

Version Control Snapshot

Serviço para fazer um snapshot de arquivos e configurações mapeados de um computador, uma rede e/ou um banco de dados e armazenar os arquivos correspondentes no TFS. As modificações ocorridas poderão ser verificadas no histórico do TFS, comparando versões diferentes de um mesmo arquivo.

Pré-requisitos

- Espaço em disco (equivalente a 2 vezes o tamanho em disco dos dados que serão controlados pelo serviço)
- Instalação do Visual Studio, conforme compatibilidade de suas DLLs do TFS com a versão do Windows do servidor:

Microsoft.TeamFoundation.Client.dll Microsoft.TeamFoundation.Common.dll Microsoft.TeamFoundation.VersionControl.Client.dll Microsoft.TeamFoundation.VersionControl.Common.dll	Versão 10	.Net Framework 4	Visual Studio 10.0	Windows Server 2008 R2 Windows Server 2008 (minimum SP2)
Microsoft.TeamFoundation.Client.dll Microsoft.TeamFoundation.Common.dll Microsoft.TeamFoundation.VersionControl.Client.dll Microsoft.TeamFoundation.VersionControl.Common.dll	Versão 15.129	.Net Framework 4.5	Visual Studio 2017	*Windows Server 2012 R2 *Windows Server 2012 Windows Server 2016 Windows Server 2008 R2 (minimum SP1)

*** COMPATIBILIDADES TESTADAS E VERIFICADAS**

* o ANEXO 1 possui os detalhes das compatibilidades

ATENÇÃO: Verificado que não é compatível com o Windows Server 2003

Problemas conhecidos

- As pastas de destino dos arquivos não podem estar sendo usadas por outro processo. Por exemplo, não podem ficar abertas no Windows Explorer durante a execução do serviço.

Etapas de funcionamento do serviço

Etapas 1 (Serviço):

Controle de gerenciamento do serviço.

Etapas 2 (Geração de arquivos):

Geração dos arquivos contendo os dados do snapshot.

Etapas 3 (TFS):

Identificação dos arquivos a serem enviados ao repositório e realização do check-in.

ETAPA 1 (Serviço)

Configurações no App.config:

```
<setting name="NumberOfTries" serializeAs="String">  
  <value>3</value>  
</setting>
```

Número máximo sequencial de tentativas que a geração dos arquivos e processo de check-in estarão submetidos, caso ocorram erros durante a execução do serviço.

```
<setting name="LogFileName" serializeAs="String">  
  <value>LastStatus.log</value>  
</setting>
```

Arquivo de log para registro de sucesso ou erro após execução do serviço. Caso esse arquivo não esteja disponível no TFS, o conteúdo do mesmo deverá ser verificado no servidor local, no caminho definido na configuração "TempRepositoryPathToCurrentFiles", pois, provavelmente, ocorreu algum erro ao fazer o check-in do mesmo.

ETAPA 2 (GERAÇÃO DE ARQUIVOS)

Como funciona os mapeamentos?

Os dados para geração dos arquivos são informados em arquivos de configuração cuja localização está informada no *app.config*. Cada gerador de arquivos possui um modo e/ou regra de geração diferente. Por exemplo, um controla a criação de arquivos a partir de consultas no banco de dados, outro a partir de processos externos, outro através de cópias de arquivos físicos etc.

Segue abaixo os geradores de arquivos presentes no serviço:

- MAPEAMENTO DE PASTAS
- MAPEAMENTO DE PROCESSOS
- MAPEAMENTO DE CONSULTAS NO BANCO DE DADOS

Configuração gerais

Configurações no App.config:

```
<setting name="LocalAdministratorUserName" serializeAs="String">
  <value>Administrator</value>
</setting>
<setting name="LocalAdministratorPassword" serializeAs="String">
  <value>123456789</value>
</setting>
```

Obs: Credenciais locais de administrador para iniciar processos.

MAPEAMENTO DE PASTAS

Configurações no App.config:

```
<setting name="MapFolderPropertiesSource_FilePath" serializeAs="String">
  <value>.\Configurations\MapFolderPropertiesSource.xml</value>
</setting>

<setting name="NetworkPathAccessSettings_FilePath" serializeAs="String">
  <value>.\Configurations\NetworkPathAccessSettings.xml</value>
</setting>
```

Obs: Caso o valor seja deixado em branco (<value></value>), não serão criados arquivos a partir do respectivo gerador de arquivos.

Exemplo:

```
<MapFolderProperties>
  <MapFolderPopertyItem>
    <Source>\\10.10.215.114\c$\Inetpub\wwwroot\Site</Source>
    <TargetFolder>PainelControle</TargetFolder>
  </MapFolderPopertyItem>
  <MapFolderPopertyItem>
    <Source>C:\inetpub\wwwroot\PortalWeb</Source>
    <TargetFolder>SPortal</TargetFolder>
  </MapFolderPopertyItem>
</MapFolderProperties>
```

Source: Endereço completo (full path) no servidor ou na rede que contém todos os arquivos a serem copiados.

TargetFolder: Pasta de destino dos arquivos a serem gerados.

Observações:

Caso haja mapeamentos de pastas na rede, os dados de autenticação de usuários devem ser informados no arquivo "NetworkPathAccessSettings.xml", conforme exemplo:

```
<NetworkPathAccessSettings>
  <NetworkPathAccessItem>
    <NetAddress>10.10.215.114</NetAddress>
    <NetUser>Administrator</NetUser>
    <NetPassword>123456</NetPassword>
  </NetworkPathAccessItem>
</NetworkPathAccessSettings>
```

NetAddress: IP ou nome do servidor na rede.

NetUser: Nome do usuário para autenticação.

NetPassword: Senha do usuário para autenticação.

MAPEAMENTO DE PROCESSOS

Configurações no App.config:

```
<setting name="ProcessCommandLinePropertiesSource_FilePath" serializeAs="String">  
  <value>.\Configurations\ProcessCommandLinePropertiesSource.xml</value>  
</setting>
```

Obs: Caso o valor seja deixado em branco (<value></value>), não serão criados arquivos a partir do respectivo gerador de arquivos.

```
<setting name="GenerateCADParametersForAllSetNames" serializeAs="String">  
  <value>True</value>  
</setting>
```

Obs: Valores possíveis: "True" ou "False". Caso seja "True" os "set names" definidos no arquivo "cfg_param_unload_data_template.ctf" serão substituídos por todos os "set names" do cliente.

Exemplo:

```
<ProcessCommandLineProperties>  
  <ProcessCommandLinePropertyItem>  
    <ProcessFileName>regedit.exe</ProcessFileName>  
    <ProcessArguments>/e "[TARGET_FOLDER_FULL_PATH]\REG_NAME.reg"  
    "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Example"</ProcessArguments>  
    <TargetFolder>CAD\WIN_REGISTRY</TargetFolder>  
  </ProcessCommandLinePropertyItem>  
</ProcessCommandLineProperties>
```

ProcessFileName: Nome ou endereço completo do arquivo executável.

ProcessArguments: Argumentos a serem passados ao processo.

TargetFolder: Pasta de destino dos arquivos a serem gerados.

Substituições automáticas de textos:

O trecho "[TARGET_FOLDER_FULL_PATH]" passado como argumento é substituído pelo endereço de destino padrão ("*TempRepositoryPathToCurrentFiles*").

MAPEAMENTO DE CONSULTAS NO BANCO DE DADOS

Configurações no App.config:

```
<setting name="SqlPropertiesSource_FilePath" serializeAs="String">  
  <value>.\Configurations\SqlPropertiesSource.xml</value>  
</setting>
```

Obs: Caso o valor seja deixado em branco (<value></value>), não serão criados arquivos a partir do respectivo gerador de arquivos.

Exemplo:

```
<SqlProperties>  
  <SqlPropertyItem>  
    <SQL>SELECT ID, NAME FROM PRODUCT</SQL>  
    <TargetFolder> SQL</TargetFolder>  
    <FileName>Product.xml</FileName>  
    <ConectionName>MAIN</ConectionName>  
  </SqlPropertyItem>  
</SqlProperties>
```

SQL: Consulta SQL (Query) a ser executada no banco de dados

FileName: Nome do arquivo que será criado com os resultados da consulta convertido em XML.

TargetFolder: Pasta de destino dos arquivos a serem gerados.

ConectionName: Nome da conexão de acordo com as conexões configuradas em "connectionStrings" do ".config" do serviço.

ETAPA 3 (TFS)

Configurações no App.config:

Configurações referentes às **credenciais de acesso ao TFS**:

```
<setting name="TfsUserName" serializeAs="String">
  <value>DOMINIO\user-name</value>
</setting>
<setting name="TfsPassword" serializeAs="String">
  <value>123456789</value>
</setting>
<setting name="TfsOwner" serializeAs="String">
  <value>tfs-owner</value>
</setting>
```

Obs: Esse será o usuário padrão que será utilizado em todos os clientes

Configurações referentes aos **mapeamentos dos repositórios**:

```
<setting name="CollectionURI" serializeAs="String">
  <value>http://197.12.113.20:8080/tfs/companycollection</value>
</setting>
<setting name="WorkspaceName" serializeAs="String">
  <value>WORKSPACE-NAME</value>
</setting>
<setting name="ServerRepositoryPath" serializeAs="String">
  <value>$/CLIENT1/SHOT</value>
</setting>
<setting name="LocalRepositoryPath" serializeAs="String">
  <value>D:\SHOT\FILES</value>
</setting>
```

Obs: Caso o espaço de trabalho (workspace) não exista, será criado um novo e serão feitos os mapeamentos dos repositórios local e remoto conforme configurado.

ATENÇÃO: Conceder permissão de escrita à pasta criada em "LocalRepositoryPath"

Configurações referentes ao **CHECK-IN**:

```
<setting name="DownloadFilesFromTFS" serializeAs="String">
  <value>True</value>
</setting>
```

Obs: Valores possíveis: "True" ou "False". Quando "True", diariamente é feito o download de todos os arquivos do TFS, substituindo o repositório local. Recomenda-se manter essa opção ativa ("True"), porém ela é a ação mais lenta do serviço, retardando bastante sua conclusão. Sendo assim, essa opção poderá ser desativada depois da segunda vez que o serviço for rodado, caso o repositório não sofra por intervenções externas a esse serviço.

```
<setting name="TempRepositoryPathToCurrentFiles" serializeAs="String">
  <value>D:\SHOT\FILES-TEMP</value>
</setting>
```

Obs: Essa pasta armazenará todos os arquivos resultantes dos geradores de arquivos e servirá de base para atualização do repositório. Ou seja, se algum gerador de arquivos for desativado, seus respectivos arquivos serão excluídos do repositório do TFS.

ATENÇÃO: Conceder permissão de escrita à pasta criada em "TempRepositoryPathToCurrentFiles"


```
<setting name="CheckInComment" serializeAs="String">
  <value>Check-In automático</value>
</setting>
```

Obs: Mensagem padrão agregada ao check-in. O serviço realiza ações de check-in distintas para edição, adição ou exclusão de arquivos, mantendo essa mensagem padrão nos três casos.

CONDIÇÕES PARA ATUALIZAÇÃO DE ARQUIVOS NO TFS

Configurações no App.config:

```
<setting name="IgnoreFilePathWords" serializeAs="String">
  <value>rdDataCache|rdDownload</value>
</setting>
```

Obs: Exclui todos os arquivos da pasta configurada em `TempRepositoryPathToCurrentFiles` que possuem as palavras separadas por “|” em seu caminho absoluto.

Ex: O arquivo “SiteWeb\rdDataCache\whatever.txt” será excluído por conter “rdDataCache” em seu caminho absoluto.

```
<setting name="DiffFilesAttributesMap_FilePath" serializeAs="String">
  <value>.\Configurations\DiffFilesAttributesMap.xml</value>
</setting>
```

Obs: Caso o valor seja deixado em branco (`<value></value>`), o arquivo será enviado se houver alguma diferença detectada pelo serviço do TFS.

Exemplo:

```
<DiffFilesAttributes>
  <DiffFilesAttributesItem>
    <Path>CAD\CADDBM\PARAMETERS</Path>
    <FileTransferCondition>IF_DIFF</FileTransferCondition>
    <XmlDiffOptions>IgnoreChildOrder|IgnoreNamespaces|IgnoreComments
  </XmlDiffOptions>
</DiffFilesAttributesItem>
</DiffFilesAttributes>
```

Path: Pasta cujos arquivos serão verificados

FileTransferCondition: Verifica se o arquivo será copiado baseado nas condições:

- IF_DIFF: compara a data da última modificação
- ALWAYS: substitui sempre
- NEVER: nunca substitui (essa opção pode ser utilizada para impedir que pastas indesejadas sejam enviados ao TFS)

XmlDiffOptions: Depois da verificação se o arquivo deverá ser copiado através do valor definido em “FileTransferCondition”, os arquivos do tipo XML podem ser submetidos a uma segunda verificação, escolhendo uma ou mais das opções de comparação a seguir.

- **IgnoreChildOrder:** A ordem dos nós filhos de cada elemento é ignorada quando VERDADEIRO. Quando essa opção é selecionada, dois nós com o mesmo valor que diferem apenas por sua posição entre os nós filhos irmãos são tratados como os mesmos nós.
- **IgnoreComments:** Os comentários dos nós não são comparados quando VERDADEIRO.
- **IgnoreDtd:** A declaração de tipo de documento (DTD) não é comparada quando VERDADEIRO.
- **IgnoreNamespaces:** As “URIs” de namespace dos nomes de elemento e atributo não são comparados quando VERDADEIRO. Esta opção também implica que os prefixos de nomes são ignorados. Quando essa opção é selecionada, dois nomes com o mesmo nome local, mas com um URI e prefixo de namespace diferentes, são tratados como os mesmos nomes.
- **IgnorePI:** As instruções de processamento não são comparadas quando VERDADEIRO.
- **IgnorePrefixes:** Os prefixos de nomes de elemento e atributo não são comparados quando VERDADEIRO. Quando essa opção é selecionada, dois nomes que têm o mesmo nome local e URI de namespace, mas têm um prefixo diferente, são tratados como os mesmos nomes.
- **IgnoreWhitespace:** Espaços brancos significativos não são comparados quando VERDADEIRO e todos os nós de texto são normalizados descartando-se os caracteres de espaço em branco inicial e final (#x9, #x10, #x13, #x20) e substituindo as sequências de caracteres de espaço em branco por um único caractere de espaço (#x20) simples.
- **IgnoreXmlDecl:** A declaração XML não é comparada quando VERDADEIRO.
- **None:** Especifica que todas as opções acima são FALSAS.

Caso seja escolhida mais de uma opção, as mesmas deverão ser separadas por “|”. Ex:
IgnoreChildOrder|IgnoreNamespaces|IgnoreComments

Link da documentação oficial sobre a “enumeration” XmlDiffOptions (Microsoft XML Diff):
https://documentation.help/Microsoft-XML-Diff/xmldiffpatch_xmldiffoptions_ref_1tsz.htm

Atenção: Utilizar “IgnoreComments” na pasta de destino dos arquivos que guardam a configuração do CADDBM é uma boa prática, uma vez que cada vez que o processo cfg_param_utility_console.exe é executado são criados arquivos XML com a data de criação registrada em comentários.

```
<Path>PARAMETERS</Path>
<FileTransferCondition>IF_DIFF</FileTransferCondition>
<XmlDiffOptions>IgnoreChildOrder|IgnoreNamespaces|IgnoreComments
</XmlDiffOptions>
```

DESENVOLVIMENTO

Depuração:

Deve-se executar o Visual Studio como administrador, para elevar a permissão para manipulação de arquivos (cópia, edição e exclusão).

Criação de um novo gerador de arquivos:

- 1) Criar um “file builder”:
 - a. Criar uma classe que estende a classe abstrata `“VersionControlSnapshot.Builders.FileBuilder”`.
 - b. Implementar o método `“BuildFiles()”` na nova classe.
- 2) Criar uma “factory”:
 - a. Criar um arquivo XML de configuração do novo Builder na pasta `“Configurations”`.
 - b. Habilitar a propriedade `“Copiar para Diretório de Saída”` do novo arquivo de configuração para fazer a cópia.
 - c. Criar uma classe que estende a classe abstrata `“VersionControlSnapshot.Factories.FileBuilderFactory”`.
 - d. Implementar o método `“CreateFileBuilder()”` na nova classe.
 - e. Adicionar a nova “factory” ao objeto `FileBuilderManager` instanciado na função `BuildFiles()` da classe `“VersionControlSnapshot.Services.ServerRepositoryUpdaterService”`

ANEXO 1

(compatibilidade versão das DLLs do TFS com versão do Windows)

Package and TFS version mapping table

PACKAGE AND TFS VERSION MAPPING TABLE	
Package version	TFS Version
17.x.x	Azure DevOps Server vNext
16.153.x	versions >= Azure DevOps Server 2019 Update 1
16.143.x	versions >= Azure DevOps Server 2019
15.131.x	versions >= TFS 2018 Update 1
15.112.x	versions >= TFS 2017 Update 1
14.102.x	versions >= TFS 2015 Update 3
14.95.x	versions >= TFS 2015 Update 2
14.89.x	versions >= TFS 2015 Update 1
14.83.x	versions >= TFS 2015

Server operating systems

SERVER OPERATING SYSTEMS	
Azure DevOps Server or TFS version	Supported server operating systems
Azure DevOps Server 2020	Windows Server 2020 Windows Server 2019 Windows Server 2016
Azure DevOps Server 2019	Windows Server 2019 Windows Server 2016 Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter)
TFS 2018	Windows Server 2016 Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter)
TFS 2017	Windows Server 2016 Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter) Windows Server 2008 R2 (minimum SP1) (Standard, Enterprise, Datacenter)
TFS 2015	Windows Server 2016 Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter)

SERVER OPERATING SYSTEMS	
Azure DevOps Server or TFS version	Supported server operating systems
	Windows Server 2008 R2 (minimum SP1) (Standard, Enterprise, Datacenter)
TFS 2013	Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter) Windows Server 2008 R2 (minimum SP1) (Standard, Enterprise, Datacenter)
TFS 2012	Windows Server 2012 R2 (Essentials, Standard, Datacenter) Windows Server 2012 (Essentials, Standard, Datacenter) Windows Server 2008 R2 (Standard, Enterprise, Datacenter) Windows Server 2008 (minimum SP2) Windows Small Business Server 2011 (Standard, Essentials, Premium Add-On) Windows Home Server 2011
TFS 2010	Windows Server 2008 R2 (Standard, Enterprise, Datacenter) Windows Server 2008 (minimum SP2) Windows Server 2003 R2 Windows Server 2003 (minimum SP2)