# Anwender für Dienst erstellen

Erstelle einen User TeamCityOperator, Domänen User, lokaler Administrator. TeamCity Build Agent Service muss unter diesem User laufen. Anmelden unter diesem User. [[Link](http://devnet.jetbrains.com/message/5269541?tstart=0)]( <http://devnet.jetbrains.com/message/5269541?tstart=0>) beschreibt wie die Kultur Info des Users gesetzt werden müssen: alles ist english! Danach den Dienst TeamCity Build Agent Service neustarten.

# TeamCity

[Anweisungen](http://www.troyhunt.com/2010/11/you-deploying-it-wrong-teamcity\_25.html) des Herrn Hunt befolgen.

# Anpassungen

Folgende Anpassungen an der Installation vornehmen:

- Destination Folder: C:\TeamCity

- Configuration Directory: C:TeamCity\.BuildServer

- TeamCity Server Port: 8008

Die "Agent Properties" werden in C:\Teamcity\buildagent\conf\buildAgent.properties gespeichert.

Als Service Account dem TeamCityOperator wählen.

Sollte ein Backup vorhanden sein, diesen wiederherstellen.

Den Pfad zu Java jre in TeamCity als Teil des PATH definieren. Um die Umgebungsvariablen zu definieren:

- Control Panel starten

- System and Security wählen

- System wählen

- Links oben "Advanced System Settings" wählen

- Reiter "Advanced" wählen

- Unten Button "Environment Variables" klicken

- Im Bereich "System Variables" nach Variable Path suchen

- Variable Path um den "C:\TeamCity\jre\bin" Directory erweitern. So kann immer eine Java Version gefunden werden.

### Nach der Installation von TeamCity und vor der Beginn der Einrichtgung

Domänen User SRV\_AUTOACT in die Gruppe der Administratoren aufnehmen.

In der oberen Menuleiste Agents kontrollieren: falls „1“ steht dann ist ein Agent bereits eingebunden und kann verwendet werden. Falls nicht bitte die Anleitung in <http://devnet.jetbrains.com/thread/311239> befolgen.

ACHTUNG Vor der Installation Team Explorer deinstallieren

## Einrichtung Projekt AutoAct

Neuen Projekt anlegen.

Tab VCS Roots wählen, „Create VCS Root“ klicken.

Type of VCS: Team Foundation Server

VCS root name und ID: vms047

URL: <http://vms047:8080/tfs/defaultcollection/>

Root: $/Migration/Work

Agent checkout: Enforce Overwrite All Files = True

Checking interval: use global server setting (60 seconds)

WICHTIG: “Test connection” klicken und speichern.

# Einrichtung der Build Konfiguration

Im Tab Projekt\General „Create Build Configuration“ wählen

Name: Build

Description: Build für AutoAct

Build Number Format: 1.0.%build.counter%.{build.vcs.number.1}

Erklärung: die %build.counter% ist Platzhalter für die Build Number wie diese von TeamCity definiert wird.

{build.vcs.number.1} ist TeamCity Platzhalter für die VCS interne Revisionsnummer, abhängig vom Type des CVS.

Button VCS Settings wählen

## Einrichtung des VCS Roots

Attach Exisiting VCS Root, dort den VCS Root das im Schritt „Einrichtung Projekt WorkAutoAct“ erstellt wurde. VCS Root erscheint oben, dort „Edit Checkout Rules“ klicken und folgendes eintragen:

+:.=>/

Erklärung: <http://confluence.jetbrains.com/display/TCD8/VCS+Checkout+Rules>

+: Include

. Root

=> Checkout in

/ Work Verzeichnis von TeamCity

„Add Build Step“ klicken

## New Build Step

Runner Typ ist Visual Studio (sln)

Step Name: Build

Solution File Path: SapORM\SapORM.sln

Visual Studio: Microsoft Visual Studio 2010

Targets: Rebuild

Configuration: Debug

Speichern.

## New Build Step

Runner Typ ist Visual Studio (sln)

Step Name: Build

Solution File Path: AutoAct\AutoAct\AutoAct.sln

Visual Studio: Microsoft Visual Studio 2010

Targets: Rebuild

Configuration: Debug

Speichern.

### Build Trigger

In der Menuleiste auf der rechten Seite bitte Build Trigger wählen

Add New Trigger klicken

VCS Trigger wählen

Trigger a Build on each Check in = true

Include Several check-ins in a build if from the same commiter = true

Quite Period Use Default Value

Speichern

# Build Configuration DeployDebug

Name: DeployDebug

ID: AutoAct\_DeployDebug

Build number format: 1.0.%build.counter%.{build.vcs.number.1}

VCS Einstellungen wie oben beschrieben.

Definiere drei Schritte:

* SAPORM build: ist eine Kopie des ersten Schrittes aus der Konfiguration AutoAct\Build.
* Build: ist eine Kopie des Schrittes Build aus der Konfiguration AutoAct\Build. Dies ist notwendig da wir sowohl einen Debug als auch einen Release Build haben. ACHTUNG: Configuration ist Debug
* Deploy: kopiert den Inhalt des Verzeichnises bin\Debug in C:\Programme\autoAct\Debug
* Eine Dependency erstellen:
  + Depend on: AutoAct :: Build
  + Do not run new build if there is a suitable one = true
  + Only use successful builds = true

# Build Configuration DeployRelease

Name: DeployRelease

ID: AutoAct\_DeployRelease

Build number format: 1.0.%build.counter%.{build.vcs.number.1}

VCS Einstellungen wie oben beschrieben.

Definiere drei Schritte:

* SAPORM build: ist eine Kopie des ersten Schrittes aus der Konfiguration AutoAct\Build.
* Build: ist eine Kopie des Schrittes Build aus der Konfiguration AutoAct\Build. Dies ist notwendig da wir sowohl einen Debug als auch einen Release Build haben. ACHTUNG: Configuration ist Release
* Deploy: kopiert den Inhalt des Verzeichnises %system.teamcity.build.checkoutDir%\AutoAct\AutoAct\bin\Release in C:\Programme\autoAct\Release
* Eine Dependency erstellen:
  + Depend on: AutoAct :: Build
  + Do not run new build if there is a suitable one = true
  + Only use successful builds = true

# Build Configuration ResetDebug

Name: ResetDebug

ID: AutoAct\_ResetDebug

Build number format: 1.0.%build.counter%.{build.vcs.number.1}

VCS Einstellungen: Keine. Konfiguration ist nicht vom CVS abhängig.

Definiere einen Schritte: Aufruf der Exe aus dem Debug Verzeichnis mit dem Paramter reset

C:\Programme\AutoAct\Debug\AutoAct.exe reset

## Sicherung

### Erstellung eines Sicherungsarchiv

Benötigte Umgebungsvariablen erstellen. Bitte als Systemvariablen erstellen

TEAMCITY\_APP\_DIR: Default ist *C:\TeamCity\webapps\ROOT*

TEAMCITY\_DATA\_PATH: Wert bitte der site/admin Seite entnehmen

Ausführung von TeamCity-Backup.cmd profen.

### Sicherung des Sicherungsarchiv

Mit Hilfe von Cobian soll eine Sicherung des Sicherungsarchiv erstellt werden.

# Bekannte Probleme

Hatte ein Problem mit dem Build Agent http://devnet.jetbrains.com/message/5316796?tstart=0

Der build Agent könnte nicht verwendet werden da ihm dot.net 4.0 fehlte. Was klar nicht stimmte. Müsste die Verzeichnisse im BuildAgent Verzeichnis löschen und TeamCity hat diese neugeladen. Danach ging es.

# Verbindungen

Sollte es notwendig sein Serveradressen zu verwenden (vs benannte Server), Mapping in der hosts Datei erstellen.

WICHTUG: telnet Tests führen für alle notwendigen ports/Adressen Kombinationen.

### Benachrichtigungen

TeamCity verwaltet die Benachrichtigungen per User und nicht als Server Einstellung. Also ich kann mich als Anwender anmelden und dann für mich einen Abo auf Benachrichtigungen erstellen, per Email z.B.