



Configuration Guide

Version 1.5

Table of contents

1. Overview	3
1.1 Purpose.....	3
1.2 Preconditions	3
1.3 Applying changes	3
2. Step-by-Step Configuration Example	5
2.1 Role Hierarchy Configuration and User Permissions	5
2.1.1 Default Role Hierarchy	5
2.1.2 Role Hierarchy Configuration Example	5
2.2 Adding Supplementary Work Item Fields	8
2.3 Configuring Work Item (WI) States.....	11
2.4 Adding Virtual Projects	11
2.5 Configuring Supplementary Work Item Fields for Data Export	12
3. Configuration Parameters Reference	14
3.1 Abbreviations	14
3.2 Configure User Permissions.....	14
3.3 Configure Work Items Types.....	15
3.4 Configure Virtual Projects.....	16
4. Known Limitations	18
4.1 Collection level user groups are not supported	18
4.2 Underscore character in role names	18

1. Overview

1.1 Purpose

This document describes the manual configuration steps required to make TX Chrono properly operate in a particular environment that does not match the settings TX Chrono is installed by default with.

Chapter 2 is an example of a step-by-step configuration

Chapter 3 is a complete reference on the XML parameters

Chapter 4 is a list of known limitations

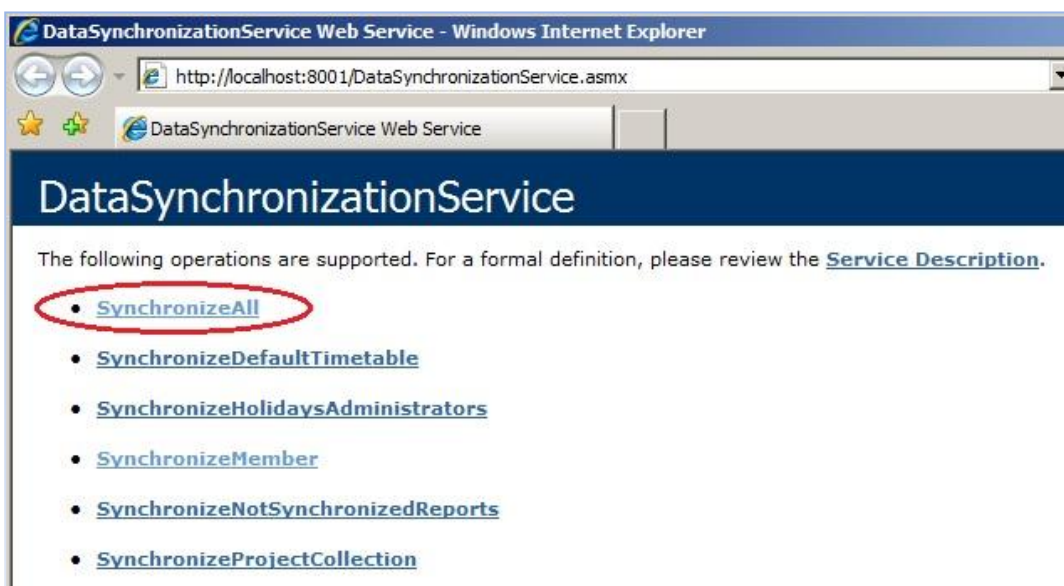
1.2 Preconditions

All the TX Chrono components have been successfully installed as described in the **TX Chrono Installation Guide**.

1.3 Applying changes

To apply the changes after the TX Chrono configuration files have been modified, one needs to execute the **SynchronizeAll** command as shown below.

Open the IIS manager and select Sites**TXChronoTimeReportingService**, then right click **DataSynchronizationService.asmx** and execute the **Browse** context command. On the page that appears click **SynchronizeAll**.



Picture 1. The SynchronizeAll Command

Click the **Invoke** button. If the operation succeeds, it should return **true** as shown bellow

```
<?xml version="1.0" encoding="utf-8" ?>  
<boolean xmlns="http://tempuri.org/">true</boolean>
```

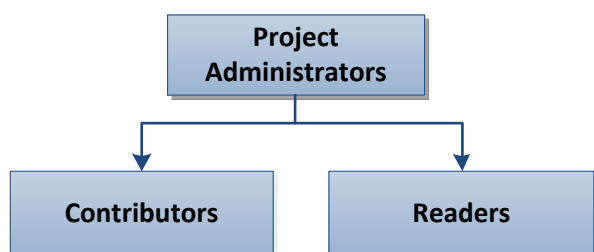
2. Step-by-Step Configuration Example

2.1 Role Hierarchy Configuration and User Permissions

User rights are determined by the roles hierarchy. This hierarchy is set up in the configuration file of TX Chrono Reporting Service. This hierarchy contains Project and Server roles from Team Foundation Server.

2.1.1 Default Role Hierarchy

By default, this role hierarchy contains the Project Administrators role at the top. The Readers role and Contributors role are configured to be child roles of the Project Administrators role. Therefore, the Project Administrators role is a parent role for the Readers and Contributors roles.



Picture 2. The Default Role Hierarchy

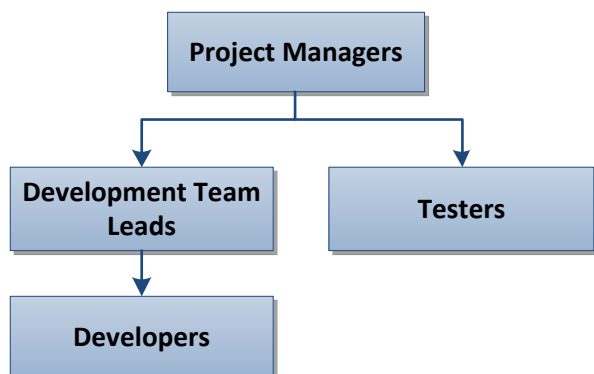
Users, who have at least one parent role, have access to the **View Timesheets** page. These users can view time reports of the users of the child roles.

By default, Project Administrators can view and approve or decline timesheets of their subordinates.

2.1.2 Role Hierarchy Configuration Example

The role hierarchy can be configured right after the installation of TX Chrono Reporting Service. After the changes are done, one has to wait for about an hour so that TX Scheduler would synchronize the TX Chrono Database with the Team Foundation Server data.

The sample role hierarchy is shown below. It is similar to the one you may have on your project.



Picture 3. The Example Role Hierarchy

Each role corresponds to a project group in TFS. The role hierarchy is stored in the web.config file of TX Chrono Reporting Service. To create your own role hierarchy, take the following steps:

1. Locate the folder with the installed TX Chrono Reporting Service and open the **web.config** file with any text editor (Notepad, for example);
2. Find the **<confirmationSection>** section. By default, the role hierarchy displayed in image 1 will be implemented as follows:

```

<role name="Project Administrators" hasProjectRoles="true">
  <juniorRole name="Contributors" hasProjectRoles="true"/>
  <juniorRole name="Readers" hasProjectRoles="true"/>
</role>
  
```

In order to implement the hierarchy displayed in the image below, you should replace it with the following XML:

```

<role name="Project Managers" hasProjectRoles="true">
  <juniorRole name="Development Team Leads" hasProjectRoles="true"/>
  <juniorRole name="Testers" hasProjectRoles="true"/>
</role>
<role name="Development Team Leads" hasProjectRoles="true">
  <juniorRole name="Developers" hasProjectRoles="true"/>
</role>
  
```

Below is the description of the elements of the role tree.

Element	Attribute	Description
role		Describes a parent role. Can contain juniorRole elements as a child node.
	name	Contains the name of the role. This attribute is required.
	hasProjectRoles	Identifies the project role. If set to true, it means that the role marked with this attribute is a Project role, otherwise it is a Server role.
juniorRole		Describes a child role. Cannot contain child node.

The hierarchy has indirect dependence, i.e. a Project Manager controls (can view reports of) Development Team Leads; Development Team Leads control Developers, and therefore the Project Manager controls Developers, as well. Regarding the report confirmation, one can confirm reports of users with juniorRoles only: Project Managers confirm reports of Development Team Leads, Development Team Leads confirm reports of Developers.

Note: To use Server roles with TFS 2010, you should grant them the “View collection-level information” permissions to these roles in each project collection, which is used by TX Chrono.

3. Locate the **<appSettings>** element. Find the **add** element with the key attribute set to AdminGroupName and change it to contain the Project Managers role.

```
<add key="AdminGroupName" value="Project Managers" />
```

4. Find the **add** element with the key attribute set to IsAdminGroupNameProjectRole and change its values to True. This key specifies the type of the Project Managers role. True means that the specified role is a project role in TFS, false—server role.

```
<add key="IsAdminGroupNameProjectRole" value="true" />
```

5. Save the **Web.config** file and close it.
6. Go to the TX Chrono User Interface installation folder and open the **Web.config** file in any text editor (Notepad, for example).
7. Locate the **<appSettings>** element. Find the **add** element with the key attribute set to AdminRoleName and change it to contain the Project Managers role.

```
<add key="AdminRoleName" value="Project Managers" />
```

8. Locate the **<location path="Senior">** element. Find the **<allow>** element and change its value to the Project Managers and Development Team Leads roles:

```
<allow roles="Project Managers, Development Team Leads" />
```

Note: If you need several roles to be added, you should separate them by comma.

9. Save and close this **web.config**
10. In the same folder, find the **Web.Sitemap** file and open it in any text editor (Notepad, for example).
11. In this file, find the **<siteMapNode>** elements with the URL attribute starting with the “~/Senior/” string. Change the **roles** attribute value to the Project Managers role:

```
<siteMapNode url="~/Senior/TimeSheetView.aspx"
  title="View timesheets"
  description="View timesheets of other employees"
  roles="Project Managers, Development Team Leads"
/>
```

```
<siteMapNode url=~"/Senior/TimeSheetConfirmation.aspx"
             title="Confirm timesheets"
             description="Confirm timesheets of other employees"
             roles="Project Managers, Development Team Leads"
/>
```

Note: If you need several roles to be added, you should separate them by comma.

12. Close and save the **Web.sitemap** file.

2.2 Adding Supplementary Work Item Fields

TX Chrono is able to work with multiple work item fields, not just with the Completed Work field. Furthermore, it is possible to configure TX Chrono to display different fields for different work item types.

To configure work item types that should be displayed in TX Chrono, open the **web.config** file and change the value of the **WorkItemTypes** key in the **appSettings** section. The value should contain a list of work item types delimited by commas, for example 'Task,Bug'. By default, the only **Task** work item type is specified.

By default, TX Chrono displays the Completed Work field for work items with type Task. The person working with TX Chrono is able to set different values for each day of week. This is convenient, however, sometimes it is not sufficient. Below some other field types are described.

To configure your work item fields, open the **ApplicationSettings.xml** file. It is located in the **Configs** folder in the **TX ChronoTimeReportingService** directory.

There are two ways to configure work item fields for work item types:

1. Changing the **DefaultProjectSettings** section

The **ApplicationSettings.xml** file contains the **DefaultProjectSettings** section that allows configuring fields that will be displayed for work items from all the projects.

Below is an **example** of the **DefaultProjectSettings** section:

```
<DefaultProjectSettings>
  <LastModificationDate>0001-01-01T00:00:00</LastModificationDate>
  <WorkItemTypes>
    <WorkItemType>
      <Name>Task</Name>
      <Fields>
        <WorkItemTypeField>
          <Name>Completed Work</Name>
          <Alias>CW</Alias>
        </WorkItemTypeField>
      </Fields>
    </WorkItemType>
  </WorkItemTypes>
  <ReferenceName>Microsoft.VSTS.Scheduling.CompletedWork</ReferenceName>
  <FieldType>Editable</FieldType>
  <EditType>Daily</EditType>
  <IsDefaultField>true</IsDefaultField>
</DefaultProjectSettings>
```

```

    <WorkItemTypeField>
      <Name>Remaining Work</Name>
      <Alias>RW</Alias>
    <ReferenceName>Microsoft.VSTS.Scheduling.RemainingWork</ReferenceName>
      <FieldType>Editable</FieldType>
      <EditType>Weekly</EditType>
      <IsDefaultField>>false</IsDefaultField>
    </WorkItemTypeField>
    <WorkItemTypeField>
      <Name>Estimate</Name>
      <Alias>Est</Alias>
      <ReferenceName>Microsoft.VSTS.CMMI.Estimate</ReferenceName>
      <FieldType>ReadOnly</FieldType>
      <EditType>None</EditType>
      <IsDefaultField>>false</IsDefaultField>
    </WorkItemTypeField>
  </Fields>
</WorkItemType>
</WorkItemTypes>
</DefaultProjectSettings>

```

The **DefaultProjectSettings** section contains one or more **WorkItemType** sections. Each **WorkItemType** section consists of the name of the work item and the **Fields** section that has one or more **WorkItemTypeField** sections.

The **WorkItemTypeField** section has the following **obligatory fields**:

- **Name** – the name of a work item field (should be the same as in TFS)
- **Alias** – the abbreviation that will be displayed on the Project Report page instead of the field name. If alias is empty, the field name will be displayed. Aliases should be equal for fields with the same name.
- **ReferenceName** – the reference name of the work item field (should be equal to the reference name of the field in TFS).

See the following for the list of field reference names available in TFS by default:
<http://msdn.microsoft.com/en-us/library/ms194971.aspx>

- **FieldType** – there are two valid values for this field:
 - **Editable** – the field is available for editing;
 - **ReadOnly** – cannot be modified.
- **EditType** – defines how the field will be modified.

There are two edit types available for Editable fields:

- **Daily** – fields are available for editing the whole week; the history of changes is stored;
- **Weekly** – fields are available for editing for the current date only; the history of changes is not stored;
- **None** – cannot be modified.

For the ReadOnly fields, only the **None** edit type is available.

- **IsDefaultField** – one of the defined fields per work item type should be selected by default. Default fields allow reporting time in cells with task totals. The reported time will be added to the default field. This possibility is available only if no time was reported on this task before/during the current week or time was reported on default field only.

2. Adding the **ProjectGroupSettings** section.

This section allows making **custom configuration** for a **group of projects**. It is usually used to change settings made in the **DefaultProjectSettings** sections for a group of projects.

Below is an example of the **ProjectGroupSettings** section:

```
<ProjectGroupSettings>
  <LastModificationDate>0001-01-01T00:00:00</LastModificationDate>
  <ProjectGroups>
    <ProjectGroup>
      <Projects>
        <Project>
          <Name>First Project</Name>
          <Uri>vstfs:///Classification/TeamProject/00000000-0000-0000-0000-000000000000</Uri>
        </Project>
        <Project>
          <Name>Second Project</Name>
          <Uri>vstfs:///Classification/TeamProject/00000000-0000-0000-0000-000000000001</Uri>
        </Project>
      </Projects>
      <WorkItemTypes>
        <WorkItemType>
          <Name>Task</Name>
          <Fields>
            <WorkItemTypeField>
              <Name>Completed Work</Name>
              <Alias>CW</Alias>
              <ReferenceName>Microsoft.VSTS.Scheduling.CompletedWork</ReferenceName>
            </WorkItemTypeField>
            <FieldType>Editable</FieldType>
            <EditType>Daily</EditType>
            <IsDefaultField> true</IsDefaultField>
          </WorkItemTypeField>
        </WorkItemType>
      </WorkItemTypes>
    </ProjectGroup>
  </ProjectGroups>
</ProjectGroupSettings>
```

In case **no fields are defined** for a work item type, **work items** of this type **will not be displayed** even if they are specified in the **web.config** file.

In order to **hide all fields** defined in **DefaultProjectSettings** for a particular work item type for a group of projects, one should leave the **Fields** section **empty**.

2.3 Configuring Work Item (WI) States

By default, TX Chrono makes a WI available for reporting time when the WI gets into the “Active” state. To enable WI states other than Active, open the **web.config** file and change the value of the **activeStateTitles** key in the **appSettings** section. The value should contain a list of work item states delimited by commas, for example ‘Active, In Progress, Approved.’

2.4 Adding Virtual Projects

TX Chrono is able to work with work items from TFS projects and tasks from projects defined in TX Chrono itself, the so-called **virtual projects**. These projects are used for adding **tasks** that are **available for all users**, for example Vacation, Day Off, Sickness, etc.

Virtual projects could be defined in the **ApplicationSettings.xml** file. Below is an example of the **VirtualProjectSettings** section where a virtual project called **First Virtual Project** is defined:

```
<VirtualProjectSettings>
  <LastModificationDate>0001-01-01T00:00:00</LastModificationDate>
  <Projects >
    <VirtualProject>
      <Id>-1</Id>
      <Name>First Virtual Project</Name>
      <Tasks>

        <VirtualTask>
          <Id>-1</Id>
          <Title>Vacation</Title>
          <State>Active</State>
          <Fields>
            <VirtualTaskTimeField>
              <Name>Completed Work</Name>
              <Alias>CW</Alias>
              <FieldType>Editable</FieldType>
              <EditType>Daily</EditType>
              <IsDefaultField>true</IsDefaultField>
              <Value></Value>
            </VirtualTaskTimeField>
          </Fields>
        </VirtualTask>
        <VirtualTask>
          <Id>-2</Id>
          <Title>Sickness</Title>
          <State>Active</State>
          <Fields>
            <VirtualTaskTimeField>
              <Name>Completed Work</Name>
              <Alias>CW</Alias>
              <FieldType>Editable</FieldType>
              <EditType>Daily</EditType>
              <IsDefaultField>true</IsDefaultField>
              <Value></Value>
            </VirtualTaskTimeField>
          </Fields>
        </VirtualTask>
      </Tasks>
    </VirtualProject>
  </Projects >
</VirtualProjectSettings>
```

```

<ResponsiblePersons>
  <string>remley.b</string>
</ResponsiblePersons>
<ResponsibleRoles>
  <string>Project Managers</string>
</ResponsibleRoles>
</VirtualProject>
</Projects>
</VirtualProjectSettings>

```

To define a virtual project, one should add a **VirtualProject** section as shown above. This section includes:

- Virtual project **name**.
- Virtual project **id**.

Note: It must be negative for Virtual Project.

- List of **tasks** – each task has the following required fields:
 - **Id** – task id

Note: It must be negative for Virtual Project tasks.

- **Title** – task title that will be displayed on the Timesheet page
- **State** – should always be set to **Active**
- List of **Fields** - each field has the following required tags:
 - **VirtualTaskTimeField** – virtual field declaration
 - **Name** – the name of a work item field
 - **Alias** – the text that will be displayed on the Project Report page instead of the field name (Title attribute, see above). If alias is empty, the task name will be displayed. The alias should be the same for fields with equal names (including fields from the **DefaultProjectSettings** section and the **ProjectGroupSettings** section).
 - **FieldType**
 - **EditType**
 - **IsDefaultField**
 - **Value**
- List of **responsible persons** – a list of logins that are able to **view** and **confirm** the time reported on tasks of a corresponding **virtual project**.
- List of **responsible roles** – a list of **TFS server roles**; **members** of these roles are able to **view** and **confirm** time reported on tasks of a corresponding **virtual project**.

2.5 Configuring Supplementary Work Item Fields for Data Export

Data export is used for integration with third-party applications.

To configure fields for export, add the **ExportField** elements to the **ExoprFields** collection, which is located inside the **ExportSettings** section. For example, the **ExportSetting** section with the field Completed Work configured for export will look like:

```
<ExportSettings>
  <ExportFields>
    <ExportField>
      <Name>Completed Work</Name>
      <Alias>CW</Alias>
      <FieldType>Editable</FieldType>
      <EditType>Daily</EditType>
    </ExportField>
  </ExportFields>
</ExportSettings>
```

Only those fields that were configured as fields of virtual tasks or in the **DefaultProjectSettings** or **ProjectGroupSettings** sections can be added to the list of fields for export.

3. Configuration Parameters Reference

3.1 Abbreviations

Abbreviation	Meaning
TRS	TXChronoTimeReportingService
UI	TXChronoUI

3.2 Configure User Permissions

File	Containing XML Element	Parameter	Description
\\TRS\web.config	configuration	confirmationSection	Edit this section to specify the hierarchy of user permissions.
<i>same</i>	confirmationSection	role	Every item (role) in the hierarchy must represent an existing TFS user group. The users of a group in the hierarchy will be able to view the timesheets of all the descendant groups and confirm the timesheets of the child groups. The hierarchy must have a single top role that has no ancestors.
<i>same</i>	role	name	The name of the role. Must exactly match the name of an existing TFS user group defined on the server or project level
<i>same</i>		hasProjectRoles	False if the TFS user group is defined in TFS on the Server Level True otherwise
<i>same</i>	appSettings	AdminGroupName	The name of the top role in the user groups hierarchy
<i>same</i>		IsAdminGroupNameProjectRole	False if the TFS group corresponding to the top role is defined on the Server Level True otherwise
<i>same</i>		HolidaysAdministratorsGroupName	TFS user group allowed to edit holidays
\\UI\		<location path="Senior">	Locate the <location path="Senior"> element. Find the <allow> element and change its

File	Containing XML Element	Parameter	Description
web.config			value to the list of the roles that have descendants in the hierarchy
<i>same</i>		AdminRoleName	Same as AdminGroupName
\\un Web.Sitemap		siteMapNode	Find the <siteMapNode> elements with the URL attribute starting with the “~/Senior/” string. In the roles attribute specify the roles that have descendants in the hierarchy (and thus will need rights to access the Confirm and View timesheets pages)

3.3 Configure Work Items Types

File	Containing XML Element	Parameter	Description
\\TRS\ web.config	AppSettings	activeStateTitles	List of work item states. Chrono will display a work item in timesheets as the work item gets into a state from this list. It should be a comma delimited list e.g. 'Active, In Progress, Approved.'
\\TRS\Conf igs\ Applicatio nSettings. xml	ApplicationSet tings	DefaultProject Settings	This sections defines what work item fields are to be shown in timesheets
<i>same</i>	WorkItemTypes	WorkItemType	This section describes a work item field
<i>same</i>	WorkItemType	Name	The name of the work item, e.g. Task, Bug
<i>same</i>		Fields	This section describes the fields of the specified work item type
<i>same</i>	Fields	WorkItemType Field	This section describes one of the fields of the work item type
<i>same</i>	WorkItemTyp eField	Name	The name of a work item field (should be the same as in TFS)
<i>same</i>		Alias	The abbreviation that will be displayed on the Project Report page instead of the field name. If alias is empty, the field name will be displayed. Aliases should be equal for fields with the same name.
<i>same</i>		ReferenceName	The reference name of the work item field (should be equal to the reference name of the field in TFS). See the following for the list of field reference

File	Containing XML Element	Parameter	Description
			names available in TFS by default: http://msdn.microsoft.com/en-us/library/ms194971.aspx
<i>same</i>		FieldType	<p>There are two valid values for this field:</p> <p>Editable – the value of the field is available for editing;</p> <p>ReadOnly – the value of the field will be displayed in timesheets but one will not be able to modify it through Tx Chrono.</p>
<i>same</i>		EditType	<p>Defines how the field will be modified.</p> <p>There are two edit types available for Editable fields:</p> <p>Daily – fields are available for editing the whole week; the history of changes is stored;</p> <p>Weekly – fields are available for editing for the current date only; the history of changes is not stored;</p> <p>None – cannot be modified.</p> <p>For the ReadOnly fields, EditType should be always set to None</p>
<i>same</i>		IsDefaultField	<p>One of the defined fields per work item type should be selected by default. Default fields allow reporting time in cells with task totals. The reported time will be added to the default field. This possibility is available only if no time was reported on this task before/during the current week or time was reported on default field only.</p>

3.4 Configure Virtual Projects

Virtual projects contain tasks that are defined in TxChrono itself and does not correspond with work items in TFS.

File	Containing XML Element	Parameter	Description
\\TRS\Conf igs\ Applicatio nSettings.xml	Projects	VirtualProject	This section defines a virtual project

File	Containing XML Element	Parameter	Description
<i>same</i>	VirtualProject	name	Virtual project name
<i>same</i>		Id	Virtual project id (must be negative)
<i>same</i>	Tasks	VirtualTask	This section defines the tasks of the virtual project
<i>same</i>	VirtualTask	id	Task id (must be negative)
<i>same</i>		Title	Task title that will be displayed on the Timesheet page
<i>same</i>		State	Should always be set to Active
<i>same</i>		Fields	This sections defines fields of the virtual task
<i>same</i>	VirtualTaskTimeField	Name	The name of the work item field
<i>same</i>		Alias	The text that will be displayed on the Project Report page instead of the field name (Title attribute, see above). If alias is empty, the task name will be displayed. The alias should be the same for fields with equal names (including fields from the DefaultProjectSettings section and the ProjectGroupSettings section)
<i>same</i>		FieldType	Should be set to Editable
<i>same</i>		EditType	Defines how the field will be modified. There are two edit types available for Editable fields: Daily – fields are available for editing the whole week; the history of changes is stored; Weekly – fields are available for editing for the current date only; the history of changes is not stored
<i>same</i>		IsDefaultField	One of the fields should be marked as Default. This parameter can be to True or False
<i>same</i>		Value	Should be left empty
<i>same</i>	ResponsiblePersons	string	A list of logins that are able to view and confirm the time reported on tasks of a corresponding virtual project
<i>same</i>	ResponsibleRoles	string	A list of TFS server roles . Members of these roles are able to view and confirm time reported on tasks of a corresponding virtual project

4. Known Limitations

4.1 Collection level user groups are not supported

TFS 2010 introduced the possibility to define user groups on the project collection level. At this time TX Chrono does not support such collection-level user groups. Therefore make sure you avoided using collection-level user groups when configuring the TX Chrono role hierarchy as described in 2.1.2.

4.2 Underscore character in role names

The role names included in the TX Chrono role hierarchy must not contain the underscore (_) character.