



MyScript Application Toolkit 2.2 Release Note

Table of Contents

1	Release Objectives	3
2	Component version	3
3	Installation	3
4	Diagram Widget	4
5	New APIs	4
5.1	Android	4
5.1.1	Math Widget	4
5.1.2	Geometry Widget	5
5.1.3	Diagram Widget	5
5.2	iOS	5
5.2.1	Math Widget	5
5.2.2	Geometry Widget	5
5.2.3	Diagram Widget	5
6	API Changes	6
6.1	Android	6
6.1.1	Math Widget	6
6.1.2	Geometry	6
6.2	iOS	6
6.2.1	Math Widget	6
6.2.2	Geometry Widget	6
6.3	Windows (UWP)	7
6.3.1	Math Widget	7
6.3.2	Geometry widget	7
7	Update Instructions	7

1 RELEASE OBJECTIVES

This is a new release of the MyScript Application Toolkit.

The objectives of this release are as follows:

- ATK 2.2 includes the following components for iOS, Android and Windows:
 - Single Line Text Widget (SLTW)
 - Single Char Widget (SCW)
 - Math Widget (MAW)
 - Geometry Widget (GEW)
 - Diagram Widget (DIW) (new: **now available for iOS**)
- Bug fixes and optimization
- Based on the latest **MyScript SDK**

2 COMPONENT VERSION

Requirements	iOS	<i>Targeted version:</i> from 8.0 to 9.3 ¹ <i>Development environment:</i> Xcode 7.3 and iOS SDK
	Android	<i>Targeted version:</i> ARM/x86 32 bit: from 4.0 to 6.0, ARM/x86 64 bit: 5.0 to 6.0 ² <i>Development environment:</i> Android Studio and Android tools and SDK (level 15 or higher)
	Windows	<i>Targeted version:</i> Windows 10 Universal application <i>Development environment:</i> Visual Studio 2015 and Windows 10.
Supported localization	None (English only)	
Version	2.2	
How to check version after installation	Each package comes with its <i>atk-info.json</i> text file that gives the version of the package.	

3 INSTALLATION

- Download your package,
- Unzip your package.

¹ ATK 2.2 was tested on iOS 10.0 beta. Although not fully qualified, it should function properly with the final iOS 10.0 release.

² ATK 2.2 was tested on Android 7.0 Nougat on a limited set of compatible devices, and should work properly with this version.

- Copy the certificate into the right directory:
 - For iOS the downloaded certificate must be deployed under:

`Samples\MyCertificate.c`

- For Android the downloaded certificate must be deployed under:

`samples\myscript-certificate\src\main\java\com\myscript\certificate\MyCertificate.java`

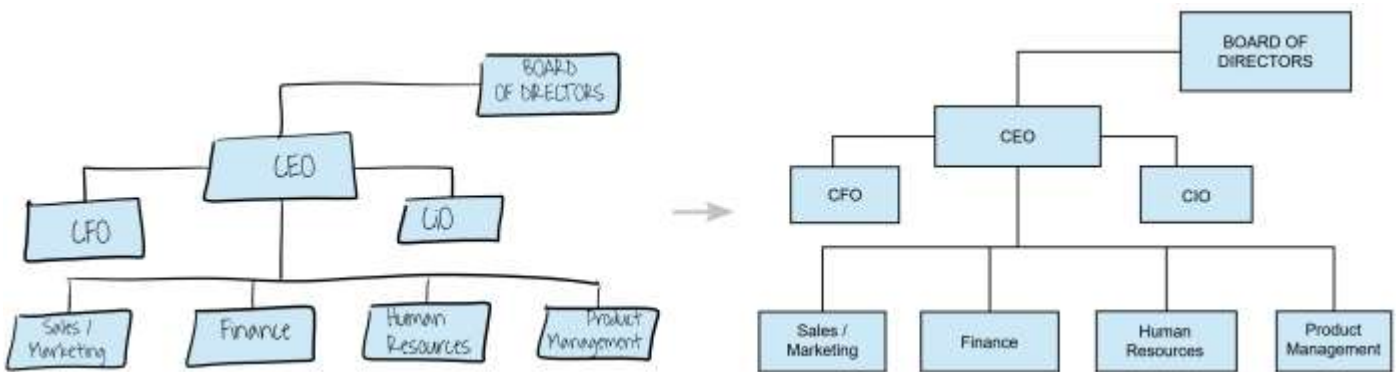
- For Windows the downloaded certificate must be deployed under:

`Samples\MyCertificate\MyScript.Certificate\MyCertificate.cs`

4 DIAGRAM WIDGET

With ATK 2.2, Diagram Widget is **now available for iOS**.

Diagram Widget is an interactive input method for text and shapes, enabling users to draw some kinds of diagrams freehand, with a finger or a stylus. It is designed to ensure a natural, handwriting-based HMI with real-time recognition and interactivity on digital ink.



It enables users to think visually when they take notes, brainstorm or create a document on a touch or stylus-enabled device.

Diagram Widget supports the following diagrams: org charts, flowcharts, concept maps, mind maps.

It offers: real-time text and shape recognition, beautification, interactive ink, interaction with the recognition, connection, export and serialization.

With support for over 10 different geometrical shapes, 58 languages and connectors.

5 NEW APIS

ATK 2.2 introduces new APIs for Android and iOS to support programmatic stroke and pen input in Math, Geometry and Diagram Widgets. This improves the consistency with the Windows platforms, where these APIs were already available.

5.1 Android

5.1.1 Math Widget

The following methods were added to `MathWidgetApi`:

- `void addStroke(List<CaptureInfo> points)` – Add a stroke to the model

- `void penDown(CaptureInfo info)` – Simulate a pen down, as if a user was writing
- `void penMove(CaptureInfo info)` – Simulate a pen move, as if a user was writing
- `void penUp(CaptureInfo info)` – Simulate a pen up, as if a user was writing
- `void penAbort()` – Abort a pen capture sequence

5.1.2 Geometry Widget

The following methods were added to `GeometryWidgetApi`:

- `void addStroke(List<CaptureInfo> points)` – Add a stroke to the model
- `void penDown(CaptureInfo info)` – Simulate a pen down, as if a user was writing
- `void penMove(CaptureInfo info)` – Simulate a pen move, as if a user was writing
- `void penUp(CaptureInfo info)` – Simulate a pen up, as if a user was writing
- `void penAbort()` – Abort a pen capture sequence

5.1.3 Diagram Widget

The following methods were added to `DiagramWidgetApi`:

- `void penDown(CaptureInfo info)` – Simulate a pen down, as if a user was writing
- `void penMove(CaptureInfo info)` – Simulate a pen move, as if a user was writing
- `void penUp(CaptureInfo info)` – Simulate a pen up, as if a user was writing
- `void penAbort()` – Abort a pen capture sequence

Addition of `DiagramWidgetApi.OnPenListener`, callback invoked when the user is writing on the widget

5.2 iOS

5.2.1 Math Widget

The following methods were added to the `MAWMathView` class:

- `-(void)addStroke:(NSArray<MAWCaptureInfo*>*)points` – Add a stroke to the model
- `-(void)penDown:(MAWCaptureInfo*)info` – Simulate a pen down, as if a user was writing
- `-(void)penMove:(MAWCaptureInfo*)info` – Simulate a pen move, as if a user was writing
- `-(void)penUp:(MAWCaptureInfo*)info` – Simulate a pen up, as if a user was writing
- `-(void)penAbort` – Abort a pen capture sequence

5.2.2 Geometry Widget

The following methods were added to the `GWMathView` class:

- `-(void)addStroke:(NSArray<GWCaptureInfo*>*)points` – Add a stroke to the model
- `-(void)penDown:(GWCaptureInfo*)info` – Simulate a pen down, as if a user was writing
- `-(void)penMove:(GWCaptureInfo*)info` – Simulate a pen move, as if a user was writing
- `-(void)penUp:(GWCaptureInfo*)info` – Simulate a pen up, as if a user was writing
- `-(void)penAbort` – Abort a pen capture sequence

5.2.3 Diagram Widget

New in this ATK version.

6 API CHANGES

ATK 2.2 introduces the following API changes. Please make sure to update existing code.

6.1 Android

6.1.1 Math Widget

`MathWidgetAPI.OnWritingListener` was replaced by `MathWidgetApi.OnPenListener`, with callbacks that match the newly added pen event methods.

6.1.2 Geometry

`GeometryWidgetAPI.OnWritingListener` was replaced by `GeometryWidgetApi.OnPenListener`, with callbacks that match the newly added pen event methods.

6.2 iOS

6.2.1 Math Widget

In `MAWMathViewDelegate`, the following methods were removed:

- `mathViewDidBeginWriting:`
- `mathViewDidEndWriting:`

Replaced by:

- `mathView:didPenDownWithCaptureInfo:`
- `mathView:didPenMoveWithCaptureInfo:`
- `mathView:didPenUpWithCaptureInfo:`
- `mathViewDidPenAbort:`

6.2.2 Geometry Widget

In `GWMathViewDelegate`, the following methods were removed:

- `geometryViewDidBeginWriting:`
- `geometryViewDidEndWriting:`

Replaced by:

- `geometryView:didPenDownWithCaptureInfo:`
- `geometryView:didPenMoveWithCaptureInfo:`
- `geometryView:didPenUpWithCaptureInfo:`
- `geometryViewDidPenAbort:`

6.3 Windows (UWP)

6.3.1 Math Widget

In the `MathWidget` classes, the `WritingBeginHandler` and `WritingEndHandler` events were removed, as their functionalities are already covered by the pen handler events.

In addition, the `AddStroke()` method was updated to get in parameter a list of `CaptureInfo` items instead of a table for a better consistency with other components and platforms.

6.3.2 Geometry widget

`OnWritingListener` was replaced by `OnPenListener`, with callbacks that match the newly added pen event methods.

As a result, in the class `GeometryWidget`, the following method was removed:

```
void SetOnWritingListener(OnWritingListener listener);
```

Replaced by:

```
void SetOnPenListener(OnPenListener listener);
```

In addition, the `AddStroke()` method was updated to get in parameter a list of `CaptureInfo` items instead of a table for a better consistency with other components and platforms.

7 UPDATE INSTRUCTIONS

Important: if you update your code from a previous version of ATK, please make sure not to mix together libraries from different ATK version. You will also need to update to the latest language resources.