List of contents

[TFSEventWorkFlows Documentation 2](#_Toc317606617)

[1. Installation of the plugin on the Team Foundation Server 2](#_Toc317606618)

[2. Configuration of the TFSEventWorkflowServerPlugin 2](#_Toc317606619)

[3. Description of the workitem activities in the TFSActivityLib 3](#_Toc317606620)

[3.1. GetWorkItemChangedEventData 3](#_Toc317606621)

[3.2. GetWorkItem 3](#_Toc317606622)

[3.3. SaveWorkItem 3](#_Toc317606623)

[3.4. ReadWorkItemField 3](#_Toc317606624)

[3.5. GetValidationOfWorkItem 3](#_Toc317606625)

[3.6. GetParentWorkItem 5](#_Toc317606626)

[3.7. GetChildWorkItems 5](#_Toc317606627)

[3.8. CreateNewWorkItem 5](#_Toc317606628)

[3.9. CreateLink 5](#_Toc317606629)

[3.10. ChangeWorkItemField 5](#_Toc317606630)

[3.11. GetLastWorkItemHistoryComment 7](#_Toc317606631)

[4. Description of the workspace activities in the TFSActivityLib 7](#_Toc317606632)

[4.1. CopyDirectory 7](#_Toc317606633)

[4.2. CopyDirectory 7](#_Toc317606634)

[4.3. CreateWorkspace 7](#_Toc317606635)

[4.4. DeleteWorkspace 8](#_Toc317606636)

[4.5. SetWorkspaceMapping 8](#_Toc317606637)

[4.6. GetFromSourceControl 8](#_Toc317606638)

[4.7. AddToSourceControl 8](#_Toc317606639)

[5. How to create a custom workflow with the TFSActivitysLib in Visual Studio 2010 9](#_Toc317606640)

[5.1. How to add TFSActivitysLib to the workflow designer 9](#_Toc317606641)

[5.2. InArguments for a workflow 11](#_Toc317606642)

[7. Sample Configuration: 13](#_Toc317606643)

# TFS Event Workflows Documentation

The TFS Event Workflows consists of the following components:

* **TFSEventWorkflowServerPlugin** - The Sever Plugin subscribes to the Team Foundation Sever events and executes the proper workflow.
* **TFSActivityLib** – Collection of activities to build a custom workflow.
* **LoggingLib** – Responsible for writing errors to a log file using Log4Net.

## Installation of the plugin on the Team Foundation Server

Build the TFS Event Workflow project and copy the assemblies from the output to the plugins folder of the TFS.

Path to plugins folder:

**…\Microsoft Team Foundation Server 2010\Application Tier\Web Services\bin\Plugins**

Files to copy:

* artiso.TFSEventWorkflows.LoggingLib.dll
* artiso.TFSEventWorkflows.TFSActivitiesLib.dll
* artiso.TFSEventWorkflows.TFSEventWorkflowsServerPlugin.dll
* artiso.TFSEventWorkflows.TFSEventWorkflowsServerPlugin.dll.config
* log4net.dll

**TFS Event Workflows Setup:**

The TFS Event Workflow can also be installed by executing the "TFSEventWorkflowsSetup.msi" located in the TFSEventWorkflowsSetup project. Set the installation path to the plugins folder of the TFS.

(…\Microsoft Team Foundation Server 2010\Application Tier\Web Services\bin\Plugins).

## Configuration of the TFSEventWorkflowServerPlugin

To subscribe to a Team Foundation Server event it must be configured in the configuration file (artiso.TFSEventWorkflows.TFSEventWorkflowsServerPlugin.dll.config).

In the tfsEventConfig section the following values must be set:

**name:**

A name for the entry. Must be unique.

**fullTypeName:**

The full name of the TFS event.

**eventAssemblyName:**

The full name of the assembly which contains the TFS event.

**workflowFileName:**

The path of the workflow file which will be executed with the event. If the workflow file is in the same directory as the server plugin dll the workflow file name is sufficient (e.g. MyWorkflow.xaml).

Example configuration which subscribes to the CheckinNotification and WorkItemChangedEvent:

<tfsEventConfig>

<tfsEvents>

<add name="workItemChangedEvent1"

fullTypeName="Microsoft.TeamFoundation.WorkItemTracking.Server.WorkItemChangedEvent"eventAssemblyName="Microsoft.TeamFoundation.WorkItemTracking.Server.Dataaccesslayer"workflowFileName="AddTaskToNewUserStory.xaml" />

</tfsEvents>

</tfsEventConfig>

## Description of the workitem activities in the TFSActivityLib

### GetWorkItemChangedEventData

**Purpose:**

Gets the data of the WorkItemChangedEvent and returns them to the workflow.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItemChangedEvent | WorkItemChangedEvent | in |  |
| TeamFoundationRequestContext | TeamFoundationRequestContext | in |  |
| WorkItem | WorkItem | out | The saved workitem. |
| TFSCollectionUrl | string | out | Url of the TFS collection. |

### GetWorkItem

**Purpose:**

Gets a workitem by the workitem id.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| TFSCollectionUrl | string | in | Url of the TFS. |
| WorkItemID | int | in | ID of the workitem. |
| WorkItem | WorkItem | out | The wanted Workitem. |

### SaveWorkItem

**Purpose:**

Saves the given workitem.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in | The Workitem to save. |
|  |  |  |  |

### ReadWorkItemField

**Purpose:**

Reads the given workitem field and returns the value.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in |  |
| FieldReferenceName | string | in | The reference name of the field. |
| Value | object | out | The value of the Field. |

### GetValidationOfWorkItem

**Purpose:**

Checks if the given workitem is valid and returns a list of invalid fields.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in | The workitem to validate. |
| CollectionOfNotValidFields | List<Fields> | out | List of invalid fields. |

### GetParentWorkItem

**Purpose:**

Gets the parent workitem of the given workitem.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in | The child workitem. |
| ParentWorkItem | WorkItem | out | The parent workitem. |

### GetChildWorkItems

**Purpose:**

Gets all child workitems of the given parent workitem.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in | The parent workitem. |
| ChildWorkItems | List<WorkItem> | out | List of all child workitems |

### CreateNewWorkItem

**Purpose:**

Creates a new workitem.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| TeamProjectName | string | in | The team project name. |
| TFSCollectionUrl | string | in | Url of the TFS collection. |
| WorkItemType | string | in | The type of the new workitem. |
| NewWorkItem | WorkItem | out | The new workitem. |

### CreateLink

**Purpose:**

Creates a link between two workitems.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| ParentWorkItem | WorkItem | in | The parent workitem. |
| ChildWorkItem | WorkItem | in | The child workitem. |
| LinkType | string | in | The type of link between the two workitems. |

### ChangeWorkItemField

**Purpose:**

Changes the value of a workitem field.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in |  |
| FieldReferenceName | string | in | The reference name of the workitem field. |
| NewValue | object | in | The new value of the field. |

### GetLastWorkItemHistoryComment

**Purpose:**

Gets the last history comment of a workitem.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| WorkItem | WorkItem | in |  |
| HistoryComment | string | out | The history comment. |

## Description of the workspace activities in the TFSActivityLib

### CopyDirectory

**Purpose:**

Copies a directory and all of its content to another directory.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| SourceFolder | string | in | Path of the folder to copy. |
| TargetFolder | string | in | Path of the target folder. |
| OverrideExisitingFiles | bool | in | If true overrides all existing files in the target folder. |

### CopyDirectory

**Purpose:**

Copies a file to another path.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| SourcePath | string | in | Path of the file to copy. |
| TargetPath | string | in | Path of the target destination. |

### CreateWorkspace

**Purpose:**

Creates a workspace and maps a local folder to source control folder.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| TFSCollectionUrl | string | in | Url of the TFS collection. |
| ServerPath | string | in | Path to the mapping folder in the source control. |
| LocalPath | string | in | Path to local mapping folder. |
| WorkspaceName | string | in | Name of the new workspace. |
| WorkspaceComment | string | in | Comment of the new workspace. |
| Workspace | Workspace | out | The new workspace. |

### DeleteWorkspace

**Purpose:**

Deletes a workspace.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| Workspace | Workspace | in | The workspace to delete. |

### SetWorkspaceMapping

**Purpose:**

Sets a new mapping for a workspace.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| Workspace | Workspace | in | The workspace to map. |
| ServerPath | string | in | Path to the mapping folder in the source control. |
| LocalPath | string | in | Path to local mapping folder. |

### GetFromSourceControl

**Purpose:**

Loads files from source control to the local folder.

**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| Workspace | Workspace | in |  |
| ServerPath | string | in | Path to the mapping folder in the source control. |

### AddToSourceControl

**Purpose:**

Performs a "Checks out for edit" on all files in the local workspace. Then copies all files from the upload directory in the local workspace and performs a "Checkin pending changes".

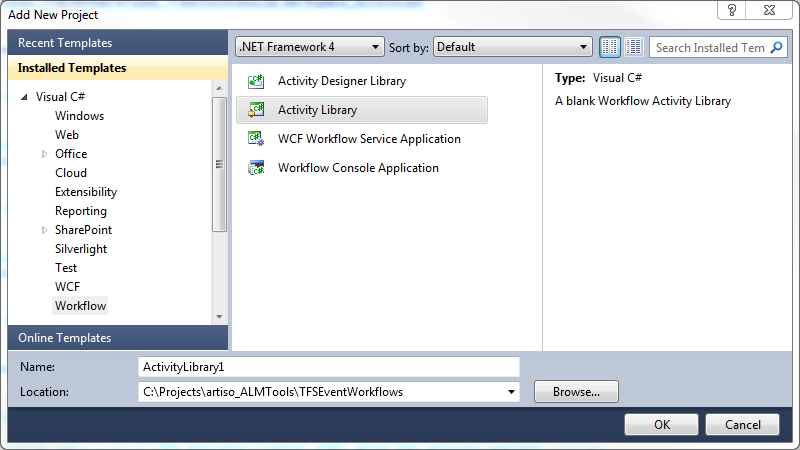
**Arguments:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Direction | Description |
| Workspace | Workspace | in |  |
| DirectoryToUpload | string | in | Path to the directory which contains the files to upload. |
| CheckInComment | string | in | Comment of the checkin. |

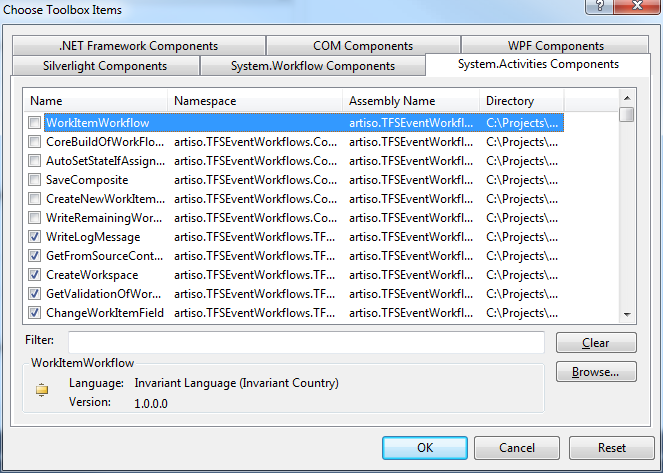
## How to create a custom workflow with the TFSActivitysLib in Visual Studio 2010

### How to add TFSActivitysLib to the workflow designer

1. Create a new solution. Add a new "Activity Library" project from the "Workflow" template.



1. To add a reference to the TFSActivitysLib (artiso.TFSEventWorkflows.TFSActivitiesLib.dll).
2. Open the workflow XAML file.
3. Add the TFSActivitysLib to the Toolbox by right-click on the Toolbox and select "Choose Items…"

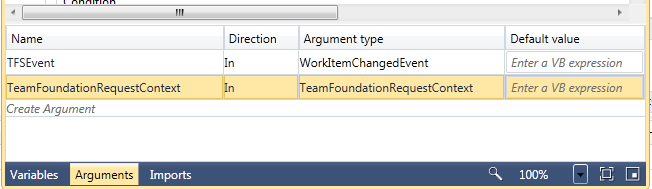


1. In the "Choose Toolbox Items" window click "Browse…" and select the artiso.TFSEventWorkflows.TFSActivitiesLib.dll
2. The activities from the TFSActivitysLib are in the Toolbox and can be dragged in the workflow designer.

### InArguments for a workflow

To invoke a new workflow with the TFSEventWorkflowServerPlugin the workflow must accept two InArguments:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Direction | Description | |
| TFSEvent | depends on the TFS Event | in | | The event which triggered the workflow (e.g. WorkItemChangedEvent) | |
| TeamFoundationRequestContext | TeamFoundationRequestContext | in | |  | |



Arguments for the workflow

## Sample Configuration:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

  <configSections>

    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,log4net"/>

    <section

      name="tfsEventConfig" type="artiso.TFSEventWorkflows.TFSEventWorkflowsServerPlugin.TFSEventConfig, artiso.TFSEventWorkflows.TFSEventWorkflowsServerPlugin"/>

  </configSections>

  <tfsEventConfig>

    <tfsEvents>

      <add

name="checkInEvent"

fullTypeName="Microsoft.TeamFoundation.VersionControl.Server.CheckinNotification"

eventAssemblyName="Microsoft.TeamFoundation.VersionControl.Server"

workflowFileName="CreateNewWorkItemAfterCheckIn.xaml" />

      <add

name="workItemChangedEvent1"

fullTypeName="Microsoft.TeamFoundation.WorkItemTracking.Server.WorkItemChangedEvent"

eventAssemblyName="Microsoft.TeamFoundation.WorkItemTracking.Server.Dataaccesslayer"

workflowFileName="AddTaskToNewUserStory.xaml" />

      <add

name="workItemChangedEvent2"

fullTypeName="Microsoft.TeamFoundation.WorkItemTracking.Server.WorkItemChangedEvent"

eventAssemblyName="Microsoft.TeamFoundation.WorkItemTracking.Server.Dataaccesslayer"

workflowFileName="EmptyWorkflow.xaml" />

    </tfsEvents>

  </tfsEventConfig>

  <log4net>

    <appender name="OfficeFile" type="log4net.Appender.FileAppender">

      <!--Path of the log file. The file name must contain a wildcard for the date patern: %date{yyyy-MM-dd}-->

      <file type="log4net.Util.PatternString" value="C:\Temp\TFSEventWorkflow\_Log\_%date{yyyy-MM-dd}.log"/>

      <layout type="log4net.Layout.PatternLayout">

        <param name="ConversionPattern" value="%date{yyyy-MM-dd HH:mm:ss.fff} %-5level [%3.3thread] %-50.50logger{2}  | %message %exception %newline"/>

      </layout>

    </appender>

    <root>

      <!--Logging level:

      ALL: Shows all messages

      ERROR: Only critical erros in the workflow-->

      <level value="ALL"/>

      <appender-ref ref="OfficeFile"/>

    </root>

  </log4net>

  <appSettings>

  </appSettings>

</configuration>