MSBuild Tasks Library

Overview

Allows you to call dynaTrace automation tasks from MSBuild

Name	**MSBuild Tasks Library**
Version	1.0.0
dynaTrace Versions	5.x, 6.x
Author	dynaTrace software
License	dynaTrace BSD
Support	Supported
Download	• MSBuild Tasks for Dynatrace AppMon 6.5 • MSBuild Tasks for dynaTrace 6.3 • MSBuild Tasks for dynaTrace 6.2 • MSBuild Tasks for dynaTrace 6.1 • MSBuild Tasks for dynaTrace 6.0 • MSBuild Tasks for dynaTrace 5.6

Description

The Download Package includes an .NET Library that implements several tasks to be used by MSBuild scripts.

Following tasks can be executed:

- Start, Stop or Restart the dynaTrace Server
- Start and Stop Session Recording
- Change Active Configuration
- Set Test Information

This library makes use of the .NET Command Library.

Installation

1. Save the attached file locally to the computer where the dynaTrace AppMon Client is installed.
2. Unzip the file.
3. Copy the two extracted dlls to a directory where it can be picked up by MSBuild when executing your build scripts
4. View the extracted msbuild_sample.xml file. It shows how you can register the new build tasks in your MSBuild scripts

Usage

You have to register the new Build Tasks with the UseTask configuration entry in your build scripts like this:

```
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.StartSessionRecordingTask" />
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.StopSessionRecordingTask" />
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.SetActiveConfigurationTask" />
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.ServerServiceTask" />
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.ConfigureApplicationTask" />
<!-- for dynaTrace 3.5 - 6.0 -->
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.SetTestInformation" />
<!-- for dynaTrace 6.1 and above -->
<UsingTask AssemblyFile="dynaTrace.MSBuild.Task.dll"
TaskName="dynaTrace.MSBuild.Task.StartTestTask" />
```

Each task can now be used in your targets. Following properties are available for the different tasks:

StartSessionRecordingTask

- ProfileName: (mandatoy) - specifies the System Profile you want to start recording on
- RecordSessionName: (optional) - allows you to define the name of the recorded session
- dtVersion: (optional) - this is the dynaTrace Client Version that you want to use. In case multiple versions of dynaTrace are installed you use this to connect to the correct version
- dtInstallDir: (optional) - this is the dynaTrace Client Installation directory. If not specified the value is read from the registry
- ServerName: (optional) - the dynaTrace Server Hostname to connect to. Default: localhost
- Username: (optional) - the username to connect to the dynaTrace Server. Default: admin
- Password (optional) - the password to connect to the dynaTrace Server. Default: admin

StopSessionRecordingTask

- ProfileName: (mandatory)
- dtVersion, dtInstallDir, ServerName, Username, Password: see description of StartSessionRecording

SetActiveConfigurationTask

- ConfigurationName: (mandatory) - configuration name to switch to
- ProfileName: (mandatory) - System Profile where you want to switch the active configuration
- dtVersion, dtInstallDir, ServerName, Username, Password: see description of StartSessionRecording

ServerServiceTask

- ServiceCommand: (mandatory). Allows one of the following values: startup, restart, shutdown
- dtVersion, dtInstallDir, ServerName, Username, Password: see description of StartSessionRecording

ConfigureApplicationTask

- Executable: (mandatory). Defines the full path to the executable you want to configure, e.g.: C:\MySample\MyApp.exe
- Active: (mandatory). True or False. Allows to switch a configuration active or inactive
- AgentName: (optional). Name of the Agent. Default: name of the executable file, e.g.: MyApp.exe
- CommandLine: (optional). Command Line that must match the process. Default: empty==any command line value
- ServerName: (optional). dynaTrace Servername to let the Agent connect to. Default: localhost
- ServerPort: (optional). dynaTrace server port to let the Agent connect to. Default: 9998
- LogFile: (optional). location where to store Agent logfile
- LogLevelLife: (optional). LogLevel for the Log File. Possible Values: LOG_NONE, LOG_SEVERE, LOG_WARNING, LOG_INFO, LOG_FINE, LOG_FINER, LOG_FINEST. Default: LOG_NONE
- LogLevelConsole: (optional). LogLevel for Console Logging. Default: LOG_INFO

SetTestInformation (from dynaTrace 3.5 till dynaTrace 6.0)

- VersionMajor: Major Version of your Software
- VersionMinor: Minor Version of your Software
- VersionRevision: Revision of your Software
- VersionBuild: Build Number of your Software
- VersionMilestone: Milestone of your Software
- Marker: Marker Name that should be shown in the chart for this test execution
- Agent Group: The name of the agent group that this test information should be set for. Applied to all agent groups if not set.

- SystemProfile: The name of the system profile that this test information should be set for.
- CustomProperty: A custom property that should be set for this test information

StartTest (from dynaTrace 6.1)

- VersionMajor: Major Version of your Software
- VersionMinor: Minor Version of your Software
- VersionRevision: Revision of your Software
- VersionBuild: Build Number of your Software
- VersionMilestone: Milestone of your Software
- Marker: Marker Name that should be shown in the chart for this test execution
- SystemProfile: The name of the system profile that this test information should be set for.
- Category: The name of the test category, available choices: 'unit', 'performance', 'uidriven', 'load', 'webapi'
- LoadTestName: The name for the load test, only used when Category is set to 'load'

StartTest task registers a new test run in dynaTrace and sets the DT_TESTRUN_ID environment variable picked up by the .NET agent. Tests should be executed after this is made so that the agent sees the environment variable and will use it when reporting data back to dynaTrace.

Internals

The Library uses the dynaTrace.Command.dll which is installed with the dynaTrace Client. dynaTrace.Command.dll uses the dynaTrace Web Service interfaces. For every dynaTrace.Command.dll call the library creates an EventLog Entry with detailed information about the command line arguments that were passed. This should help with troubleshooting.

The task ConfigureApplicationTask needs write access to the HKLM Registry on your local machine. In case your user does not have the appropriate access rights you will run into errors.

Sample MSBuild Script

The download package includes a sample msbuild file. Check it out for a reference

An additional sample can be seen here: VSTS Sample Test Project

Attachments

File	Modified ▲

No files shared here yet.