

DTP Enterprise Pack

DTP Enterprise Pack is a suite of process intelligence tools for defining how risk associated with the software under development is measured, reported, and addressed. The tools enable you to configure development policies and services that define risk thresholds and possible actions when those thresholds are crossed. Development Testing Platform collects raw observations from various infrastructure sources throughout the SDLC, correlates and processes it according to your policies. The combination of policies and data-processing artifacts can automatically populate reports in Development Testing Platform, prioritize remediation tasks, trigger processes in third-party systems, etc.

DTP Enterprise Pack includes the following tools:

- **Extension Designer** (formerly PIE Slice Designer): This component enables you to define Process Intelligence logic that processes data. The Extension Designer leverages Development Testing Platform APIs, as well as APIs from third-party systems (e.g., Jira), to query data for processing and generate reports.
- **Policy Center**: This is the interface for encapsulating a group of non-functional requirements, such as software security objectives, coding standards compliance, reliability goals, etc., into policies. Practices for automatically monitoring policies, such as static analysis, unit testing, etc., are also configured in Policy Center. Additionally, Policy Center enables you to set quality gates, which measure compliance with one or more policies at specific points in the SDLC.

In this section:

- [Getting Started with DTP Enterprise Pack](#)
- [Extension Designer](#)
- [Policy Center](#)
- [Extensions for DTP Enterprise Pack](#)

What's New in DTP Enterprise Pack 5.3.2

SDLC Analytics with Extension Designer

Extension Designer provides an interface for creating and managing standalone snippets of data processing logic, called “workflows” or “slices”, that provide visibility into potential problem areas within the code, as well as potential problematic processes.

Extension Designer 5.3.2 leverages the latest Development Testing Platform APIs, as well as APIs from third-party systems (e.g., JIRA), to query data for processing and generate reports.

This release also features a consolidated interface for Enterprise Pack extensions (PIE Slices, DTP Workflows, and Policy Center practices) and simpler integration with Policy Center and DTP.

See [Extension Designer](#) for additional information.

New and Updated Extensions

This release includes updates to the following artifacts:

PIE Slices

- **Change Based Testing**: Enables organizations to efficiently reduce risk by focusing on the tests that need to be re-executed due to changes in the code base, rather than the whole test suite.
- **Risky Code Changes**: When code changes it represents risk, this slice analyzes the changes in the code by looking at how maintainable the code is versus how well it is tested. The slice then assigns a risk level and shows the end-user how much “quality debt” exists.
- **Modified Coverage**: Getting 100% coverage on the entire code base is impossible when you have a large legacy code base. But it is the new or modified code that represents the risk. By focusing on the changing code, this slice enables you to efficiently eliminate the risk associated with change.
- **Test Stability Index**: Test stability is an ongoing issue for many organizations. Developers and tester know that unstable tests are going to fail from time to time, so they are ignored. But when there are thousands of tests, determining a failure associated with a legitimate problem is difficult. This slice helps remove the noise by scoring each test to categorize it’s stability. This allows teams to focus in two ways:
 1. Fix the “real failures” now, because this usually indicates you have an immediate problem/regression.
 2. Quantify which tests need to be stabilized.
- **Key Performance Indicator**: The volume of static analysis violations is not always the best indicator of team performance. The KPI slice enables you to define multiple custom profiles that weight static analysis rules differently and calculate an overall score for that KPI.

DTP Workflows

- **Traceability Report**: Provides a customizable Test Traceability widget and Requirement Traceability widget. Each widget drills down into interactive reports that help you understand how well requirements are being implemented and tested.
- **Static Analysis Violation Reporter for Atlassian JIRA**: Provides an example of how to use the Atlassian JIRA importer to create and manage JIRA bugs and/or tasks for problems (e.g., static analysis violations or unit test failures) found during test and analysis.
- **Test Failures by Build**: Creates a widget that lists DTP build IDs with dynamic analysis test failures.
- **DTP Build Review Workflow**: Provides an example of how to automate the creation of Change Reviews (see [Change Explorer](#) for more details), as well as specify a user who will receive automated email notifications about the reviews created.

- **Export DTP Explorer Data to CSV:** This workflow accepts any parameter URL from DTP's [Test Explorer](#) and [Violations Explorer](#) and generate a CSV output.
- **DTP File Based Licensing Report:** If you are using a file-based DTP license, this workflow provides detailed information about your Parasoft DTP license consumption/usage.

Policy Center Practices

- **Code Coverage:** Defines expectations around code coverage and ensures that an acceptable level of coverage is achieved.
- **Defects for JIRA:** Defines expectations around the number of Bugs present in a JIRA project to ensure that large numbers of bugs are not left unattended.
- **Metrics:** Allows you to monitor various code metric values by defining thresholds and aggregators.
- **Regression Test:** Ensures that the number of acceptable regression test failures is within the specified threshold for the project.
- **Static Analysis:** Ensures that static analysis violations that are considered high severity are remediated within an acceptable period of time.

See [Extensions for DTP Enterprise Pack](#) for additional information.

Additional Updates for Enterprise Pack

- Node Changes: See [Migrating PIE Slices to Extension Designer](#) for information about the Extension Designer nodes changed in this update.
- Creating Custom Widgets for DTP: The ready-to-deploy custom widgets shipped with PIE Slice Designer 1.3.x have been replaced in Extension Designer 5.3.2 with building blocks for creating your own custom widgets. See [Creating Custom DTP Widgets Using Extension Designer](#) for additional information.
- Policy Center Practice deployment is integrated with Extension Designer, providing a single framework to manage all DTP extensions (Policy Center Practices, DTP Workflows, and PIE Slices).
- Consolidated configuration UI for Policy Center and Extension Designer--no need to edit separate properties file to setup Policy Center
- Unified startup script for the embedded database, Policy Center, and Extension Designer.
- Extension Designer is now integrated into the common DTP UI framework, enabling easy access to the application from the DTP settings menu.
- Extension Designer now allows you to manage multiple instances of "services" (see [Important Concepts and Terminology](#)).
- New Event Broker to manage more robust events with multiple service instances.
- Enhanced Artifact Manager (see [Downloading and Installing Artifacts](#)). Installing an artifact automatically runs the `install-addon` command when DTP is installed on the same machine as the same user.
- New Endpoint, Components, Parameters node allow flows to provide custom widget and reports directly to DTP without copying files to DTP.
- Upgraded Infrastructure:
 - Updated Node.js to 6.10.0
 - Updated MongoDB to 3.4.2
 - Updated Node-Red to 0.16.2