CONFIDENTIAL INFORMATION

ATLAS 10 AUTOMATION API GUIDE

ATLAS PLATFORM

REVISION HISTORY

Version	Date	Author	Reviewer	Changes
1.0	05/10/2018	Matthew Bristow	Steven Morgan	Initial Draft.

CONTENTS

1 Important Notes

Feedback/Support

If there are issues, please contact your Track Support Engineer for further assistance.
You can also submit bugs and suggestions for future releases through the ATLAS 10 Zendesk Portal or email ATLAS 10 Support.

2 Application Version

ATLAS 10.2.18311.1

3 Application Licensing

ATLAS 10 Evaluation

4 OCS Software Dependencies

SQLRace currently supported Version 2.1.18309.3
Recommended SQLRace Database Version 1.50

5 Prerequisites

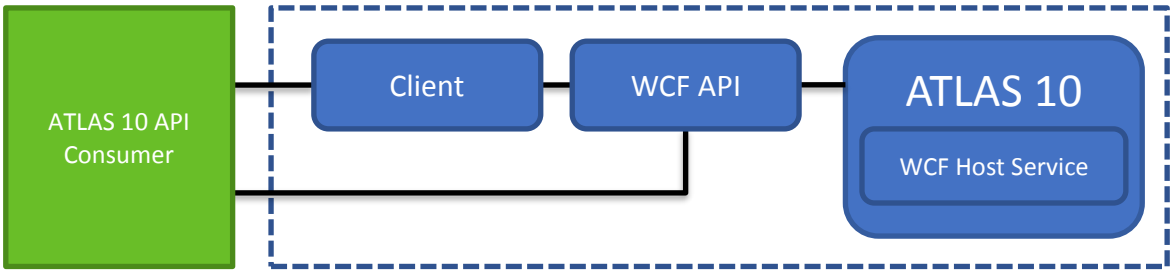
Microsoft .NET Framework 3.5
Microsoft .NET Framework 4.6.2

6 Assumptions

Experience of the ATLAS 10 platform
Understanding of programming fundamentals

7 Introduction

ATLAS 10 implements a runtime service-orientated automation API (Application Programming Interface) using WCF (Windows Communication Foundation). The WCF API can be consumed directly like a traditional WCF service but to make consuming the API an easier experience to develop against we have wrapped it into a Client (MAT.Atlas.Automation.Client.dll) DLL (dynamic link library).



It has been designed to allow ATLAS 10 to be controlled by an application that can consume Microsoft .NET Framework 4.6.2 such as:

- C#
- VBA (Visual Basic for Applications)
- MATLAB

Any such program can:

- Extract data and other information from ATLAS 10
- Write transient data into ATLAS 10
- Automate many user actions such as loading sessions, adding displays, and parameters

N.B. if you want to create Sessions or get data in-process then you need to use the SQLRace API.

For more details on the ATLAS 10 Automation API please see the ATLAS 10 API Documentation which is installed with ATLAS 10. This documentation covers the common classes and interfaces for interacting and extending ATLAS 10.

Start Menu > McLaren Applied Technologies > ATLAS 10 API Documentation

8 Known Issues and Limitations

- Adding a transient parameter needs a parameter group.
- Only the first sessions are loaded into the Session Browser when using ApplicationService.LoadFileSessions and ApplicationService.LoadSqlRaceSessions but they are loaded within ATLAS 10.
- Users must manually un/register the automation API DDLs to be used in COM, and thus VBA.
- Currently the API only works when running your application on the same machine that is running ATLAS 10.
- To call parameters without their app name included ie “vcar” instead “vcar:chassis” you need to turn on “Resolve without app name” in ATLAS 10.

ATLAS 10 > Tools > Options > General > Data > Resolve without App Name

9 Examples

Three examples have been included demonstrating how to consume the automation client.

9..1 C# Example

This example demonstrates how to add a transient parameter. It does this by calling the Client directly from within a C# Console Application.

9..1.1 Prerequisites

Ensure the following components are installed before you get started:

a. ATLAS 10
b. Visual Studio 2015/2017*

* Any version that supports Microsoft .NET Framework 4.6.2

9..1.2 Structure of Source Code

HelloWorld.CSharp

Name	Description
Program.cs	This file contains the **entry point** of the application **Main**. This function contains all the code for this example.

9..1.3 APIs and Libraries

This example references the following libraries:

- MAT.Atlas.Automation.Client.dll
- MAT.Atlas.Automation.Api.dll

Both can be found within the ATLAS 10 installation directory.

9..1.4 Debug and Run

a. Launch ATLAS 10
b. Load a Session into Set 1 via the Session Browser
c. Launch Visual Studio
d. Open the Helloworld.CSharp project
e. Add breakpoints in the Program.cs and click *Start* or press F5 to start debugging.

9..2 VBA Example

This example demonstrates how to:

- Export data of a parameter to Excel
- Import data from Excel into ATLAS 10 as a transient parameter
- Display the original parameter and new transient parameter in ATLAs 10

This is achieved by calling the Client within an Excel module.

9..2.1 Prerequisites

Ensure the following components are installed before you get started:

a. ATLAS 10
b. Registration of automation DLLs
c. Excel

9..2.2 Structure of Source Code

HelloWorld.Vba\Excel

Name	Description
HelloWorld.Vba.xlsb	This is the demonstration file utilising the modules listed below.

HelloWorld.Vba\Modules

Name	Description
HelloWorld.bas	This file contains the **entry point** of the application **HelloWorldAtlas10**. This function contains all the code for this example.

9..2.3 APIs and Libraries

This example references the following libraries:

- MAT_Atlas_Automation_Client
- MAT_Atlas_Automation_Api

*N.B. before these can be used they first must be registered. It is planned for this to be part of the installer once officially released. See **9..2.4 Registering Automation API DLLs**.*

9..2.4 Registering Automation API DLLs

The following DLLs need to be registered to allow the WCF API to be usable from COM (and thus VBA):

- MAT.Atlas.Automation.Client
- MAT.Atlas.Automation.Api

If you are upgrading ATLAS 10 and plan to use the latest automation DLLs you must unregister previous registrations first.

N.B. registration is not required to use the WCF API from C# or MATLAB.

9..2.4.1 Register

a. Run cmd.exe as administrator
b. Change directory to the location of regasm.exe (Registration Assembly Tool)

```
cd C:\Windows\Microsoft.NET\Framework\v4.0.30319
```

N.B. this can be found in the .Net Framework 4 installation folder. This version number may vary depending upon the exact .Net version installed.

c. Register MAT.Atlas.Automation.Api.dll with the command below.

```
regasm "c:\Program Files\McLaren Applied Technologies\ATLAS 10\MAT.Atlas.Automation.Api.dll" /register /tlb /codebase
```

d. Register MAT.Atlas.Automation.Client.dll with the command below.

```
regasm "c:\Program Files\McLaren Applied Technologies\ATLAS 10\MAT.Atlas.Automation.Client.dll" /register /tlb /codebase
```

9..2.4.2 Unregister

a. Run cmd.exe as administrator
b. Change directory to the location of regasm.exe (Registration Assembly Tool)

```
cd C:\Windows\Microsoft.NET\Framework\v4.0.30319
```

N.B. this can be found in the .Net Framework 4 installation folder. This version number may vary depending upon the exact .Net version installed.

c. Register MAT.Atlas.Automation.Api.dll with the command below.

```
regasm "c:\Program Files\McLaren Applied Technologies\ATLAS 10\MAT.Atlas.Automation.Api.dll" /unregister /tlb
```

d. Register MAT.Atlas.Automation.Client.dll with the command below.

```
regasm "c:\Program Files\McLaren Applied Technologies\ATLAS 10\MAT.Atlas.Automation.Client.dll" /unregister /tlb
```

9..2.5 Debug and Run

a. Launch ATLAS 10
b. Load a Session into Set 1 via the Session Browser
c. Open HelloWorld.Vba.xlsm
d. Go to the Developer tab
e. Select Visual Basic
f. Double click the HelloWord under Modules to open
g. Add breakpoints in and click the Atlas 10 Hello World button or press F5 to start debugging.

9..3 MATLAB Example

This example demonstrates how to add a transient parameter. It does this by calling the Client directly from MATLAB.

9..3.1 Prerequisites

Ensure the following components are installed before you get started:

a. ATLAS 10
b. MATLAB R2017a*

*earlier versions will most likely work too.

9..3.2 Structure of Source Code

Name	Description
HelloWorld.m	This file contains a single function **WCFAPIExample** which contains all the code for this example.

9..3.3 APIs and Libraries

This example references the following libraries:

- MAT.Atlas.Automation.Client.dll
- MAT.Atlas.Automation.Api.dll

Both can be found within the ATLAS 10 installation directory.

9..3.4 Debug and Run

a. Launch ATLAS 10
b. Launch MATLAB
c. Change current folder to that which contains HelloWorld.m
d. Add breakpoints as appropriate
e. Click Run