

# Flex 1.1 - Release Notes

## On this page:

- [1.1.3-funcrel](#)
  - [Other Updates](#)
- [1.1.2-funcrel](#)
  - [Resolved issues](#)
  - [Updates](#)
    - [Improved analysis support](#)
    - [Reduced number of "UA Plugin: No Property found..." warnings](#)
    - [Flex Rules Updates](#)
- [1.1.1-funcrel](#)
  - [Resolved issues](#)
- [1.1.0-funcrel](#)
  - [Updates](#)
  - [Resolved issues](#)

## 1.1.3-funcrel

### Other Updates

Details
Now, the Adobe Flex analyzer does not depend on hard-coded "operator" role.
Incorrect wording "hardcoded" has been changed to "hard-coded"
Now, the Adobe Flex Analyzer does not support Oracle or SQL Server as Storage Service.

## 1.1.2-funcrel

### Resolved issues

R&D ID	Ticket ID	Summary
FLEX-165	15961	Flex warning- UA Plugin: No property (adobe.ac.pmd.rules.binding.ToolongBindingExpression) found in meta model
FLEX-139	15466	FLEX Warning : UA Plugin: No property found in metamodel
FLEX-140	15464	QR - Avoid type not found within analysis scope (Flex) should be updated

### Updates

#### Improved analysis support

Analysis support has been improved for certain Flex objects:

- [getter/setter/functions](#)
- [member variables](#)
- [classes](#)

Therefore after an upgrade to this version of the extension and the generation of a post-upgrade consistency snapshot, analysis results may be impacted: there may be an increase in the number of objects identified during the analysis and as a consequence, there may be impacts with regard to the violations detected for these Flex objects.

#### Related issues

- [FLEX-162 - Improved detection of getter/setter/functions](#)
- [FLEX-163 - Improved detection of member variables](#)

- FIEX-164 - Improved detection of Flex classes

## Reduced number of "UA Plugin: No Property found..." warnings

Changes have been made to reduce the warnings of type "UA Plugin: No Property found...". These changes will result in the following two impacts after an upgrade to this version of the extension and the generation of a post-upgrade consistency snapshot:

- Some of the Flex rules may have new violations
- The following rules will show all violations removed and then added back again because the violation object type has changed:
  - 1003198: Avoid MXML component with more than one entry point (Flex)
  - 1003200: Avoid MXML component with more than two entry points (Flex)

## Flex Rules Updates

The documentation of all Flex rules has been updated. As part of the update, rule titles have been updated as shown in the table below:

ID	Old Title	New Title
100 3010	Avoid class that imports an internal class from another function area (Flex)	Avoid class that imports an internal class from another functional area (Flex)
100 3016	Avoid long switch case (Flex)	Avoid long cases in switch (Flex)
100 3020	Avoid empty method (Flex)	Avoid empty overridden method (Flex)
100 3024	Avoid empty used private method (Flex)	Avoid empty private method (Flex)
100 3026	Avoid too complex methods. (Flex)	Avoid too complex methods (Flex)
100 3036	Avoid unused private attribute (Flex)	Avoid unused private field (Flex)
100 3038	Avoid Cairngorm event name not containing the function area name before the actual event name (Flex)	Cairngorm: Avoid Cairngorm event name without functional area (Flex)
100 3040	Avoid Event name not explicitly set (Flex)	Avoid Event name not set explicitly (Flex)
100 3044	Avoid too short field name (Flex)	Avoid field with name too short (Flex)
100 3046	Avoid too short variable name (Flex)	Avoid variable with name too short (Flex)
100 3056	Avoid class name not starting by a majuscule character (Flex)	Avoid class name that does not start with capital letter (Flex)
100 3058	Avoid field incorrectly named. (Flex)	Avoid fields with names that can reduce code readability (Flex)
100 3060	Avoid variable incorrectly named. (Flex)	Avoid variables with names that can reduce code readability (Flex)
100 3066	Avoid logger not correctly formatted (Flex)	Avoid incorrectly formatted logger (Flex)
100 3074	Avoid modelLocator that is Bindable at a class level (Flex)	Cairngorm: Avoid modelLocator that is Bindable at a class level (Flex)
100 3080	Avoid control statement in Constructor (must be as lightweight as possible) (Flex)	Avoid complex constructor (Flex)
100 3084	Avoid calls to the StyleManager that do not pass FALSE as the second parameter (Flex)	Avoid recursive StyleManager call (Flex)
100 3096	Avoid using public static fields (Flex)	Avoid non-const public static fields (Flex)
100 3100	Avoid protected accessors in a final class (Flex)	Avoid protected fields in final class (Flex)
100 3110	Avoid function that are too long (Flex)	Avoid functions that are too long (Flex)

100 3116	Avoid class (in script) with too many fields (Flex)	Avoid Flex Script with too many fields (Flex)
100 3120	Avoid ModelLocator not only accessible from the main application file (Flex)	Cairngorm: Avoid ModelLocator not only accessible from the main application file (Flex)
100 3122	Avoid FrontController that adds all its commands within the Controller constructor (Flex)	Cairngorm: Avoid FrontController that adds all its commands within the Controller constructor (Flex)
100 3126	Avoid method with more than one exit point (Flex)	Avoid more than one return path (Flex)
100 3128	Avoid using the with keyword (Flex)	Avoid use of with statement (Flex)
100 3132	Avoid switch statement with too few branches (Flex)	Avoid switch statement with less than 3 cases (Flex)
100 3134	Avoid identical switch cases (Flex)	Avoid identical cases in switch (Flex)
100 3136	Avoid test that does not contain at least one assertion (Flex)	Avoid unit test that does not contain at least one assertion (Flex)
100 3138	Avoid BindingUtils class that uses hard coded strings (Flex)	Avoid BindingUtils class (Flex)
100 3140	Avoid ChangeWatcher class that uses hard coded strings to specify the attribute name, to listen to. Prefer listening to events or setters (Flex)	Avoid use of ChangeWatcher class (Flex)
100 3142	Avoid CairngormEventDispatcher called explicitly (Flex)	Cairngorm: Avoid CairngormEventDispatcher called explicitly (Flex)
100 3144	Avoid more than one reference of ModelLocator per class (Flex)	Cairngorm: Avoid more than one reference of ModelLocator per class (Flex)
100 3148	Avoid clone event not overridden in a custom event (Flex)	Avoid clone method not overridden in a custom event (Flex)
100 3150	Avoid event dispatched in a constructor (Flex)	Avoid event dispatched from a constructor (Flex)
100 3152	Avoid DispatchEvent function not dispatching constant strings (Flex)	Avoid dispatching events using hardcoded strings (Flex)
100 3154	Avoid addEventListener containing hard coded strings (Flex)	Avoid listening to events using hard coded strings (Flex)
100 3156	Avoid type not found within analysis scope (Flex)	Avoid partial source code delivery of application (Flex)
100 3158	Avoid unspecified event type (Flex)	Avoid event without type specified (Flex)
100 3160	Avoid test containing a hard coded boolean value. (Flex)	Avoid conditions with hardcoded boolean values (Flex)
100 3164	Avoid using Object class in fields (Flex)	Avoid dynamic (Object) type fields (Flex)
100 3166	Avoid using Object class in variables (Flex)	Avoid dynamic (Object) type variables (Flex)
100 3170	Avoid using Dictionary class (Flex)	DELETED: Avoid using Dictionary class (Flex)
100 3176	Avoid ArrayElementType metadata not specified for the array-type field (Flex)	Avoid array-type field without ArrayElementType metadata (Flex)
100 3178	Avoid incorrect signature of the message interceptor. (Flex)	Avoid incorrect message interceptor signature (Flex)
100 3180	Avoid packages with misplaced metadata (Flex)	Avoid packages having artifacts with misplaced metadata (Flex)
100 3182	Avoid classes with misplaced metadata (Flex)	Avoid classes having artifacts with misplaced metadata (Flex)
100 3184	Avoid type metadata argument redundant with the handler argument type (Flex)	Avoid redundant type metadata argument for handler type (Flex)

100 3186	Avoid method metadata argument redundant with the handler name (Flex)	Avoid redundant method metadata argument for handler (Flex)
100 3188	Avoid packages with unknown metadata attribute (Flex)	Avoid packages having artifacts with unknown metadata attribute (Flex)
100 3190	Avoid classes with unknown metadata attribute (Flex)	Avoid classes having artifacts with unknown metadata attribute (Flex)
100 3192	Avoid globally bindable classes. (Flex)	Avoid globally bindable classes (Flex)
100 3194	Avoid constructor with a return type (Flex)	Avoid constructor with void return type specification (Flex)
100 3196	Avoid switch statement without a default statement (Flex)	Avoid switch statement without a default case (Flex)
100 3198	Avoid MXML component with more than 1 public variable (Flex)	Avoid MXML component with more than one entry point (Flex)
100 3200	Avoid MXML component with more than 2 public variables (Flex)	Avoid MXML component with more than two entry points (Flex)
100 3204	Avoid too long script block (Flex)	Avoid script block that are too long (Flex)

## 1.1.1-funcrel

### Resolved issues

R&D ID	Ticket ID	Summary
FLEX-136	14266	Unable to download extension via CLI. The specified path, file name, or both are too long.

## 1.1.0-funcrel

### Updates

- New **object icons** have been introduced. See [Flex 1.1](#).
- **FLEX-91** - In previous releases of the extension, **local variables** in the source code were resolved as "FLEX Variable" objects. This behaviour has now been changed and **FLEX Variable objects are no longer resolved**. Therefore, after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ. In addition, the following rules included FLEX Variable objects in their scope and have been modified to use FLEX Getter, Setter, Function and Script objects instead of FLEX Variables. There will be no impact on the number of violations for these rules:

1003032	Avoid unused variable (Flex)
1003046	Avoid too short variable name (Flex)
1003052	Avoid variable names ending with numeric characters (Flex)
1003060	Avoid variable incorrectly named (Flex)
1003166	Avoid using Object class in variables (Flex)
1003174	Avoid * and use strongly typed objects instead in variables (Flex)

- **FLEX-77** - In previous releases of the extension, call links between Flex functions were never resolved, by design. This behaviour has now been changed and links between Flex functions are resolved.
- **FLEX-127** - The use of the \$ sign in the names of functions, fields, getters and setters is now supported and the object will be fully resolved. For example:

```

package {

public class DollarInName {
    mx_internal function get $transform() : int {
        return 0;
    }
    mx_internal function set $transform(int val) : void {
    }
    mx_internal function $transformThat() : void {
    }
}
}

```

- **FLEX-133** - The **RemoteObject** parameter is now resolved correctly as an object. For example, in the following code, all RemoteObject parameters will be resolved:

```

<?xml version="1.0" encoding="utf-8"?>

<s:ProxyFactory xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx"
    xmlns:comp="com.att.ntscp.view.component.*" width="99%"
    >

    <mx:RemoteObject id="SingleStatementMultiline1"
        named="something"
        showBusyCursor="true" />

        <mx:RemoteObject id="MultilineMultiline1" destination="helpDest" showBusyCursor="true">
            <mx:method name="setAttUid" />
            <mx:method name="getCommandHelp" result="onResult(event)" fault="onFault(event)" />
        </mx:RemoteObject>

    <mx:RemoteObject id="SingleStatementSingleline1" named="something" showBusyCursor="true" />

        <mx:RemoteObject id="MultilineMultiline2CommentInBetween" destination="helpDest"
showBusyCursor="true">
            <mx:method name="setAttUid" />
            <!-- no comment! -->
            <mx:method name="getCommandHelp" result="onResult(event)" fault="onFault(event)" />
        </mx:RemoteObject>

    <mx:RemoteObject id="SingleStatementMultiline2" named="something"
        showBusyCursor="true" />

    <mx:RemoteObject id="SingleStatementMultiline2CommentBetweenLines"
        named="something"
        showBusyCursor="true" />

</s:ProxyFactory>

```

- FLEX Getter and Setter objects (new in Flex 1.1.0) will now be included in the scope for violations in the following rules, therefore, after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: increased number of violations providing improved accuracy.

1003014	Avoid nested if statement (Flex)
1003020	Avoid empty method (Flex)
1003026	Avoid complex methods (Flex)
1003030	Avoid function with unused parameter (Flex)
1003054	Avoid local variable hiding a class property (Flex)

1003056	Avoid class name not starting by a majuscule character (Flex)
1003082	Avoid instantiating variable in a loop (Flex)
1003088	Avoid calling Alert.show directly (Flex)
1003110	Avoid function that are too long (Flex)
1003126	Avoid method with more than one exit point (Flex)
1003146	Avoid call to callLater explicitly (Flex)
1003152	Avoid DispatchEvent function not dispatching constant strings (Flex)
1003154	Avoid addEventListener containing hard coded strings (Flex)

## Resolved issues

The **Flex 1.1.0** contains all bug fixes from previous releases. The following table lists all bugs fixed in **Flex 1.1.0** and that are not already fixed in the previous released versions:

R&D ID	Ticket ID	Summary	Comments
FLEX-77	10507	Missing links between Flex functions.	After an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: increased number of links between Flex functions impacting rule results and function point values.
FLEX-101	13921	Flex warn: An unnamed object of type 'FLEX_script' has been detected.	<p>When using previous releases of the extension, warnings were visible in the analysis log as follows:</p> <pre>An unnamed object of type 'FLEX_script' has been detected</pre> <p>This resulted in the creation of FLEX_Script objects with no name. This bug has now been fixed, therefore, after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: all FLEX_Script objects are now resolved correctly with their name and as such the existing unnamed objects will be reported as deleted and the objects now with names will be reported as newly added.</p>
FLEX-102	14008	Analysis does not complete and is stuck processing a file.	-
FLEX-107	-	No violations for "Avoid variable with too short name"	This rule was not producing any violations. This bug has now been fixed, therefore after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: increased number of violations for this rule.
FLEX-124	-	Flex extension does not detect functions that have names with 3 or less characters	Flex functions that have names with 3 characters or less are now correctly resolved as objects. Therefore, after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: increased number of objects and potential for increased number of violations on these objects.
FLEX-125	-	An unnamed object is created when alias is given with single quote.	<p>Aliases defined in code using single quote marks were being resolved as "unnamed" during the analysis - see code example below:</p> <pre>[RemoteClass(alias='example.src.com.company.service.Device')]</pre> <p>This bug has now been fixed, therefore after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: objects are now correctly resolved with their name when single quote marks are used.</p>
FLEX-126	-	Unable to detect object end of Flex_Field	<p>When using previous releases of the extension, sometimes it was not possible to detect the end of objects in the analyzed code. This resulted in:</p> <ul style="list-style-type: none"> <li>objects such as FLEX Fields, Getters, Setters and Functions not being resolved</li> <li>a warning in the analysis logs "Unable to detect object end for Flex_Field and Flex_Class"</li> <li>incorrect bookmarks in code for rule violations.</li> </ul> <p>This bug has been fixed and after an upgrade to Flex 1.1.0 and the generation of a post upgrade snapshot on unchanged source code, results may differ: object ends are detected correctly, thus increasing the number of objects detected and violations are correctly bookmarked in the code.</p>