Best Practice Analyzer 3.0
List of Checked Best Practices

Area	Warning Message	Explanation	Technical ID
Script	MEDIUM - This script calls the CONF function directly. Instead, implement a Configuration Data Class to encapsulate all the configuration properties.	**Avoid using the CONF function inside scripts directly**. This check identifies all the invocations of the CONF function inside the scripts. Implement a Configuration Data Class to encapsulate all the configuration properties, instead of invoking the CONF function directly. Using a Data Class makes your Solution more robust in case of changes in the configuration properties, and allows you to define a default value.	FUNCTION_configurationFunction
Script	LOW - This script contains a constant. Wrap all your constants into a Data Class.	**Avoid using constants inside scripts**. This check identifies all the static values (number or strings) inside scripts. Any fixed or static value that is consistent across a Solution should be contained in a Data Class. This way, your Solution is easier to maintain and readable. For more information, see the Constants section.	FUNCTION_equalOperator
Script	HIGH - This script contains an empty Catch block. Review the script.	**Do not leave the Catch statement block empty**. This check analyzes the content of Catch statements inside scripts. The Catch block should contain the logic to deal with the failure that caused the exception. Leaving it empty might cause unexpected behavior.	SCRIPT_emptyCatch
Script	MEDIUM - This script is too long (more than $scriptLength lines). Refactor your code.	**Avoid long functions**. Avoid implementing long Script Functions to improve maintainability and readability. As a general rule, functions should fulfill one single objective. If your function has multiple purposes, consider breaking it down into multiple private sub-functions. For more information, see the Sizing section.	SCRIPT_scriptSize
Other	MEDIUM - This Data Class contains a function that returns a static Boolean value. Update the logic.	**The functions inside the Data Classes should not return a static Boolean value**. This check analyzes all the Data Class functions and identifies the functions that return a static Boolean value. You should consider updating the logic.	SCRIPT_staticBooleans
Script	MEDIUM - The Variable name $variableName is too short. Choose a longer name.	**Variables should be descriptive and self-explanatory**. This check extracts all the variables whose name is shorter than the one defined in the configuration property bestPracticesAnalyzer.variables.variableNameLength (link to config property explanation). Consider using a variable name that is descriptive and self-explanatory to improve maintainability and readability. For more information, see the Variable section.	VARIABLE_variab
Screen	HIGH - In this Screen the Remember Variables option is selected. If possible, deselect it and use Process Variables instead.	**Do not use the Remember Variable option if not strictly necessary**. This check identifies all the Screens that have the **Remember Variables** option selected. The Remember Variable option can introduce performance degradation in the screen processing time, therefore it should be used only if strictly needed.	BO_rememberVariablesOnScreens
Other	LOW - This Business Object is probably used for testing. In that case, remove it when no longer necessary.	**Test Business Object should not be committed in production environment**. Business Objects and their related versions take up disk space and memory. If a Business Object is used only for testing purposes, it should not be imported in production environments. This check is performed on all Business Objects and extracts the ones whose name contains the word *test*.	BO_testBO
Script	LOW - This script calls the DELETEDIRECTORY function. Make sure it's used in the right place with the right input parameters.	**Avoid using the DELETEDIRECTORY function if not strictly necessary**. This check identifies all the invocations of the DELETEDIRECTORY function inside scripts. The DELETEDIRECTORY function might potentially remove vital folders. Make sure it is used in the right place with the right input parameters.	FUNCTION_deleteDirectoryFunction
Script	HIGH - Depending on the function, see the Distributed Functions appendix.	**Cluster-related checks**. This check scans all scripts and detects the invocations to Cluster-related critical functions. Messages specific to the function: Distributed Functions.	FUNCTION_distributedFunction
Script	LOW - Use the $recommendedFunction function instead of $functionName function.	**Unrecommended Functions**. This check scans all scripts and detects the invocations to non-recommended functions. For more information, see the Scripting Best Practices article. See the full list of checks in the List of Functions and Operators appendix.	FUNCTION_notRecommendedFunction
Screen	MEDIUM - This Screen contains a script that calls the TRANSLATE function. Use a Label Business Object instead (the translation is handled within the Appway Platform).	**Do not use the TRANSLATE function inside Screens**. The Best Practice Analyzer identifies all the invocations to the TRANSLATE function inside Screen scripts. When a label is rendered inside the Screen, it is already automatically translated in your browser language. Using the TRANSLATE function is a functional bad practice (as it forces a translation into a specific language and ignores	FUNCTION_translateInScreens

		the browser language settings) and a performance bad practice (as the operation is not needed).	
Other	HIGH - This Integration Link script contains business logic. Move the business logic into a Script Business Object and invoke it from the Integration Link script.	**IntegrationLink scripts should point to a Script Function**. This check verifies that every script inside the Integration Link is not longer than one line. Integration Links run in one version only and they can be called by Process Instances using multiple different version filters. If your Integration Link contains complex script code in any of the scriptable fields, such script code will be running always in Latest Committed version regardless of the caller process version filter. This could result in a serious problem, as the old Process Instances might send data that cause an exception the Integration Link script. Fixing the exception for both old and new Processes is not possible if the big script logic is in the Integration Link itself, as this is running on the Latest Committed version. The recommended approach is to wrap the logic in a Script Function with a stable and generic interface that can be invoked by inheriting the Process Instance version filter. See details in the Integration Link Versioning article.	SCRIPT_integrationLinkScriptSize
Other	HIGH - This Data Class function uses the $super variable to access the parent class. Use the SUPER() function instead.	**Always use SUPER instead of $super**. This check identifies all the occurrences of the $super variable inside Data Classes functions. In order for your code to be compliant with multiple inheritance, you are recommend to use the SUPER() function instead of the $super variable. The SUPER() function accepts two arguments: The Data Type of the specific parent class that you want to use. The Object that you want to cast to the parent Data Type. See more information on SUPER and $super.	SCRIPT_superUsages
Other	HIGH - This URL is not context path aware. Add the context path by using the CONTEXTPATH function.	**URLs should be context path-aware**. This check verifies that all the specified URLs use the CONTEXTPATH() function. It ignores all the absolute URLs (starting with "http://" and "https://"). Every relative URL should include the reference to the context path. If the context path is not specified, the solution might not work if installed on an environment in which the context path is not null.	URL_contextPath
Other	HIGH - The Receive Message Intermediate Event has a message filter defined. Create a specific queue instead.	**The Receive Message Intermediate Event should not define a message filter**. This check identifies all the Receive Message Intermediate Events that have a Filter property defined. When you use a filter, the component listens and processes all the messages, and then it filters the relevant one. Using a specific queue is more efficient.	WM_receiveIntermediateEventMessageFilter
Other	HIGH - The Receive Message Intermediate Event has a dynamic queue defined. If possible, avoid using a dynamic queue.	**Avoid using dynamic queues inside the Receive Message Intermediate Event if not necessary**. This check identifies all the Receive Message Intermediate Events that have message queues defined dynamically. Appway evaluates the message queue expression every time it checks whether a message has arrived. To evaluate the expression, Appway loads the value store for this Process Instance into the memory, so that it has access to the variables in the current scope. This can have a negative impact on performance in several ways. Since Appway checks whether a message has arrived every 5 minutes by default, therefore this value store is kept in memory almost all the time. If there are many Process Instances with a token sitting on this Intermediate Event, the overall memory usage of Appway is increased significantly. On the other hand, if Appway is configured to cache value stores in memory for a shorter time, the affected value stores are uploaded from the disk every 5 minutes, potentially leading to high CPU usage and I/O traffic.	WM_receiveIntermediateEventQueueName
Screen	HIGH - This Screen contains a Raw HTML Screen Component. Replace it with the Design System and Style Business Objects.	**Avoid using Raw HTML Screen Components**. This check extracts all the occurrences of Raw HTML Screen Components inside Screens: HTML Head Style, HTML Head Script, HTML Head Link, HTML Head Title, HTML Body Script, HTML MetaTag. These components allow adding custom CSS and Javascript. A solution using custom CSS/JS relies on the inner structure (e.g. DOM and HTML classes) of Screen Components. When you update the solution, if the component inner structure changed, your custom code most likely no longer works, and this also affects running processes. Therefore, you need to patch Screen Business Objects thus increasing the upgrade effort dramatically. Moreover, custom CSS/JS can interfere with Appway code, e.g. in client-side rendering.	SC_rawHtmlComponent
Other	MEDIUM - This Process contains a Placeholder task. Replace it before releasing the Process.	**Processes must not contain placeholders in production environments**. This check extracts all the Processes that contain at least one Placeholder task. While Placeholders tasks are useful in the design phase, they must be replaced before importing the solution to production.	WM_componentPlaceholder
Screen	MEDIUM - This Screen Component is using Legacy Styles. Use the Migration Tool to remove them and apply Style Business Objects.	**Screen Components should use Style Business Objects where possible**. This check identifies all the Screen Components for which the option **Use Legacy styles** option is selected. These Screen Components can be styled using the respective Style Business Objects. You can use the Migration Tool to remove legacy styles.	SC_componentLegacyStyle
Screen	MEDIUM - The Text Screen Component does not contain a	**All the Text components inside Screens should be wrapped in a Label Business Object**. This check is performed on the following	SC_missingLabels

	Label Business Object or an expression for the Text property. Add one.	Screen Components: Button Text components (Text, Heading1 and so on, Paragraph, Preformatted Text, Code) Link Info Box Checkbox/Radiobutton It detects all the occurrences of static text (excluding Label Business Objects or Script Functions). Including static text inside a Screen Component makes it impossible to translate it into multiple languages. Wrap all your visible text inside Label Business Objects (more info).	
Screen	HIGH -The Phase property of the Script Screen Component is set to Always. Consider changing this value.	**Script Screen Components should not be executed in all phases**. This check detects all the occurrences of Script Screen Components for which the Phase property is set to Always, meaning that the same script is executed in all four phases. Generally, it is not necessary to execute the same script in all phases, and should be avoided to reduce the computation time of Screens (more info).	SC_alwaysPhase
Screen	MEDIUM - The Appway Test ID of a Screen Component used in a For/ForEach/While container is static. Use a dynamic expression instead.	**Test IDs should use a dynamic expression**. If Screen Components contain a For/ForEach/While container, a static Test ID will be repeated for each element, thus causing errors. We recommend using a dynamic expression for these Test IDs. See the End-to-End Testing documentation.	SC_appwayTestingIdInLoops