DTP Components

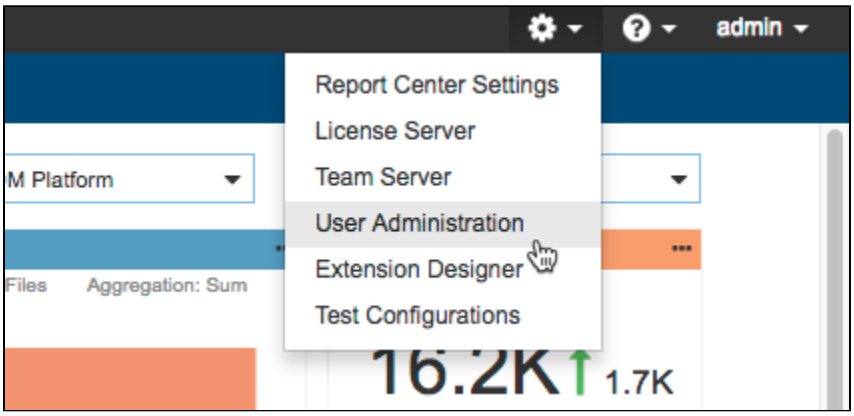
In this section:

What's in DTP

If you have a license for complete Parasoft DTP functionality, then your installation will include the components listed in the following table (also see Upgrading DTP from Standard to Enterprise Edition). Contact your Parasoft representative if you are interested in upgrading your license.

Component	Description
DTP interface	The URL is used to connect to the Tomcat report server running on port 80 (for Windows) and port 8080 (for Linux): http://[hostname-of-server]:80 (for Windows) http://[hostname-of-server]:8080 (for Linux) The interface provides access to Report Center and other applications in the DTP infrastructure.
Report Center	Provides end-to-end SDLC process visibility and control.
Extension Designer	(Additional license required) Interface for creating data processing flows that can automatically create specific views of your SDLC data, trigger external workflows, monitor policy compliance, etc. See Extension Designer.
License Server	Allows administrators to add and manage licenses
Team Server	Enables centralized administration and sharing of team artifacts.
User Administration	Allows administrators to grant and manage user permissions.

If you have administration permissions, you can access the servers and administration pages from the Administration menu:



Additional Components

This documentation is focused on Report Center and other server-based components of DTP, but Parasoft also provides code analysis and test execution components for the DTP ecosystem, as well as desktop plug-ins that communicate with DTP.

Parasoft Code Analysis and Testing Tools

Parasoft tools, such as C/C++test, Jtest, and dotTEST, drive SDLC analytics. They analyze code, execute tests, measure coverage, and perform other quality tasks. Extensions for using open source analyzers and tools are available. Contact Parasoft to download extensions.

You can run the tools on the desktop or automate code analysis and test execution as part of the build process. The Parasoft tools and third-party analyzers generate local HTML/XML files and publish details to DTP for aggregation, reporting, and analysis.

DTP IDE Plugins

Parasoft provides plugins for popular IDEs, such as Visual Studio, Eclipse, NetBeans, and IntelliJ. The plugins enable you to integrate Parasoft tools and analyzers into the IDE for local GUI-based code analysis. Additionally, the plugins enable you to retrieve findings that have been processed by DTP and import them into the IDE. Parasoft tools do not need to be plugged into the IDE for you to leverage the ability to download and import DTP findings. The IDE plugins can also be configured so that only findings with specific prioritization/metadata associated with them are downloaded and imported into the IDE.

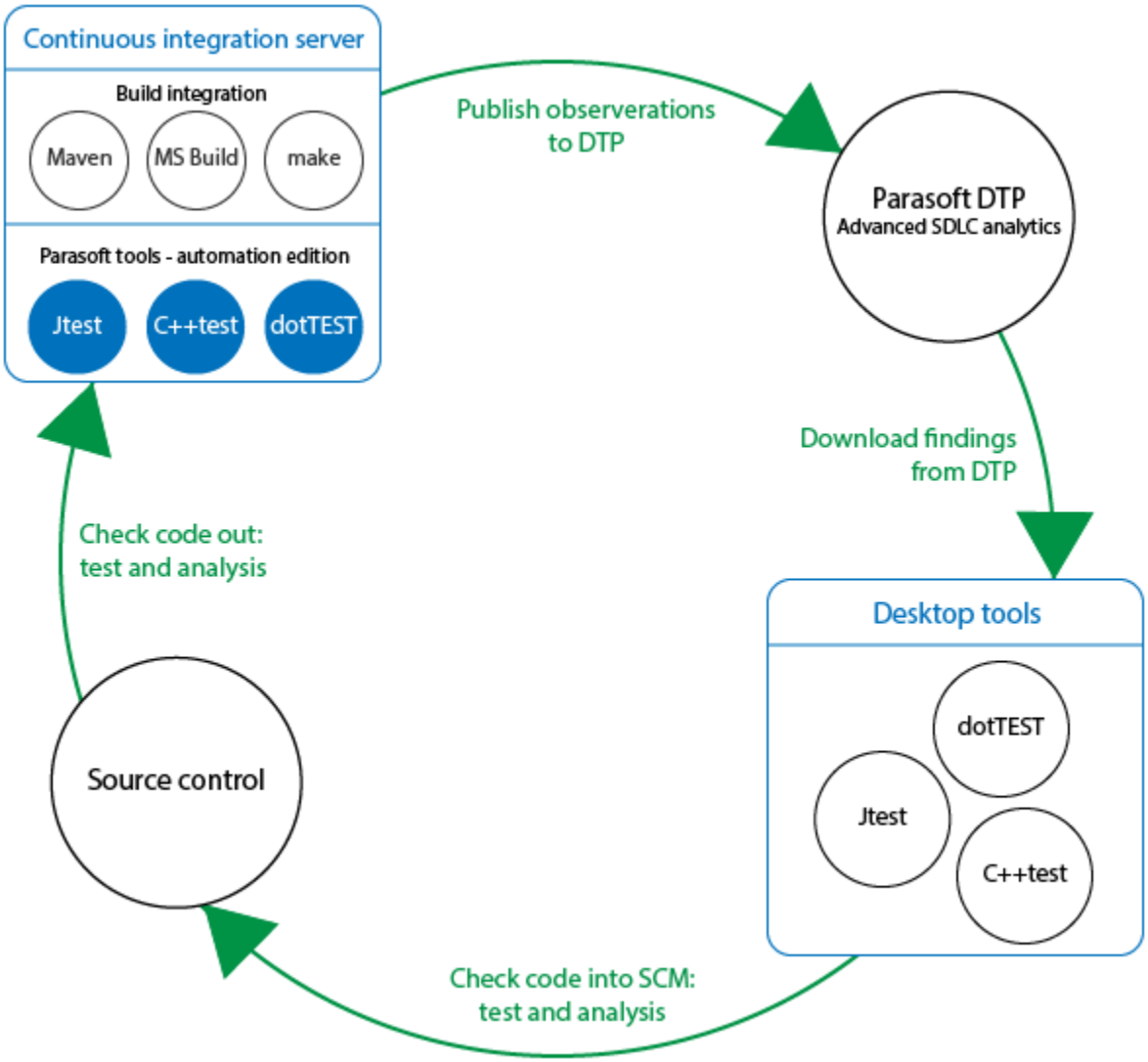
DTP Enterprise Pack

DTP Enterprise Pack is a suite of tools for deriving deeper SDLC analytics and for using the complex data to automate compliance with development policies. Enterprise Pack includes the following applications:

- **Extension Designer**: Create custom logic flows that perform additional data processing functions and calculations. You can use Extension Designer to create functions that access patterns buried deep in the SDLC data. See Extension Designer for details.
- **Extensions for Enterprise Pack**: You can extend DTP functionality with readymade extensions from Parasoft. Extensions perform a range of tasks, such as applying a secondary analysis of the test, coverage, metrics, and other analysis data sent to DTP, processing SDLC analytics against development policies and returning prioritized, actionable tasks. See Extensions for DTP Enterprise Pack.

The DTP Workflow

You can integrate DTP into your own development processes, but the following workflow describes the typical implementation.



Integrating Parasoft Tools with the Build

Parasoft tools ship with plugins for integration with your build tools (i.e., Maven, Ant, Gradle, MS Build, make, etc.). These integrations allow you to analyze code and send data to DTP automatically as part of the automated build processes and continuous integration (CI).

The tools also ship with plugins/integrations for the CI infrastructure (i.e., Jenkins, TeamCity). These integrations are also available from Parasoft.

Capturing Observations

When Parasoft tools execute analysis, they capture massive amounts of detailed data associated with the code called “observations.” Observations can be code quality data, such as static analysis violations, unit test failures, etc., as well as logistical information about the code, such as authorship, scope, and source control location.

Converting Data into Findings

When observations are sent to DTP, they are converted into “findings” and stored in the database. Findings are observations that have been analyzed, normalized, and aggregated into actionable data.

If the tools are configured to include source code information, then DTP will retrieve the source code and present to the user when viewing the normalized findings in the browser-based reporting interface (e.g., Prioritization View, Tests Explorer, Metrics Explorer).

The tools can also send copies of analyzed source code to DTP for display if DTP is unable to access the code from the source control system. Reasons for enabling access to source files using this approach include:

- security or networking constraints
- the code is ‘generated’ and not stored in source control

Working with Findings and Applying Additional Metadata

The reporting interface enables you to review, navigate, and filter findings. You can also set additional metadata, such as:

- Assign static analysis and flow analysis violations to team members for remediation
- Set due dates for remediation
- Set references to external systems, such as a defect tracking system
- Change the priority level of findings
- Set Risk/Impact categorizations

You can also leverage the REST API of DTP to extract details about findings for integration with external systems and apply analysis-flows (referred to as "slices") configured Extension Designer. Slices can be triggered on demand or through the use of event-based triggers. Examples of slice applications include:

- Generating derived data, such as risk in the application, using data available within DTP and other systems of record
- Triggering workflows in external systems, such as creating work-items or defects (e.g., in JIRA)
- Automatically applying metadata based on defined heuristics of your policy

Importing DTP Findings to the Developer Desktop

After findings are processed in DTP, developers can import them directly into their IDEs for remediation using a DTP IDE Plugin. Findings should be prioritized and filtered so that only tasks relevant to the developer are imported. If the developers also have Parasoft tools installed and licensed, they can address the findings and reanalyze the code locally before committing to source control.

Continuing the Cycle

When the developers check their code back into source control, the continuous integration process picks up the change, and the workflow is repeated. This ensures that defects are detected and prevented from becoming software bugs later in the development process when the costs of remediation are much higher. As a result, Parasoft DTP facilitates continuous testing, enabling you to accelerate the SDLC while ensuring the safety, security, and reliability of your applications.

About the DTP Structure

DTP and Data Collector connect to the database (MySQL or Oracle) as soon as they are launched. The database starts up first and shuts down last. No other applications should use the database engine directly.

Code analyzers and other tools that send data to DTP connect to Data Collector at port 32323 on the server. Users interface with the system and view the data by using a Web browser to connect to the Tomcat report server running on port 80 (Windows) or 8080 (Linux).

The URL for DTP is either `http://[hostname-of-server]:80` for Windows or `http://[hostname-of-server]:8080 for Linux.`

In rare instances, you may need to change the default port for either the DTP web server, Data Collector, or the JMS broker. See Reconfiguring DTP Ports for more information: