

# ProVAL Clarity API - Getting Started

## Introduction

The ProVAL Clarity API has been developed to work with pavement surface image (PSI) files defined by the AASHTO MP 47 specification. These files contain 2D and 3D pavement image data. This API also supports some image files.

## Requirements

[Microsoft .NET 6.0 Desktop Runtime] (<https://dotnet.microsoft.com/en-us/download/dotnet/6.0>) is required. The API cannot be used with later versions of .NET. The API is not compiled for a specific platform (x86, x64, etc.).

## Architecture

This API uses common design patterns, particularly Inversion of Control (IoC) and the Abstract Factory. IoC allows developers to substitute Clarity implementations of interfaces with their own implementations.

## Getting Started

The easiest way to read files is by using `PavementSurfaceBuilder` to create a `PavementSurface`. A `PavementSurface` contains objects that store all data from a file, such as `ImageData`. The data can then be passed accessed directly or sent to a writer, such as an `IProfileWriter`.

Due to the extensive use of IoC, it is recommended to use a Dependency Injection framework, such as `Microsoft.Extensions.DependencyInjection`, to simplify object creation.

## Models

Only a few model objects are needed to contain the data for a pavement surface:

- `PavementImageInfo`: pre-defined information, mostly defined by MP 49.
- `Metadata`: user-defined information
- `ImageData`: intensity or range data

Additionally, `IClarityImage` provides a implementation-independent model for storing an image.

## Reading Files

The following file types are supported:

- PSI
- CRG
- JPEG
- PNG

A `PavementSurfaceBuilder` is a high-level object that can read files. For more control or to substitute your own implementations, low-level interfaces and classes are also provided. These objects are what `PavementSurfaceBuilder` uses internally.

- `IFileReaderFactory` and `IFileReader`
- `ImageDataFactory`, `IIntensityReader`, `IRangeReader`

# Writing Profiles

Longitudinal and transverse profiles can only be exported from range data. `TextProfileWriter` creates delimited text files. A Pavement Profile File (PPF) that corresponds to the AASHTO E2560 specification can be created by the PPF writers. Separate writers are provided for the 1.05 (`PpfWriter1dot06`) and 1.06 (`PpfWriter1dot06`) versions of E2560. To simplify creation of these writer classes, the `ProfileWriterFactory` can be used.

## Images

.NET 6 has limited support for images. External libraries are needed for the precision and operations required by Clarity. To avoid dependencies on a specific library, the API DLL only provides the interfaces needed for working with images. An implementation of these interfaces is provided by `ProVAL.Clarity.Api.Magick.dll`. This DLL uses the Q16 version of the `Magick.NET` library. Q16 is required to work with 16-bit range data without losing precision. The DLL has only been tested with the x64 package, but it can be used with any Q16 package.

To avoid a dependency on a specific image library, `IClarityImage` provides a wrapper around an image. The provided library, `ProVAL.Clarity.Api.Magick.dll`, includes an implementation, `ClarityImage`. The easiest method of creating an `IClarityImage` is by using an `IClarityImageFactory`.

## Validation

Two classes are provided to validate PSI files. `PsiStructureValidator` verifies that the signature and trailer of a PSI file adhere to MP 47. It also checks that the section lengths defined in the file are correct. `PsiDataValidator` validates that the data in a `PavementImageInfo` object adheres to MP 47. While this object can be populated with data from a non-PSI file, validation is most useful in conjunction with PSI files.

## Analysis

The `Analysis` namespace contains classes used to analyze data. Currently, the `RuttingAnalyzer` is the only analyzer provided.