

MISRA Compliance

The Parasoft MISRA Compliance artifact is a set of assets for your DTP infrastructure that enable you to readily demonstrate compliance with MISRA development guidelines. The MISRA Compliance artifact can be adapted to support any version of the MISRA standard, but it is configured by default for MISRA C:2012. This artifact ships with the [Automotive Compliance Pack for DTP 5.4.1](#). Contact your Parasoft representative for information about downloading and licensing the compliance pack.

In this section:

- [Background](#)
- [Prerequisites](#)
- [Process Overview](#)
- [MISRA Compliance Extension Assets](#)
- [Installing and Deploying the DTP Workflow](#)
- [Adding the MISRA Dashboard](#)
- [Viewing MISRA Compliance Widgets](#)
- [Viewing MISRA Compliance Reports](#)
- [Profile Configuration](#)
- [Upgrading MISRA Compliance](#)

Background

MISRA began as a set of technical guidelines to help organizations create safety-critical software for automotive applications. The standard has since been adopted by embedded software development organizations in other safety-critical industries. MISRA C:2012 is the most recent implementation for development using the C programming language and includes 159 base guidelines. Amendment 1 is a supplemental set of 14 guidelines that expanded the total to 173.

One of the challenges associated with achieving MISRA compliance has been the lack of a standardized mechanism for demonstrating compliance. To address this issue, MISRA published "[MISRA Compliance 2016: Achieving Compliance with MISRA Coding Standards](#)," which provides a more concrete definition of "MISRA compliance" and identifies several deliverables required for demonstrating compliance with the MISRA standard.

The Parasoft MISRA Compliance artifact configures code analysis to run against MISRA guidelines and conforms the data to meet the following reporting specifications defined in MISRA Compliance 2016.

Guideline Enforcement Plan

A guideline enforcement plan (GEP) lists each MISRA guideline to indicate how compliance is checked. In the context of MISRA compliance with Parasoft, the GEP maps MISRA guidelines to Parasoft code analysis rules and DTP functionality.

Guideline Re-categorization Plan

A guideline re-categorization plan (GRP) documents agreed-upon changes to how MISRA guidelines are categorized. Guidelines are categorized as Mandatory, Required, and Advisory. A fourth category, Disapplied, may also be used for instances in which the guideline truly does not apply.

All mandatory guidelines must be followed to achieve compliance. Required guidelines should be followed, but documented exceptions are allowed (see Deviations report). Advisory guidelines are considered best practice.

Required and Advisory guidelines can be re-categorized into to stricter categories (e.g., Advisory to Required), but only Advisory guidelines can be re-categorized into less strict categories (i.e., to Disapplied).

Deviations Report

A "deviation" is a documented violation of the guideline and supporting rationale for allowing the violation to remain. In the context of MISRA compliance with Parasoft, deviations take the form of suppressed code analysis violations. Your project can have deviations and still be considered compliant if the deviations are documented in the report and do not impact safety.

Guideline Compliance Summary

A guideline compliance summary (GCS) is the primary record of overall project compliance. The GCS includes an entry for each guideline, its level of compliance, any deviations and/or re-categorizations, etc. The [MISRA Compliance Report](#) shipped with the Parasoft MISRA Compliance extension fulfills this requirement.

See [MISRA Compliance 2016: Achieving Compliance with MISRA Coding Standards](#) for additional details and information.

Prerequisites

In addition to the MISRA Compliance artifact, the following Parasoft products are required:

- DTP and DTP Enterprise Pack 5.4.1 with Enterprise license
- C/C++test 10.4.1 (Standard or Professional) with the Flow Analysis license feature enabled

Process Overview

1. Analyze code with C/C++test using the MISRA C:2012 test configuration (shipped with C/C++test) and report violations to DTP.
2. Install the [Automotive Compliance Pack for DTP 5.4.1](#) into DTP Extension Designer, which also installs the MISRA Compliance assets.
3. Deploy the MISRA DTP Workflow to your DTP environment and add the MISRA dashboard and widgets to your DTP interface.
4. Interact with the widgets and reports to identify code that needs to be fixed, as well as print out the reports for auditing purposes.



Achieving 100% Compliance

According to MISRA C:2012, there are four guidelines that cannot be statically analyzed. As a result, DTP will report 100% compliance against 169 guidelines.

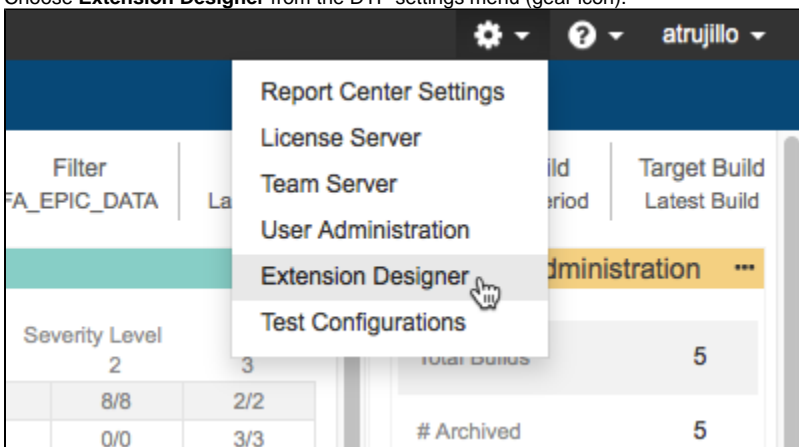
MISRA Compliance Extension Assets

The Parasoft MISRA Compliance artifact helps you create the documentation required for demonstrating compliance with MISRA C:2012. The following assets are included:

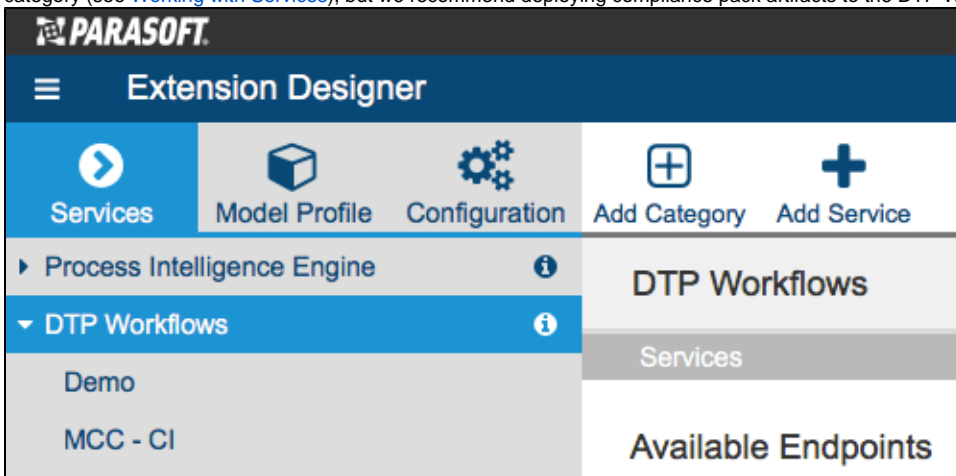
- **Compliance categories and guidelines:** These files add the MISRA Compliance option to DTP interfaces, such as widget configuration settings.
- **Dashboard template:** This file adds enables you to add the MISRA C:2012 dashboard that includes a pre-defined set of MISRA-related widgets.
- **Model and profile:** These files drive the report compliance reports necessary for demonstrating MISRA compliance. See [Profile Configuration](#) for additional information.
- **DTP Workflow:** This is the DTP Enterprise Pack artifact that includes the widgets, reports, and processing logic that show violations in the context of MISRA guidelines.

Installing and Deploying the DTP Workflow

1. The MISRA Compliance artifact is installed as part of the Automotive Compliance Pack. See [Installation](#) for instructions.
2. Choose **Extension Designer** from the DTP settings menu (gear icon).



- Click the **Services** tab and choose a service category. You can deploy the artifact to any service category you want. You can also create a new category (see [Working with Services](#)), but we recommend deploying compliance pack artifacts to the DTP Workflows service category.



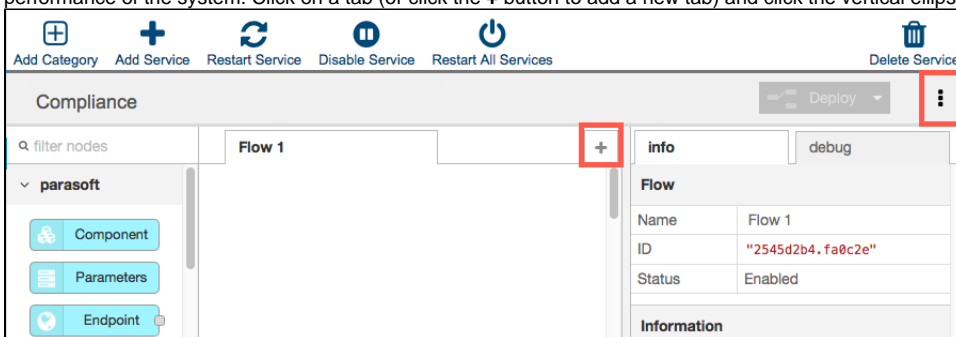
- You can deploy the artifact to an existing service or add a new service. The number of artifacts deployed to a service affects the overall performance. See [Extension Designer Best Practices](#) for additional information. Choose an existing service and continue to step 6 or click **Add Service**.
- Specify a name for the service and click **Confirm**.

Add Service

Name:
 *

Category:

- The tabs interface allows you to organize your artifacts within the service. Organizing your artifacts across one or more tabs does not affect the performance of the system. Click on a tab (or click the + button to add a new tab) and click the vertical ellipses menu.



- Choose **Import> Library> Workflows> Automotive> MISRA Compliance** and click anywhere in the open area to add the the artifact to the service.
- Click **Deploy** to finish deploying the artifact to your DTP environment.
- Return to DTP and refresh your dashboard.

You will now be able to add the MISRA dashboard and widgets, as well as access MISRA reports.

Adding the MISRA Dashboard

The MISRA dashboard is configured to show custom widgets shipped as part of the MISRA artifact. The dashboard also contains select native DTP widgets configured to show code analysis data within the context of MISRA C:2012. The information in this section is also covered in the [Adding Dashboards](#) chapter.

- Click **Add Dashboard** and specify a name when prompted.

2. (Optional) You can configure the default view for the dashboard by specifying the following information:
 - Choose the filter associated with your project in the filter drop-down menu. A filter represents a set of run configurations that enabled custom views of the data stored in DTP. See [DTP Concepts](#) for additional information.
 - Specify a range of time from the Period drop-down menu.
 - Specify a range of builds from the Baseline Build and Target Build drop-down menus.
3. Enable the **Create dashboard from a template** option and choose MISRA C Compliance from the drop-down menu.

Add Dashboard

Name:

Filter:

Period:

Baseline Build:

Target Build:

Create an empty dashboard

Create dashboard from a template:

Copy an existing dashboard:

Follow a shared dashboard:

4. Click **Create** to finish adding the dashboard.

See [Viewing MISRA Compliance Widgets](#) for information about understanding the widgets shipped with the MISRA C:2012 artifact.

Manually Adding Widgets to Your Existing Dashboard

You can also add the MISRA widgets shipped with the artifact to an existing dashboard. See [Adding Widgets](#) for general instructions on adding widgets to a dashboard. After deploying the artifact, the MISRA widgets will appear in the MISRA category in the Add Widget overlay:

Add Widget

MISRA	MISRA Compliance - Guidelines by Status	MISRA Compliance - Guidelines by Status 2 x 1 Displays the number of guidelines by compliance status, the total number of violations, and the total number of deviations. Title: <input type="text" value="MISRA Compliance - Guidelines by Status"/> Filter: <input type="text" value="Dashboard Settings"/> Target Build: <input type="text" value="Dashboard Settings"/> Category: <input type="text" value="All"/> Compliance Profile: <input type="text" value="MISRA C 2012"/>
OWASP	MISRA Compliance - Percentage	
SEI CERT	MISRA Compliance - Status	
Build Results	MISRA Violations by Category - TreeMap	
Code		
Compliance		
Coverage		
Diagnostics		
Metrics		
Process Intelligence		
Static Analysis		
Tests		
Custom		

Specify the following information when adding the MISRA Compliance - Percentage, MISRA Compliance - Status, and MISRA Violations by Category - TreeMap widgets:

Title	You can rename the widget in the Title field.
Filter	Choose a specific filter or Dashboard Settings from the drop-down menu. See Creating and Managing Filters for additional information.
Target Build	Choose a specific build from the drop-down menu. The build selected for the entire dashboard is selected by default. See Using Build Administration for additional information about understanding builds.
Category	This setting is available in the MISRA Compliance - Guidelines by Status and MISRA Compliance - Status widgets. Choose an individual category or All from the drop-down menu.
Compliance Profile	Specify a compliance profile (see Profile Configuration). The compliance profile data is used to generate some of the compliance reports.

Viewing MISRA Compliance Widgets





Each widget provides a different view of the test and analysis data to help you achieve MISRA compliance. The following widgets are shipped with the MISRA Compliance DTP Workflow.

MISRA Compliance - Status Widget

This widget shows you the general state of compliance. You can add multiple instances of the widget configured to use a different profile, e.g., a profile with disappplied guidelines, to view your current compliance status. Click on the widget to open the [MISRA Compliance Report](#).

The widget can show five possible states:

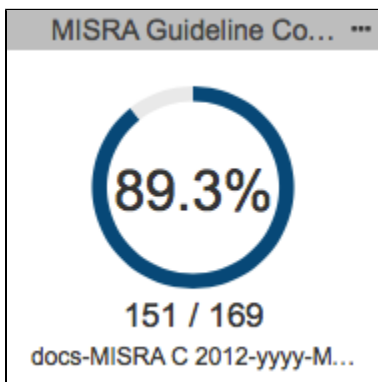
Compliant Code meets all guidelines with no deviations or changes to the guideline categories.	
--	--

	<div data-bbox="1101 130 1474 506"> <p>MISRA Compliance ...</p> <p>Category: All</p>  <p>Compliant</p> <p>Build: atm-2017-06-05 Compliance: MISRA - Comp wit...</p> </div>
<p>Missing rule(s) in analysis</p> <p>Parasoft code analysis rules documented in your profile were not included in the specified build. Make sure all rules are enabled in C/C++test and re-run analysis.</p>	<div data-bbox="1101 541 1474 917"> <p>MISRA Compliance ...</p> <p>Category: All</p>  <p>Missing rule(s) in analysis</p> <p>Build: bank-2017-04-11 Compliance: MISRA C 2012</p> </div>
<p>Compliant with Deviations</p> <p>Code meets all guidelines, but deviations have been applied. Deviations are violations that you have determined to be acceptable (see Deviations Report for additional information about deviations).</p>	<div data-bbox="1101 955 1474 1331"> <p>MISRA Compliance ...</p> <p>Category: All</p>  <p>Compliant with Deviations</p> <p>Build: atm-2017-06-05 Compliance: MISRA - Comp wit...</p> </div>
<p>Compliant with Violations</p> <p>Code meets all mandatory and required guidelines, but contains violations for advisory guidelines.</p>	<div data-bbox="1101 1369 1474 1745"> <p>MISRA Compliance ...</p> <p>Category: All</p>  <p>Compliant with Violations</p> <p>Build: atm-2017-06-05 Compliance: MISRA - Comp wit...</p> </div>
<p>Not Compliant</p> <p>Code does not meet all mandatory and/or required guidelines.</p>	



MISRA Compliance - Percentage Widget

This widget shows the completeness of MISRA compliance as a percentage. Completeness is based on number of guidelines being enforced in the [profile](#). Click on the widget to open the [MISRA Compliance Report](#).



MISRA Compliance - Guidelines by Status

This widget shows the compliance status for individual guideline categories (Mandatory, Required, Advisory) or for all categories.



The pie chart can represent up to five different guideline statuses for the selected category:

Green	Guidelines that your code is compliant.
Yellow	Guidelines that your code is deviating from but are still considered compliant. A deviation is when the guideline is not being followed according to the Parasoft static analysis rule, but is considered acceptable because it does not affect the safety of the software. Deviations represent Parasoft static analysis rules that have been suppressed.
Orange	Guidelines that your code is considered compliant with, even though the static analysis rules that enforce them contain violations. Only advisory guidelines can have this status.

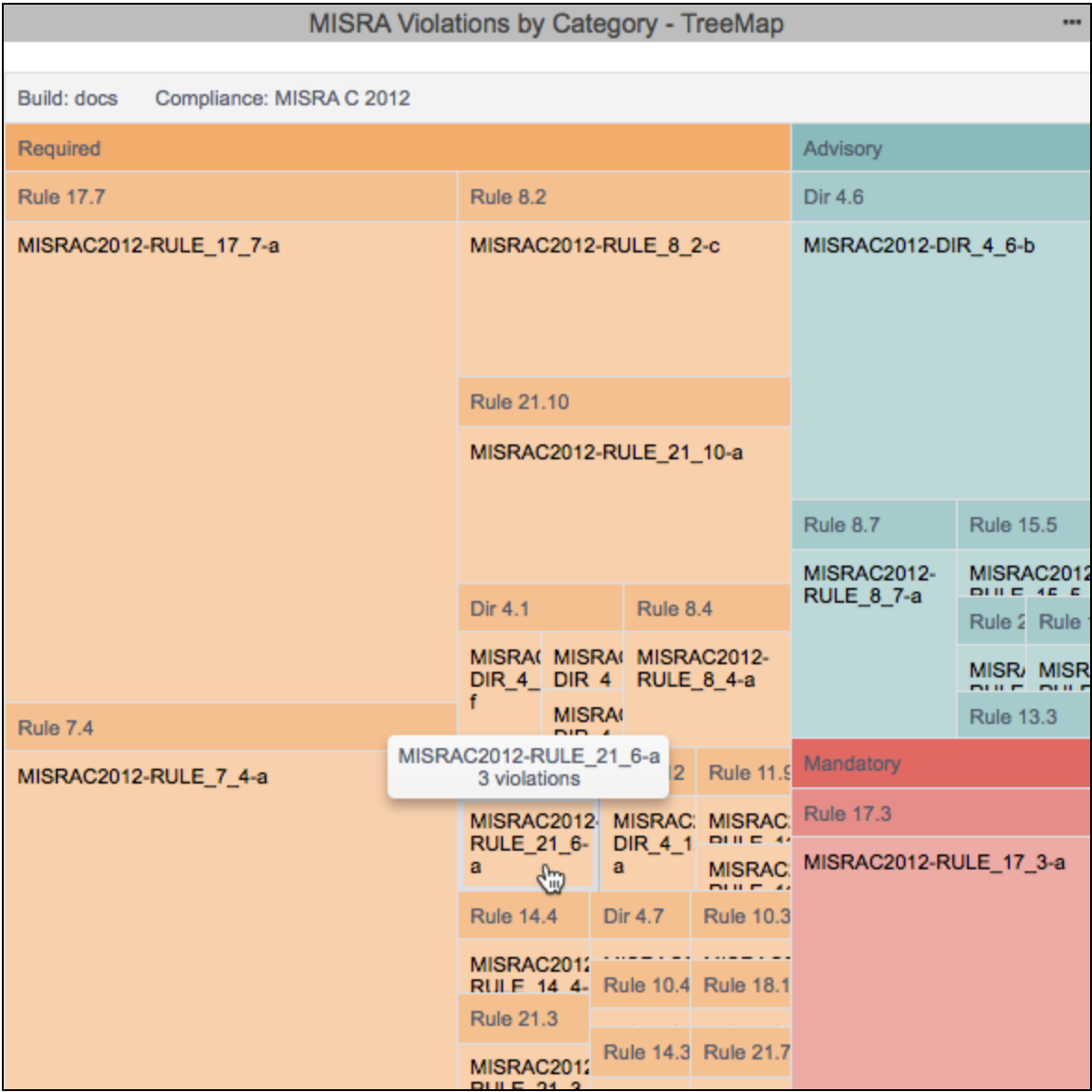
Red	Guidelines that your code is not compliant with.
Ma ro on	Guidelines that are specified in the profile , but the Parasoft rule that enforces the guideline are missing.

You can perform the following actions:

- Mouse over a pie slice to view details.
- Click on a section to open the [MISRA Compliance report](#) filtered by the category and status.
- Click on the number of violations counter to open the [MISRA Compliance report](#) filtered by the category and status.
- Click on the number of deviations counter to open the [Deviations Report](#) filtered by the category selected in the widget.

MISRA Violations by Category - TreeMap Widget

This widget provides a representation of the highest concentration of static analysis violations per MISRA category (mandatory, required, and advisory). The widget also shows the guidelines (e.g., Dir 4.6, Rule 14.3, etc.) within each category in which violations were reported. Finally, the Parasoft rule(s) enforcing each guideline are also presented. Tiles are proportional to the number of static analysis violations reported for each rule.



The widget uses the hierarchy established in the [model profile](#) to correlate rules, guidelines, and categories. You can mouse over a tile in the widget to view the number of violations associated with each rule-guideline-category.

Click on a rule to see the violation in the [Violations Explorer](#).

Viewing MISRA Compliance Reports

The MISRA Compliance Report provides an overview of your MISRA compliance status and serves as the primary document for demonstrating compliance.

[Download PDF](#)

MISRA Compliance Report

Filter: misra **Target Build:** docs **Compliance Profile:** MISRA C 2012 **Analysis Tool:** Parasoft C++test 10.4 **Compiler:** ADD COMPILER
Revision Date: 2018-12-06

Project Compliance: ✖ **Not Compliant**

[Guideline Enforcement Plan](#) [Guideline Re-categorization Plan](#) [Deviation Report \(Total: 1\)](#) [Build Audit Report](#)

Category: Compliance:

Guideline	MISRA Category	Compliance	# of Violations	# of Deviations	
				In-Code Suppressions	DTP Suppressions ↓
Rule 15.5	Advisory	⚠ Compliant with Violations	2	0	1
Dir 4.1	Required	✖ Not Compliant	4	0	0
Dir 4.2	Advisory	✔ Compliant	0	0	0
Dir 4.3	Required	✔ Compliant	0	0	0
Dir 4.4	Advisory	✔ Compliant	0	0	0
Dir 4.5	Advisory	✔ Compliant	0	0	0
Dir 4.6	Advisory	⚠ Compliant with Violations	15	0	0
Dir 4.7	Required	✖ Not Compliant	1	0	0

You can perform the following actions:

- Use the drop-down menus to sort by MISRA Category or by Compliance status.
- Click on a guideline link in the Guideline column to open the [Guideline Enforcement Plan](#). The link goes directly to the specific guideline so that you can review the Parasoft code analysis rule or rules enforcing the guideline.
- Click on a link in the # of Violations, In-Code Suppression, or DTP Suppressions column to view the violations in the [Violations Explorer](#).
- Open one of the MISRA Compliance sub-reports.
- Click **Download PDF** to export a printer-friendly PDF version of the report data.

The MISRA Compliance Report contains the following sub-reports:

- [Guideline Enforcement Plan](#)
- [Guideline Re-categorization Plan](#)
- [Deviations Report](#)

Guideline Enforcement Plan

The Guidelines Enforcement Plan (GEP) shows which static analysis rules are used to enforce the MISRA guidelines. It is intended to describe how you are enforcing each guideline.

This report uses the data specified in the compliance profile (see [Profile Configuration](#)). In the profile, you can add notes to the Compiler field, such as "no errors" or specific compiler settings that will be applied, to document your plan. These notes appear in the Compiler column.

The Analysis Tool column should refer to the static analysis rule. The Manual Review column should contain any manual verifications that will be performed in addition to the automated checks applied by the compiler and analysis tool.

MISRA Guideline Enforcement Plan

Compliance Profile: MISRA C 2012 Analysis Tool: Parasoft C++test 10.4 Compiler: ADD COMPILER Revision Date: 2018-12-06

Guideline	Category	Description	Compiler	Analysis Tool	Manual Review
Dir 1.1	Required	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood			Peer code review procedure + Parasoft DTP Change Reviews
Dir 2.1	Required	All source files shall compile without any compilation errors	No errors		Review compilation errors
Dir 3.1	Required	All code shall be traceable to documented requirements			Parasoft DTP for traceability between tests and requirements and tests and code
Dir 4.1	Required	Run-time failures shall be minimized		MISRA C2012-Dir_4_1-a MISRA C2012-Dir_4_1-b MISRA C2012-Dir_4_1-c MISRA C2012-Dir_4_1-d MISRA C2012-Dir_4_1-e MISRA C2012-Dir_4_1-f MISRA C2012-Dir_4_1-g MISRA C2012-Dir_4_1-h MISRA C2012-Dir_4_1-i MISRA C2012-Dir_4_1-j	Undecidable - review finding and manage in Parasoft DTP
Dir 4.2	Advisory	All usage of assembly language should be documented		MISRA C2012-Dir_4_2-a	Undecidable - review finding and manage in Parasoft DTP
Dir 4.3	Required	Assembly language shall be encapsulated and isolated		MISRA C2012-Dir_4_3-a	Undecidable - review finding and manage in Parasoft DTP

Guideline Re-categorization Plan

If you changed any of the MISRA guideline categories (see [Profile Configuration](#)), they will be processed and displayed in this report. Refer to the MISRA standard for additional information about guideline re-categorization plans.

MISRA Guideline Re-categorization Plan

Compliance Profile: MISRA C 2012 Analysis Tool: Parasoft C++test 10.4 Compiler: ADD COMPILER Revision Date: 2018-12-06

MISRA			
Guideline	Description	Original Category	Revised Category
Dir 1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood	Required	Required
Dir 2.1	All source files shall compile without any compilation errors	Required	Required
Dir 3.1	All code shall be traceable to documented requirements	Required	Required
Dir 4.1	Run-time failures shall be minimized	Required	Required
Dir 4.2	All usage of assembly language should be documented	Advisory	Advisory
Dir 4.3	Assembly language shall be encapsulated and isolated	Required	Required
Dir 4.4	Sections of code should not be "commented out"	Advisory	Advisory
Dir 4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous	Advisory	Advisory
Dir 4.6	typedefs that indicate size and signedness should be used in place of the basic numerical types	Advisory	Advisory
Dir 4.7	If a function returns error information, then that error information shall be tested	Required	Required
Dir 4.8	If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden	Advisory	Advisory

Deviations Report

Your code can contain violations and still be MISRA-compliant as long as the deviations from the standard are documented and that the safety of the software is unaffected. Deviations are code analysis rules that have been suppressed either directly in the code or in the DTP Violations Explorer. See the C++-test documentation for details on suppressing violations in the code. See [Suppressing Violations](#) in the Violations Explorer documentation for information about suppressing violations in DTP.

Suppressions versus the Do Not Show priority


In versions of DTP prior to 5.4, the only way to suppress static analysis violations was to set the priority to Do Not Show. This approach hides violations, but is not a true, programmatic suppression. You can convert violations flagged with the Do Not Show priority into true suppressions. See [Upgrade Notes](#) in the 5.4.0 release notes for details.


Click on the **Deviations Report** link in the MISRA Compliance report to open the Deviations Report.


MISRA Deviation Report


Filter: atm-for-misra-c Target Build: atm-for-misra-c-2018-11-26 Compliance Profile: MISRA C 2012 Analysis Tool: Parasoft C++test 10.4 Compiler: ADD COMPILER
Revision Date: 2018-12-06


Category: Only Deviations:


Dir 1.1 (Required) Any implementation-defined behaviour on which the output of the program depends shall be documented and understood  - No Rules Enabled


Dir 2.1 (Required) All source files shall compile without any compilation errors  - No Rules Enabled


Dir 3.1 (Required) All code shall be traceable to documented requirements  - No Rules Enabled


Dir 4.1 (Required) Run-time failures shall be minimized  - No Deviations


Dir 4.2 (Advisory) All usage of assembly language should be documented  - No Deviations


Dir 4.3 (Required) Assembly language shall be encapsulated and isolated  - No Deviations


Dir 4.4 (Advisory) Sections of code should not be "commented out"  - No Deviations


Dir 4.5 (Advisory) Identifiers in the same name space with overlapping visibility should be typographically unambiguous  - No Deviations


Dir 4.6 (Advisory) typedefs that indicate size and signedness should be used in place of the basic numerical types  - No Deviations


Dir 4.7 (Required) If a function returns error information, then that error information shall be tested  - No Deviations

Dir 4.8 (Advisory) If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden  - No Deviations

Dir 4.9 (Advisory) A function should be used in preference to a function-like macro where they are interchangeable  - No Deviations

Dir 4.10 (Required) Precautions shall be taken in order to prevent the contents of a header file being included more than once  - No Deviations

Dir 4.11 (Required) The validity of values passed to library functions shall be checked  - No Deviations

Dir 4.12 (Required) Dynamic memory allocation shall not be used  - 1 Deviations

Rule ID: MISRAC2012-DIR_4_12-a
Deviation Type: DTP Suppression
Action:
Suppression Reason: legacy code
Suppression Author: admin

The Deviations Report shows all guideline IDs and headers, but guidelines that have been suppressed will show additional information.

You can filter the report by MISRA category, as well as enable the **Only Deviations** option to only show deviations.

Build Audit Report

The Build Audit Report shows an overview of code analysis violations, as well as test results and coverage information, associated with the build. This report also allows you to download an archive of the data, which is an artifact you can use to demonstrate compliance with MISRA during a regulatory audit.

Runs											
Download Archive ▾											
To group by a specific column, drag and drop the desired column to this area.											
Run Configuration Attributes						Run Information					
Run ID ▾	Run Confi... ▾	Setup P... ▾	Project ▾	Test Co... ▾	Se...	Machine ▾	User ▾	Run Date ▾	Run Type ▾	Reports	
2014	105	0	atm	MISRA C 2012 Compliance	\$(s... nch)- win32_x86_64	DTP Engine for C/C++	cerberus	igarg	2017-05-26 13:27:34	Static Analysis	XML HTML PDF

In order to download an archive, the build has to be locked. See [Build Audit Report](#) for additional details about this report.

Profile Configuration

The MISRA Compliance DTP Workflow ships with a model profile (see [Working with Model Profiles](#)) configured to monitor compliance with MISRA C:2012. The profile includes information necessary for generating compliance reports (see [Viewing MISRA Compliance Reports](#)), such as fields for specifying your compiler and guideline categorization and re-categorization. You can modify the profile if you want to re-categorize guidelines to meet your specific goals or specify additional metadata for your reports. Changes will be reflected in the [Guideline Re-categorization Plan](#).

Extension Designer			
MISRA C 2012			Edit
Profile Attributes			
MISRA Compliance:	MISRA C 2012		
Compiler:	GNU GCC 6.x (x86_64)		
Analysis Tool:	Parasoft C++test 10.4		
Revision Date:	2018-12-06		
Profile Data			
Guideline	Original MISRA Category	MISRA Category	Description
Dir 1.1	Required	Required	Any implementation-defined b...
Guideline:	Dir 1.1		
Original MISRA Category:	Required		
MISRA Category:	Required		
Description:	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood		
Compiler:			
Parasoft Category ID:	MISRAC2012-DIR_1_1		
Parasoft Rule ID(s):			
Manual Review:	Peer code review procedure + Parasoft DTP Change Reviews		
Dir 2.1	Required	Required	All source files shall compile ...
Dir 3.1	Required	Required	All code shall be traceable to ...

We recommend creating a copy of the default profile and modifying the copy.

1. Click **Export Profile** to download a copy.
2. Rename the copy and click **Import Profile**.
3. Browse for the copy and confirm to upload.
4. Click on a guideline and specify your changes. If you are changing MISRA categories, the following strings are acceptable:
 - Mandatory
 - Required
 - Advisory

- Disapplied

Dir 4.7	Required	Required	If a function returns e...	+
Guideline:	Dir 4.7 *			
Original MISRA Category:	Required			
MISRA Category:	Required			
Description:	If a function returns error information, then *			
Compiler:				
Parasoft Category ID:	MISRA2012-DIR_4_7			
Parasoft Rule ID:	MISRA2012-DIR_4_7-a MISRA2012-DIR_			
Manual Review:	Undecidable - review finding and manage i			

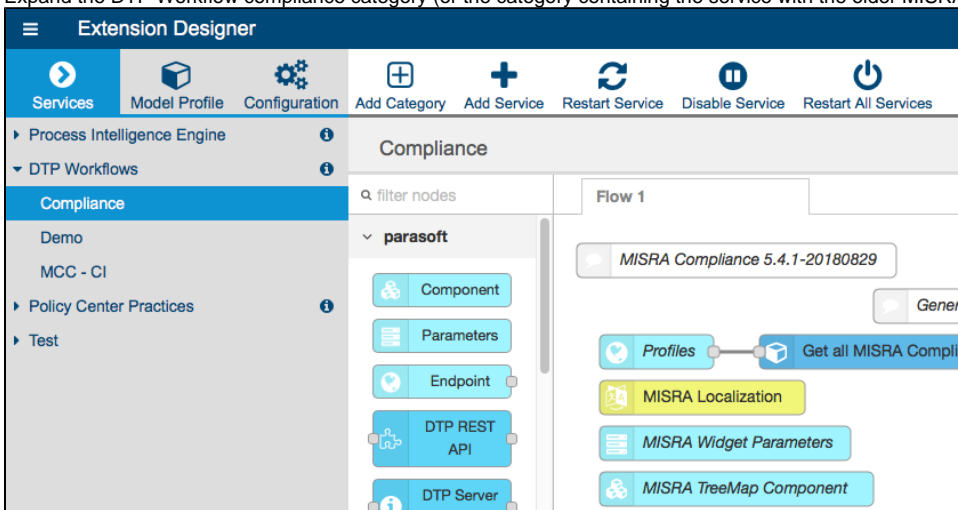
Making Mandatory and/or Required guidelines less strict will result in an invalid use of the guideline in the [Parasoft Guideline Re-categorization Plan Report](#) per the requirements outlined by MISRA (see [Guideline Re-categorization Plan](#)). Making required and advisory guidelines stricter is acceptable.

5. Click **Save**.

Upgrading MISRA Compliance

You should update any extensions when you upgrade DTP. Extensions are designed to be forward compatible, but Parasoft does not guarantee that older extensions will function as expected with newer versions of DTP. We strongly recommend installing the latest version of the artifact and removing the older version to ensure proper functionality.

1. Install the latest version of the Automotive Compliance Pack as described in the [Installation](#) section.
2. Open **Extension Designer** from the DTP settings menu (gear icon) and click the **Settings** tab.
3. Expand the DTP Workflow compliance category (or the category containing the service with the older MISRA artifact) and click on the service.



4. Click on the tab containing the MISRA flow and delete all nodes. You can use your mouse to click and drag over all nodes or use the select-all keyboard shortcut.
5. Import the new MISRA Compliance artifact by choosing **Import > Library > Workflows > Automotive > MISRA Compliance** from the vertical ellipses menu.
6. Click **Deploy** to finish deploying the newer MISRA components.
7. Click the **Configuration** tab and click the delete button (trash icon) for the older extension.

