

BPS 2022.1

Changes to SDK & API



1. Introduction	3
2. Changes to SDK.....	4
2.1. Change to targeting plugin projects	4
2.2. Deprecation of plugins for the Classic interface display mode	4
2.3. Addition of instance classes.....	4
3. Changes to BPS API	6
3.1. New version of BPS Rest API.....	6
3.2. Removal of BPS SOAP API.....	6

1. Introduction

The following document contains a list of changes in WEBCON BPS SDK and API when compared to version 2021.1. These changes break code compatibility or change how existing code works.

2. Changes to SDK

2.1. Change to targeting plugin projects

In version 2022 R1, plugin project targeting was changed to .NET Standard 2.0.

It is necessary to change the targeting of existing plugin projects by creating a new project and moving existing code to it, or by correctly modifying the existing csproj file.

The new project can be created based on a project template for BPS plugins, which is available in the Visual Studio extensions: WEBCON BPS 2022 Extensions

Due to the change to targeting plugin projects, it will also be necessary to change references to projects and DLL files to a version that targets .NET Standard 2.0

While plugins that target .NET Framework will still work in version 2022 R1, only plugins that target .NET Standard 2.0 will work correctly in 2022 R2. Therefore, it is advisable to migrate to .NET Standard 2.0 as early as 2022 R1.

2.2. Deprecation of plugins for the Classic interface display mode

Due to the Classic form interface being retired, the following plugins and methods related to the removed feature are also being removed:

- WebCon.WorkFlow.SDK.SP – project with plugin base classes for the Classic interface is no longer available.
- CustomFormField – logic and interface part has been removed. It is recommended to migrate to FormFieldExtension or replace it with the form field: *Custom control (External)*.
- ItemListExtension – logic and interface part has been removed. Version 2022 does not currently support plugins that modify the interface part of an *Item list*. The logic part may be replaced with CustomAction type plugins.
- AppPoolContext – AppPoolContext class has been removed. It used to allow code to be executed in the context of the application pool for SharePoint installations. Currently, all code for operations from the user interface is executed in the context of the application pool.

2.3. Addition of instance classes

Several tool classes have become instance classes, these include:

- DocumentAttachmentsManager
- DocumentsManager
- EmailsHelper
- UserDataProvider
- SecurityManager

- CompanyStructureHelper
- FileGenerationHelper

The creation of instances of these classes requires the plugin execution context. It can be accessed in most methods in the method's parameters via the Context property, e.g.:

```
public override void Run(RunCustomActionParams args)
{
    var context = args.Context;
}
```

Additionally, for CustomDataSource and FormFieldExtension classes, it is available in the base class, e.g.:

```
public class FormFieldExtension2: FormFieldExtension<PluginConfiguration>
{
    public override object OnDBValueSet(object valueToSet)
    {
        var context = base.Context;
        ...
    }
}
```

3. Changes to BPS API

3.1. New version of BPS Rest API

- A new version of BPS Rest API, known as version 4.0, has been added.
- Rest API version 2.0 has been designated as outdated, and it is recommended to no longer use it. It will be removed in the next version of WEBCON BPS.
- Rest API version 1.0 has been removed.

3.2. Removal of BPS SOAP API

BPS SOAP API has been removed – it is recommended to migrate to BPS Rest API version 4.0.