

## 4. Results

### 4.1. Advantages

- **Simplified Deployment:** MATLAB Standalone Archives significantly streamline the deployment process. By packaging all necessary dependencies, including the MATLAB Runtime, the executable functions independently of a full MATLAB installation on the target machine. This makes distribution and setup remarkably easier for the end-user.
- **Intellectual Property Protection (Partial):** While not foolproof, the compilation process used to create a Standalone Archive obscures the underlying MATLAB source code. This offers a degree of protection against direct access or easy reverse engineering of your intellectual property.

### 4.2. Disadvantages

- **Persistent Reliance on File Exchange:** A critical disadvantage of Standalone Archives is that they do not fundamentally solve the limitations of a file-exchange based architecture. Performance bottlenecks and scalability issues caused by the file exchange mechanism remain.

- **Platform Restrictions (Mitigated):** While Standalone Archives themselves can be platform-specific, containerizing them within a Docker image significantly mitigates this limitation. Your Docker image should be able to run on any system supporting Docker, regardless of the underlying host operating system.
- **Limited Flexibility:** Standalone Archives are inherently less flexible than a fully interactive MATLAB environment. They excel at executing well-defined sets of calculations but are less suitable for applications requiring dynamic input/output, real-time data updates, or the creation of complex web-like interfaces.
- **Overhead:** The inclusion of the MATLAB Runtime and essential dependencies within a Standalone Archive can lead to a substantial file size. This might be a consideration for deployments where storage space or transfer speeds are a concern.

## 5. Insights and Conclusion

MATLAB Standalone Archives offer a way to partially address the licensing challenges of server-based MATLAB deployments. When packaged within Docker images, they improve portability and simplify the execution process. However, their inherent limitations underscore the need for alternative solutions when the architecture demands flexibility, dynamic interactions, or scalability beyond the constraints of a file-exchange mechanism. In such scenarios, technologies like MATLAB Production Server, which enable API-like communication, or cloud-based MATLAB platforms may offer more suitable approaches.

## 6. Recommendations

- **Explore Alternatives if File Exchange is a Bottleneck:** If the file-exchange mechanism is a significant performance or scalability constraint, prioritize investigating solutions like MATLAB Production Server. This will enable you to move away from file-based communication and potentially unlock greater efficiency.
- **Focus on MATLAB Code Efficiency:** Within your Standalone Archive, ensure your MATLAB code is optimized for performance. This includes efficient algorithms, minimizing resource-intensive operations, and thorough profiling to identify any bottlenecks within the MATLAB computations.
- **Future-Proofing:** Consider the potential growth of your project. If you anticipate needing more dynamic interactions, web integrations, or greater scalability in the future, investing time in exploring more flexible technologies early on might save effort in the long run.