

Technical Report

IBM Software Solutions, Research Triangle Park, North Carolina, USA

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Lee Perlov

TR 29.3090

October 10, 1998

TeamConnection Development
IBM Software Solutions
Research Triangle Park, North Carolina

© Copyright 1998, IBM Corp.

Abstract

The VisualAge TeamConnection Enterprise Server Version 3.0 provides features that allow users to easily distribute their work products through a web page. Using VisualAge TeamConnection, code that is managed in a VisualAge TeamConnection family can easily be built, packaged, and shadowed to a directory so that a web page pointing to the shadow directory can always provide access to the currently committed baseline of a product.

This is the third in a series of technical reports reflecting the evolution of one VisualAge TeamConnection family. This report covers how to use the VisualAge TeamConnection build server, the shadowing facility and some of the special features in support of the Web that allow for a very efficient distribution mechanism using a web page interface. It expands on build concepts described in the first report in this series.

ITIRC Keywords

- VisualAge TeamConnection
- Web Interface
- Build

About The Author

Lee R. Perlov

Mr. Perlov is a staff software engineer on the VisualAge TeamConnection/CMVC development team. He joined IBM in 1985 in Gaithersburg, MD, working in the Federal Systems Division on various projects for the United States intelligence community. He then moved to RTP to work on library development and support as part of the CMVC and VisualAge TeamConnection development teams.

Mr. Perlov received a B.S. degree in Accounting from the University of Florida in 1983. He also completed two years of graduate work in the Department of Computer Science at the University of Florida.

Mr. Perlov is currently a member of the VisualAge TeamConnection Services team, as well as family, system and web administrator for the services TeamConnection family.

Abstract	2
About The Author	3
The Big Picture	6
VisualAge TeamConnection Build	8
Build Tree	9
Build script to create ZIP files	10
Zipbuild build message	11
Build script to create self-extracting ZIP files	13
Null Build script to simplify managing a build tree	14
Keywords in files	15
Shadowing	16
Accessing with the Web	17
Securing data	21
Other advantages	21

Using a web page to monitor family server activity	22
Final Thoughts	29
Bibliography	30
Copyrights, Trademarks and Service marks	31

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

The Big Picture

VisualAge TeamConnection Services, one of the departments in VisualAge TeamConnection development, created a VisualAge TeamConnection family in mid-1997 to store their documents, class materials, tools, etc. During the evolution of this family, we began to use the VisualAge TeamConnection build facility. We recently migrated our family to VisualAge TeamConnection Enterprise Server Version 3. This article discusses our current use of the build facility, featuring the enhancements in VisualAge TeamConnection Version 3.

In order to ensure that our product would build reliably and that we could track the results of our builds, we used the TeamConnection build facility. Our build tree consisted of:

- Output parts: the results of each build (mostly ZIP files and self-extracting ZIP files)
- Input parts: the source files (documents, presentations, lab exercises, web pages and tools)
- Builders: the command files that generate ZIP files, self-extracting ZIP files, etc.

VisualAge TeamConnection keeps track of the relationships between inputs and outputs and tells us what needs to be built.

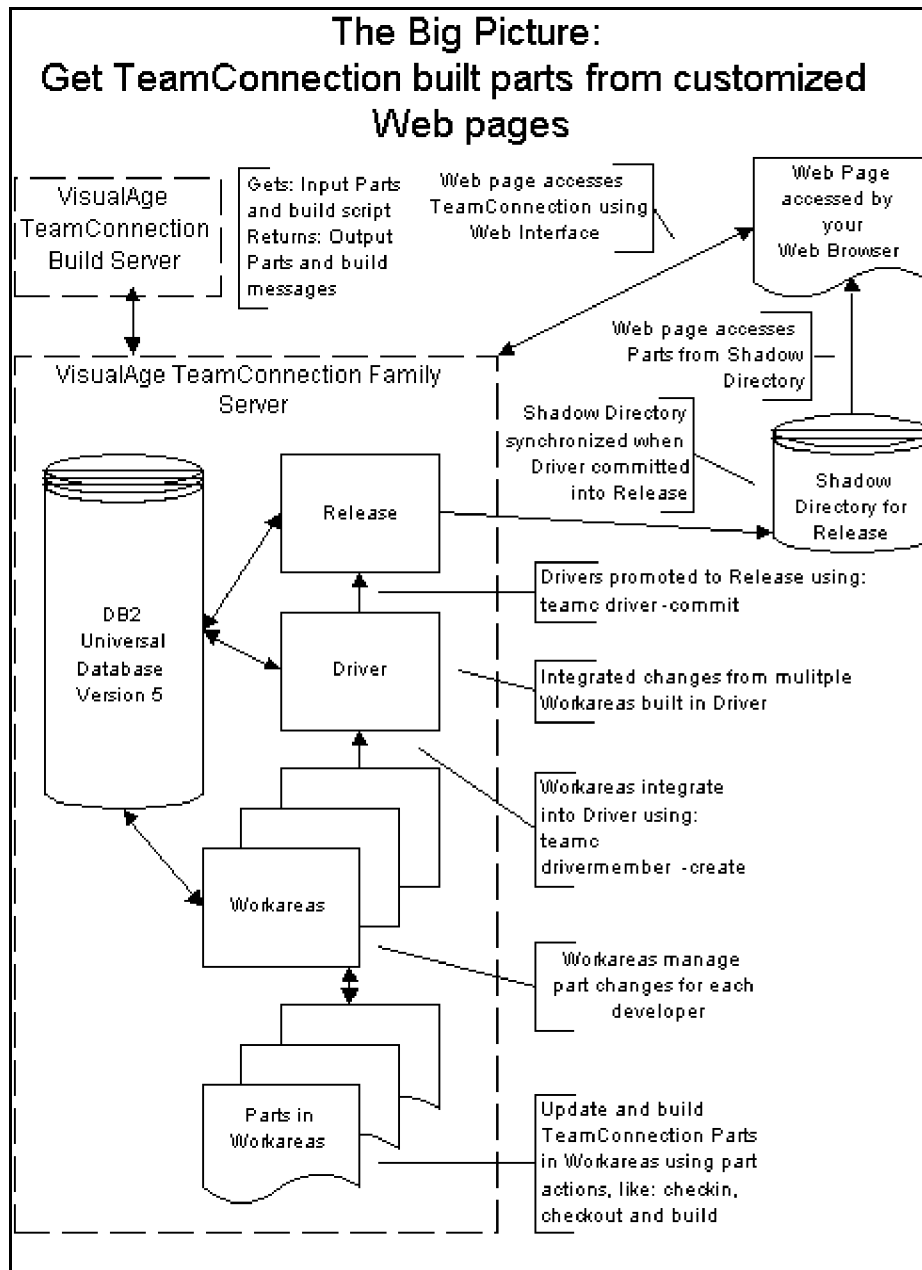
Once we built our product, we needed to distribute it to team members, some of whom did not have access to our VisualAge TeamConnection family. Using the VisualAge TeamConnection shadowing feature added in Version 3, we always distribute the proper version of our product through a shadow directory configured to contain the exact same parts as the committed Release.

To make the contents of the directory accessible to everyone, a web page on our family server points to the shadow directory. Anyone with a web browser on our intranet with password access to this page can select specific output parts listed on the page or browse the shadow directory pointed to by the page, then download the parts.

The example web page and VisualAge TeamConnection build tree provided here are customized for a small family, testfam, used to prototype changes to our production family.

Using VisualAge TeamConnection build and shadowing in our process removes manual intervention from one of the most common problem areas in any development process, building exactly what we have stored in our library and then distributing it to those that need the product.

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute



Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

VisualAge TeamConnection Build

To make delivery of our products easier, we create self-extracting ZIP files for each product, such as the class materials in **prz.exe** and supporting information in **info.exe**. To make sure everything gets built, we have a collector object that doesn't result in any output file, **top**. In order to create the self-extracting ZIP files there are three VisualAge TeamConnection build scripts:

- **zipbuild**: Creates a ZIP output file from the input files connected to the ZIP file
- **zip2exe**: Create a self-extracting ZIP output file from the input ZIP file
- **null**: Convenience build script creates no output and is used as a collector

Here are the features of VisualAge TeamConnection that were used when creating the build scripts:

- Build scripts can be any executable. We use DOS batch for Windows NT. This is easily portable to OS/2.
- Build environment variables set by VisualAge TeamConnection identify the build context (i.e. Family, Release, Workarea). We use these to set the ZIP file heading so that anyone installing the ZIP file can identify build context.
- The what magic cookie, **@(#)**, is included in the ZIP file heading includes so that the **what** (in Unix) and **tcwhat** (in Windows NT and OS/2) commands can be used to identify the build context after the ZIP file is delivered to another system.
- Build output is stored as build messages for each VisualAge TeamConnection build performed. Our build scripts display the values of many build environment variables to aid in debugging and verifying that each build performed properly.
- Input parts are extracted to the build server for each build. During the extract, any VisualAge TeamConnection keywords in text parts are expanded. We use these in our text parts to identify the version of the part being included in the ZIP file.

Currently, we start our build server on the same system as our family server. This was the easiest place to start. When our family gets busier, we will move the build server to a separate system.

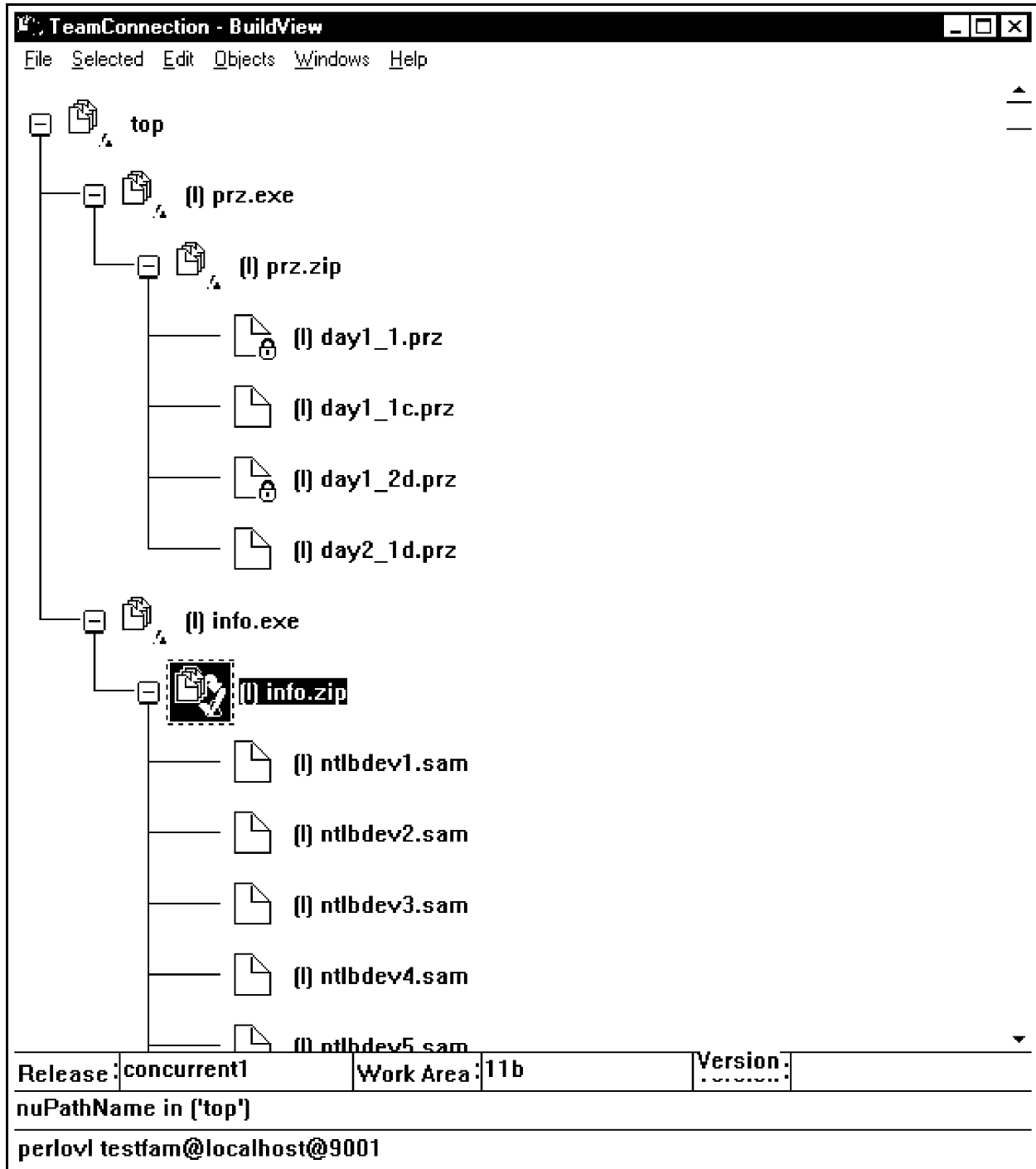
We also run an application to monitor family activity. Here are the commands we use:

Process	Command
family server	E:\TeamC\BIN\TEAMCD.EXE -n mailexit testfam 2
build server	E:\TeamC\BIN\TEAMCBLD.EXE -c e:\teamc\testfam\build -e NT -p bldsock -f testfam -l buildnt -b bldadmin
monitor tool	E:\TeamC\testfam\htm_mon.exe e:\teamc\testfam\monitor.htm 5 10

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Build Tree

Our build tree is constructed by running the builders described in this article to generate the outputs from the inputs. Also shown is a sample of the output of one of these builders.



Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Build script to create ZIP files

A build script converts inputs to outputs. A build script can be any executable. This DOS Batch command file does several informative and useful tasks:

- Displays environment variables set by VisualAge TeamConnection build facility
- Uses environment variables to create text string for build scripts
- Checks results of ZIP program and returns error if ZIP fails
- Verifies result by checking and displaying contents of ZIP file

```
@echo off
REM $KW=@(#); $ChkD=1998/08/08 12:24:09; $FN=zipbuild.cmd; $Own=perlovl; $Ver=10:1;
REM $EKW;
echo VisualAge TeamConnection Version 3 Zip file builder
echo -TC_BUILDERPARMS: %TC_BUILDERPARMS%
echo -TC_BUILD_USER: %TC_BUILD_USER%
echo -TC_FAMILY: %TC_FAMILY%
echo -TC_RELEASE: %TC_RELEASE%
echo -TC_LANG: %TC_LANG%
echo -TC_LOCATION: %TC_LOCATION%
echo -TC_INPUT: %TC_INPUT%
echo -TC_INPUTTYPE: %TC_INPUTTYPE%
echo -TC_OUTPUT: %TC_OUTPUT%
echo -TC_OUTPUTTYPE: %TC_OUTPUTTYPE%
echo -TC_WORKAREA: %TC_WORKAREA%
echo.
echo Build Start
echo @(#) Package context: Family %TC_FAMILY%, Release %TC_RELEASE%, Workarea
%TC_WORKAREA% > tmpfile
@echo on
zip -z %TC_OUTPUT% %TC_INPUT% < tmpfile
@if errorlevel 1 exit
@echo off
erase tmpfile
echo -----
dir %TC_OUTPUT%
echo Verifying contents of zip file
@echo on
unzip -t %TC_OUTPUT%
@if errorlevel 1 exit
@echo Processing complete
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Zipbuild build message

This is an example of the output of this build script, creating the **info.zip** output file from the connected input files as the result of executing a build in VisualAge TeamConnection. The resulting **info.zip** is checked into VisualAge TeamConnection.

```
There is not a parser associated with this
part object.
***** Messages from the builder *****
Version: testfam|concurrent1|10
Builder: buildzip3.cmd
Output Files: info.zip
Input Files: ntlbdev1.sam ntlbdev2.sam ntlbdev3.sam ntlbdev4.sam ntlbdev5.sam
ntlbdev6.sam ntlbdev7.sam trtcoem.pdf trtcvscm.pdf README.txt
Command: buildzip3.cmd
RC=0      (Expected: == 0) - Success
***** Build Output Follows *****
VisualAge TeamConnection Version 3 Zip file builder
-TC_BUILDERPARMS:
-TC_BUILD_USER: perlovl
-TC_FAMILY: testfam
-TC_RELEASE: concurrent1
-TC_LANG: ENU
-TC_LOCATION: e:\teamc\testfam\build\tb6c
-TC_INPUT: ntlbdev1.sam ntlbdev2.sam ntlbdev3.sam ntlbdev4.sam ntlbdev5.sam
ntlbdev6.sam ntlbdev7.sam trtcoem.pdf trtcvscm.pdf README.txt
-TC_INPUTTYPE: TcPart TcPart TcPart TcPart TcPart TcPart TcPart TcPart TcPart
-TC_OUTPUT: info.zip
-TC_OUTPUTTYPE: TcPart
-TC_WORKAREA: 10

Build Start

e:\teamc\testfam\build\tb6c>zip -z info.zip ntlbdev1.sam ntlbdev2.sam ntlbdev3.sam
ntlbdev4.sam ntlbdev5.sam ntlbdev6.sam ntlbdev7.sam trtcoem.pdf trtcvscm.pdf
README.txt 0<tmpfile
  adding: ntlbdev1.sam (deflated 86%)
  adding: ntlbdev2.sam (deflated 86%)
  adding: ntlbdev3.sam (deflated 84%)
  adding: ntlbdev4.sam (deflated 87%)
  adding: ntlbdev5.sam (deflated 88%)
  adding: ntlbdev6.sam (deflated 90%)
  adding: ntlbdev7.sam (deflated 86%)
  adding: trtcoem.pdf (deflated 13%)
  adding: trtcvscm.pdf (deflated 16%)
  adding: README.txt (deflated 61%)
enter new zip file comment (end with .):
-----
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

```
Volume in drive E is Dev1
Volume Serial Number is 68FF-2383

Directory of e:\teamc\testfam\build\tb6c

08/08/98  02:54p                561,153 info.zip
           1 File(s)                561,153 bytes
                               939,587,584 bytes free
Verifying contents of zip file

e:\teamc\testfam\build\tb6c>unzip -t info.zip
Archive:  info.zip
@(#) Package context: Family testfam, Release concurrent1, Workarea 10
  testing: ntlbdev1.sam           OK
  testing: ntlbdev2.sam           OK
  testing: ntlbdev3.sam           OK
  testing: ntlbdev4.sam           OK
  testing: ntlbdev5.sam           OK
  testing: ntlbdev6.sam           OK
  testing: ntlbdev7.sam           OK
  testing: trtcoem.pdf            OK
  testing: trtcvscm.pdf          OK
  testing: README.txt            OK
No errors detected in compressed data of info.zip.
Processing complete
***** End Of Build Output *****
```

(page 2 of 2)

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Build script to create self-extracting ZIP files

A build script can take an already created output as an input to produce another output. The **zip2exe** build script processes an input ZIP file that was previously built by the **zipbuild** build script. The zip2exe build script converts the ZIP file into a self-extracting ZIP file for Windows NT. This is useful for delivering files to a machine that does not already have the unzip.exe utility. Since the location of the unzipsfx.exe file is required by the zip2exe build script (since it could be in a different directory in a different system), the location of unzipsfx.exe is passed as a parameter of the build script:

Parameters: *d:\goodies\unzipsfx.exe*

Parameters value is set when creating the build script, or modifying the build script properties.

```
@echo off
REM $KW=@(#); $Chkd=1998/03/20 05:19:48; $FN=zip2exe.cmd; $Own=perlovl;
$Ver=98-03-16b:2;
REM $EKW;
echo TeamConnection zip file builder
echo -TC_BUILDERPARMS: %TC_BUILDERPARMS%
echo -TC_BUILD_USER: %TC_BUILD_USER%
echo -TC_FAMILY: %TC_FAMILY%
echo -TC_RELEASE: %TC_RELEASE%
echo -TC_LANG: %TC_LANG%
echo -TC_LOCATION: %TC_LOCATION%
echo -TC_INPUT: %TC_INPUT%
echo -TC_INPUTTYPE: %TC_INPUTTYPE%
echo -TC_OUTPUT: %TC_OUTPUT%
echo -TC_OUTPUTTYPE: %TC_OUTPUTTYPE%
echo -TC_WORKAREA: %TC_WORKAREA%
echo.
echo Build Start
@echo on
copy /b %TC_BUILDERPARMS%+%TC_INPUT% %TC_OUTPUT%
@if errorlevel 1 exit
@echo off
echo Verify contents of self-extracting zip file: %TC_OUTPUT%
@echo on
%TC_OUTPUT% -t
@if errorlevel 1 exit
@echo off
echo -----
dir %TC_OUTPUT%
echo Processing complete
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Null build script to simplify managing a build tree

A build script that does nothing can be very helpful when creating a build tree. A build tree is a collection of built parts, where inputs are built to create outputs. The built output parts are then used as inputs to another output, etc. Since a product can have hundreds or thousands of outputs, it is helpful for all of the built objects to be connected in a single build tree or several smaller build trees, depending on the number of connections and the resulting performance impact, so that only the top part(s) need be explicitly built. Since there is an input-output relationship to all of the other parts, anything that is out-of-date will be built as a result of building that top, collector part.

In many cases, it is easy to think of a good task for this collector part, such as distributing and installing the outputs to a set of user machines. We use a simple **null** build script to cause the build of all of the parts that are input to it. The output part is marked as a temporary file, so nothing is actually checked into VisualAge TeamConnection.

```
@echo off
REM $KW=@(#); $Chkd=1998/08/08 12:24:00; $FN=null.cmd; $Own=perlovl; $Ver=10:1;
REM $EKW;
echo VisualAge TeamConnection Version 3 NULL builder
echo -TC_BUILD_USER: %TC_BUILD_USER%
echo -TC_FAMILY: %TC_FAMILY%
echo -TC_RELEASE: %TC_RELEASE%
echo -TC_LANG: %TC_LANG%
echo -TC_LOCATION: %TC_LOCATION%
echo -TC_INPUT: %TC_INPUT%
echo -TC_INPUTTYPE: %TC_INPUTTYPE%
echo -TC_OUTPUT: %TC_OUTPUT%
echo -TC_OUTPUTTYPE: %TC_OUTPUTTYPE%
echo -TC_WORKAREA: %TC_WORKAREA%
echo.
echo Processing Complete
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Keywords in files

Keeping track of the version of each file inside the file is the best way to make sure you have the right version. This is particularly useful when a package, such as a ZIP file, is delivered to another system and installed. When problems arise it often becomes necessary to determine when and where a file was created (i.e. the context within the VisualAge TeamConnection family).

The tools used to locate keywords are **what** (available in Unix) and **tcwhat** (provided by VisualAge TeamConnection for Windows NT and OS/2) commands. These tools search text and binary files for a string that begins with **@(#)**. A line in a text file beginning with this symbol is called a *what string*. Note: The symbol used by *what* is referred to as the magic cookie for *what* in Unix. It might be comforting to know that the name is as odd as the string.

Since VisualAge TeamConnection versioning is based on the Workarea, the filename and version can be imbedded into text files using keywords. Since our class materials were all binary files, we could only use VisualAge TeamConnection keywords in our README.txt file and the heading of ZIP files built in the Release.

Keywords in README.txt file

```
README for Education class materials:  
Keywords:  
$KW=@(#); $ChkD=1998/08/08 18:36:22; $FN=README.txt; $Own=perlovl; $Ver=10:1;  
$EKW;
```

Output of using "tcwhat" on Windows NT

```
E:\TeamC\testfam>tcwhat readme.txt  
readme.txt:  
    ; $ChkD=1998/08/08 18:36:22; $FN=README.txt; $Own=perlovl; $Ver=10:1;
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Shadowing

Shadowing allows the contents of a Workarea, Driver or Release to be copied to a file system. The most common use is to shadow the committed version of the Release. Here are the commands that set up shadowing and force the first extract of parts.

Once this shadow is set up, all updates will be automatically performed when a Driver is committed.

It is useful to note that the shadow program shipped with VisualAge TeamConnection, TCshadow.c, can be modified to suit the needs of the development group. In this case, TCshadow.c has been modified to provide extra feedback each time it runs. So, each time a Driver is committed, information about all of the changes is returned as a text message.

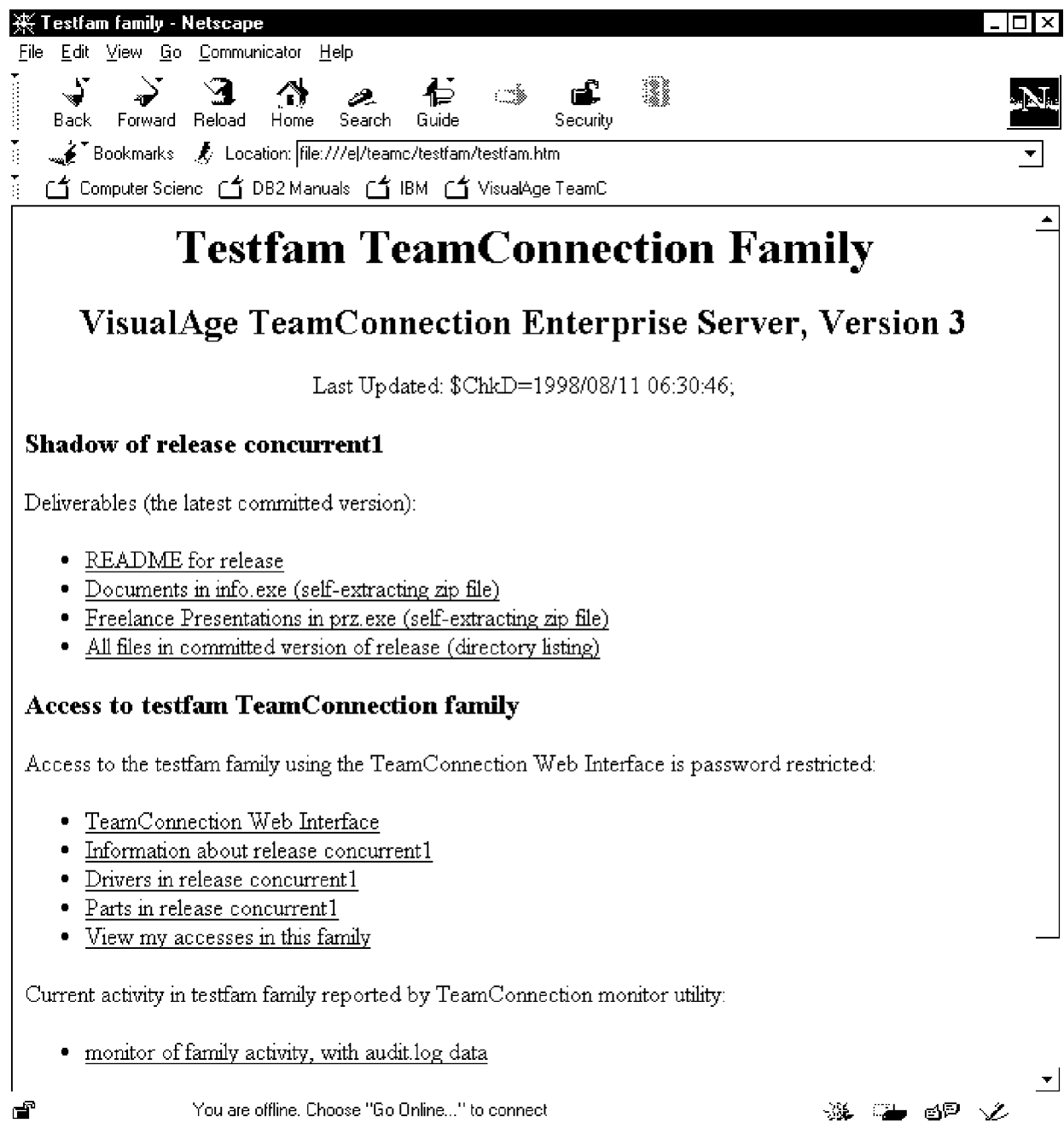
Actions used to create shadow

```
# Define a program to use by shadowing facility
teamc shadow -define shadow2 -family testfam -description "Shadow for Release using
TCshadow.c with debugging" -program e:\teamc\testfam\TCshadow.exe -verbose
# Create a specific shadow instance for shadowing the committed version of Release
# concurrent1 to a directory where the Web page can access the files
teamc shadow -create relshad1 -family testfam -release concurrent1 -type shadow2
-location e:\teamc\testfam\%N\%R\%P -contents release -synchronous -timestamp
preserve -priority 100 -verbose
# Activate shadowing for the committed Release
teamc shadow -enable relshad1 -family testfam -release concurrent1 -verbose
# Force the first copy of the shadowed files into the directory structure
# - All subsequent updates will occur during "teamc driver -commit"
teamc shadow -synchronize relshad1 -family testfam -release concurrent1 -force
-verbose
```


Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Accessing with the Web

Using a web page to provide customized access to the built output of a family makes access to the outputs easier to deliver. The sample web page used for testfam has several features to improve usability, coded in HTML.



Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

``

All files in committed version of Release (directory listing)

- Pointer to directory for browsing, using a **file**

``

README for Release

- Pointer to selected files that users might want to display, again using a **file** reference

``

Documents in info.exe (self-extracting zip file)

- Pointer to selected files that users might want to download, again using a **file** reference

`<base href="http://localhost:9001">`

`TeamConnection Web Interface`

- Pointers to the VisualAge TeamConnection web interface, main page
- All references to the testfam family begin with **http://localhost:9001**. Using a base href allows for less duplication of data within a web page, allowing each access to the Web Interface to start with only a slash, /. To access the main page needs only the one slash.

``

Information about Release concurrent1

- Pointers to specific queries within the VisualAge TeamConnection that might be useful to a user of the family. Since the syntax for accessing specific queries is documented, this is a helpful shortcut when getting data from a VisualAge TeamConnection family.
- Generating queries like this are relatively easy. Use the VisualAge TeamConnection web interface to generate the desired query or action, then go to the **History** dialog of the web browser and copy the html for that query from the history dialog into the source of a customized web page.

`<a`

`href="/report/ACCESSVIEW?USERLOGIN=TC_USER&order1=COMPNAME&cols=COMPNAME,USERLOGIN,USERNAME,USERAREA,AUTHORITYNAME,AUTHORITYTYPE">`

View my accesses in this family

- Pointer to a query that passes the userID used when logging onto VisualAge TeamConnection in order to create a personalized query through the VisualAge TeamConnection web interface
- Each column to be displayed and their order can be specified in the report.

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

```
<a href="file:///e:/teamc/testfam/monitor.htm">
```

```
monitor of family activity, with audit.log data</a>
```

- Pointer to a monitor web page that shows the actions being executed by the VisualAge TeamConnection family server
- Note: The source for this example ran on a single machine without a web server. In the production environment, the **file:///** has been replaced with an **http://**.

```
<a href="http://software.ibm.com/ad/teamcon">
```

```
TeamConnection Marketing Home Page</a>
```

- Pointers to other relevant web pages

```
<!--$KW=@ (#) ;>
```

```
<P>Last Updated: $ChkD=1998/08/11 06:30:46;</P></CENTER>
```

```
<!--$EKW;>
```

- A VisualAge TeamConnection keyword that is updated after this web page is checked in, then extracted. The extraction is performed by the shadowing facility when the Workarea containing the updated web page is committed into the Release.
- Comments, `<!-- ...>`, are used to hide the begin keyword translation and end keyword translation keywords.

```
<H3>Shadow of Release concurrent1</H3>
```

```
<P>Deliverables (the latest committed version):</P>
```

```
<MENU>
```

```
<LI>...</LI>
```

```
<LI>...</LI>
```

```
</MENU>
```

- Menus allow grouping of similar functions, such as a download section, a query section and a general information section.
- Headings and paragraphs allow for descriptions of each menu.

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Source for Web Page

```
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
  <META NAME="GENERATOR" CONTENT="Mozilla/4.05 [en] (Win95; U) [Netscape]">
  <base href="http://localhost:9001">
  <TITLE>Testfam family</TITLE>
</HEAD>
<BODY>

<CENTER>
<H1>Testfam TeamConnection Family</H2>
<H2>VisualAge TeamConnection Enterprise Server, Version 3</H2>
<!--$KW=@{#};>
<P>Last Updated: $ChkD=1998/08/11 06:30:46;</P></CENTER>
<!--$EKW;>
<H3>Shadow of Release concurrent1</H3>
<P>Deliverables (the latest committed version):</P>
<MENU>
<LI><a href="file:///e:/teamc/testfam/concurrent1/concurrent1/README.txt">README for
Release</a></LI>
<LI><a href="file:///e:/teamc/testfam/concurrent1/concurrent1/info.exe">Documents in info.exe
(self-extracting zip file)</a></LI>
<LI><a href="file:///e:/teamc/testfam/concurrent1/concurrent1/info.exe">Freelance Presentations
in prz.exe (self-extracting zip file)</a></LI>
<LI><a href="file:///e:/teamc/testfam/concurrent1/concurrent1">All files in committed version of
Release (directory listing)</a></LI>
</MENU>
<!-->
<H3>Access to testfam TeamConnection family</H3>
<P>Access to the testfam family using the TeamConnection Web Interface is password
restricted:</P>
<MENU>
<LI><a href="/">TeamConnection Web Interface</a></LI>
<LI><a href="/report/RELEASEVIEW?where=NAME%3D'concurrent1'">Information about Release
concurrent1</a></LI>
<LI><a
href="/report/DRIVERVIEW?RELEASENAME=concurrent1&order1=NAME&cols=NAME,RELEASENAME,USERLOGIN,USER
NAME,USERAREA,ADDDATE,COMMITDATE,LASTUPDATE,STATE,DRIVERTYPE">Drivers in Release
concurrent1</a></LI>
<LI><a
href="/report/PARTVIEW?CONTEXTNAME=concurrent1&CONTEXTRELEASE=concurrent1&order1=PATHNAME&cols=CO
NTEXTNAME,CONTEXTRELEASE,ADDDATE,DROPPATE,PATHNAME,TRANSLATION,RESTRICTED,COMPNAME,COMMITTEDVERSI
ON,USERLOGIN,PARTTYPE,WORKAREANAME,PARSERNAME,CURRENTVERSION,CHANGETYPE,BUILDSTATUS,WORKAREACHANG
E,TRANSSTATE,EMODE,FILETYPE,NUADDDATE,NUDROPPATE,LASTUPDATE,BASENAME,NUPATHNAME,TEMPORARY,PARENT-
NAME">Parts in Release concurrent1</a></LI>
<LI><a
href="/report/ACCESSVIEW?USERLOGIN=$TC_USER$&order1=COMPNAME&cols=COMPNAME,USERLOGIN,USERNAME,USE
RAREA,AUTHORITYNAME,AUTHORITYTYPE">View my accesses in this family</a></LI>
</MENU>
<P>Current activity in testfam family reported by TeamConnection monitor utility:</P>
<MENU>
<LI><a href="file:///e:/teamc/testfam/monitor.htm">monitor of family activity, with audit.log
data</a></LI>
</MENU>
<HR>
<H3>More information on VisualAge TeamConnection Enterprise Server Version 3</H3>
<a href="http://software.ibm.com/ad/teamcon">TeamConnection Marketing Home Page</a>
</BODY>
</HTML>
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Securing data

The data we manage in VisualAge TeamConnection is important to us. Therefore, we consider security whenever appropriate. In order to ensure security:

- The TeamConnection Web Interface requires password access to the VisualAge TeamConnection family, either PASSWORD_ONLY or PASSWORD_OR_HOST
- Any web page that accesses data from the family, such as a shadow, is password protected. This is a standard feature of most Web servers.

Other advantages

Initially, we created separate Releases for our class materials, web pages, tools, etc. However, once the delivery of these parts became closely linked, for example the web page used to extract parts changes when the list of parts delivered changes, it became obvious that most of the parts should be managed in one Release. When adding output parts, it was possible to extract the built output files instead of extracting all of the parts in each Release. After adding the web page and shadowing, it was no longer necessary to extract the Release at all in order to deliver.

The biggest advantage of using only one Release is that only one Workarea is needed for each defect or feature, and there is no need to specify *common Releases* when checking in parts. This eliminates the risk of accidentally breaking links, resulting in out-of-date parts in some Releases.

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Using a web page to monitor family server activity

Web pages on the family server can provide information about family activity. The customized VisualAge TeamConnection web page points to this **monitor** page to monitor the commands being executed in the family server. While this is not directly related to building our product, it is useful information for developers and family administrators monitoring build activity.

The C program, `htm_mon.c`, generates the monitor web page, `monitor.htm`. A few HTML features make this web page more effective:

```
<META HTTP-EQUIV="Refresh" content="5">
```

- The **htm_mon.exe** program queries the family activity by executing the VisualAge TeamConnection **monitor.exe** program at fixed intervals, 5 seconds in this case. The web page automatically refreshes at the same interval. This reduces the likelihood that the page will be in the process of writing when the refresh occurs.

```
<pre>  
</pre>
```

- To simplify HTML coding, the output of the **monitor.exe** program is written into the web page. Using the **monitor 0** syntax generates plain text in a readable format that doesn't need to be formatted by HTML.

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

The screenshot shows a Netscape browser window titled "Activity of VisualAge TeamConnection Family - Netscape". The address bar contains "file:///e:/teamc/testfam/monitor.htm". The page content includes a title "Monitoring family: testfam" and a timestamp "As of: Wed Sep 9 23:00:54 1998". The main text displays system status: "shmName: SHAREMEM_testfam", "2 of 2 teamcd daemons running.", and "Total number of hits: 8". It lists two hits with IDs and counts. A section titled "Last 10 lines of audit.log:." contains a warning about password attempts. Below this is a log of four successful events: TcLoginLogin, ReleaseView, and two Report events. The page ends with "End of output". The browser's status bar at the bottom indicates "You are offline. Choose 'Go Online...' to connect".

Activity of VisualAge TeamConnection Family - Netscape

File Edit View Go Communicator Help

Back Reload Home Search Guide Security

Bookmarks Location: file:///e:/teamc/testfam/monitor.htm

Computer Scienc DB2 Manuals IBM VisualAge TeamC

Monitoring family: testfam

As of: Wed Sep 9 23:00:54 1998

```
shmName: SHAREMEM_testfam
2 of 2 teamcd daemons running.
Press Cont-C to quit.
Total number of hits: 8
01,00225,00005,,
02,00234,00003,,
```

Last 10 lines of audit.log:.

```
      If the user continues to try to use tclogin with an invalid password
      and the limit of maximum failed attempts is reached, then the account
      for that user in the family will be disabled. Then, only a superuser
      can reactivate the account.
```

```
00225,TcLoginLogin,SUCCESS,1998/09/09,22:55:40,22:55:40,bldadmin,perlov1,localhostper
00234,ReleaseView,SUCCESS,1998/09/09,22:55:28,22:56:00,perlov1,perlov1,localhostperlo
00225,Report,SUCCESS,1998/09/09,22:56:06,22:56:07,perlov1,perlov1,localhostperlovtr.r
00225,Report,SUCCESS,1998/09/09,22:56:32,22:56:58,perlov1,perlov1,localhostperlovtr.r
```

End of output

You are offline. Choose "Go Online..." to connect

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Output of htm_mon.exe: monitor.htm

```
<META HTTP-EQUIV="Refresh" content="5">
<!doctype html public "html2.0">
<html>
<title>
Activity of VisualAge TeamConnection Family
</title>
<body>
<h1>Monitoring family: testfam</h1>
<h2>As of: Wed Sep 9 23:00:54 1998
</h2>
<pre>
shmName: SHAREMEM_testfam
2 of 2 teamcd daemons running.
Press Cont-C to quit.
Total number of hits: 8
01,00225,00005,,
02,00234,00003,,

Last 10 lines of audit.log:.
    If the user continues to try to use tclogin with an invalid
password
    and the limit of maximum failed attempts is reached, then
the account
    for that user in the family will be disabled. Then, only a
superuser
    can reactivate the account.

00225,TcLoginLogin,SUCCESS,1998/09/09,22:55:40,22:55:40,bldadmin,perl
ovl,localhostperlovtr.raleigh.ibm.com,perlovl
00234,ReleaseView,SUCCESS,1998/09/09,22:55:28,22:56:00,perlovl,perlov
l,localhostperlovtr.raleigh.ibm.com,ConfigInfo
00225,Report,SUCCESS,1998/09/09,22:56:06,22:56:07,perlovl,perlovl,loc
alhostperlovtr.raleigh.ibm.com,ReleaseView,1=1
00225,Report,SUCCESS,1998/09/09,22:56:32,22:56:58,perlovl,perlovl,loc
alhostperlovtr.raleigh.ibm.com,PartFullView,releaseName in ('concur-
rent1') order by nuPathName ,releaseName ,compName
</pre>
<h2>End of output</h2>
</body>
</html>
```


Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

Source Code for htm_mon.c

```
/*
*****
SAMPLE NAME:  htm_mon.c

USAGE:        htm_mon MONFILE FREQUENCY [LONG]

COMPILATION:  cc -D__UNIX__ -o htm_mon htm_mon.c
              gcc -D__NT__ htm_mon.c
              gcc -D__OS2__ htm_mon.c

ENVIRONMENT VARIABLES:
                TC_FAMILY, TC_DBPATH

DESCRIPTION:   This sample C program runs the monitor command and
              outputs an HTML file that can be displayed by a web
              server

*****
*             IBM VisualAge TeamConnection
*             Version 3 Release 0
*
*             5622-717
*             (C) Copyright, IBM Corp., 1998. All Rights Reserved.
*             Licensed Materials - Property of IBM
*
*             US Government Users Restricted Rights
*             - Use, duplication or disclosure restricted by
*             GSA ADP Schedule Contract with IBM Corp.
*
*             IBM is a registered trademark of
*             International Business Machines Corporation
*****
*
*             NOTICE TO USERS OF THE SOURCE CODE EXAMPLES
*
*             INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THE SOURCE CODE
*             EXAMPLES, BOTH INDIVIDUALLY AND AS ONE OR MORE GROUPS, "AS IS" WITHOUT
*             WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT
*             LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
*             PARTICULAR PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE
*             OF THE SOURCE CODE EXAMPLES, BOTH INDIVIDUALLY AND AS ONE OR MORE GROUPS,
*             IS WITH YOU.  SHOULD ANY PART OF THE SOURCE CODE EXAMPLES PROVE
*             DEFECTIVE, YOU (AND NOT IBM OR AN AUTHORIZED DEALER) ASSUME THE ENTIRE
*             COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
*
*****
*/

#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#if defined(__OS2__)
    #define INCL_DOS
    #include <os2.h>
#endif
#if defined(__NT__)
    #include <windows.h>
#endif
#endif
```

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

```
#define Long_True 1
#define Long_False 0

extern int errno;

/*-----*/
|  monHelp:
|  - Explain usage of htm_mon sample program
|  - Write output to standard error
|-----*/
void monHelp(void)
{
    fprintf(stderr, "htm_mon usage:\n");
    fprintf(stderr, "\thtm_mon                - This help message\n");
    fprintf(stderr, "\thtm_mon MONFILE FREQUENCY [LONG] - Update htm file with monitor
output.\n");
    fprintf(stderr, "htm_mon notes:\n");
    fprintf(stderr, "\tMONFILE will be overwritten during each update.\n");
    fprintf(stderr, "\tFREQUENCY must be an integer between 1 and 60.\n");
    fprintf(stderr, "\tLONG, if provided is the number of lines in the audit.log to be
included.\n");
    fprintf(stderr, "Direct users to point to the MONFILE in the designated directory\n");
    fprintf(stderr, "on your web server (same as family server), or create a web page\n");
    fprintf(stderr, "to perform that function for your users.\n");
    return;
}

main(int argc, char *argv[])
{
    FILE *tmpFile; /* Write to temporary file first */
    char *tmpFileName;
    int monInterval, auditInterval;
    char *famName;
    char cmdBuffer[1000];
    struct tm *newtime;
    struct tm *savetime;
    time_t ltime;
    int endSleep;
    int Long = Long_False;

    /* Check for correct count in parameter list */
    if (argc < 3)
    {
        monHelp();
        exit(1);
    }

    /* Get and verify FREQUENCY */
    if (argv[2] != NULL)
    {
        monInterval = 0;
        monInterval = atoi(argv[2]);
        if (monInterval < 1 || monInterval > 60)
        {
            fprintf(stderr, "htm_mon Error: FREQUENCY must be an integer > 0 and <= 60\n.");
            monHelp();
            exit(1);
        }
    }
    else
    {

```

(page 2 of 4)

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

```
fprintf(stderr,"htm_mon Error: FREQUENCY must be valid string containing an integer.\n.");
monHelp();
exit(1);
}

/* Get LONG parameter */
if (argv[3] != NULL)
{
    Long = Long_False;
    auditInterval = 0;
    auditInterval = atoi(argv[3]);
    if (auditInterval < 1)
    {
        fprintf(stderr,"htm_mon Error: LONG must be an integer > 0\n.");
        monHelp();
        exit(1);
    }
    else
    {
        Long = Long_True;
    }
}

/* Get environment variable value for TC_FAMILY */
famName = getenv("TC_FAMILY");
if (famName == NULL)
{
    fprintf(stderr,"htm_mon Error: TC_FAMILY environment variable not set.\n.");
    monHelp();
    exit(1);
}

/* Get filename and open file */
if (argv[1] == NULL)
{
    fprintf(stderr,"htm_mon Error: MONFILE must be valid string\n.");
    monHelp();
    exit(1);
}

/* Loop endlessly */
while ((void *)1)
{
    /* Open temporary monitor output file */
    tmpFileName = tmpnam(NULL);
    tmpFile = fopen(tmpFileName, "wb");

    if (tmpFile == (FILE *)NULL)
    {
        fprintf(stderr,"htm_mon Error: could not open file: %s\n.", argv[2]);
        exit(1);
    }

    fprintf(tmpFile,"<META HTTP-EQUIV=\"Refresh\" content=\"%d\">\n", monInterval);
    fprintf(tmpFile,"<!doctype html public \"html2.0\">\n");
    fprintf(tmpFile,"<html>\n");
    fprintf(tmpFile,"<title>\n");
    fprintf(tmpFile,"Activity of VisualAge TeamConnection Family\n");
    fprintf(tmpFile,"</title>\n");
    fprintf(tmpFile,"<body>\n");

    fprintf(tmpFile,"<h1>Monitoring family: %s</h1>\n", famName);
}
```

(page 3 of 4)

Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and Distribute

```
time(&lttime);
newtime = localtime(&lttime);
fprintf(tmpFile, "<h2>As of: %s</h2>\n", asctime(newtime));
fprintf(tmpFile, "<pre>\n");

fflush(tmpFile);
fclose(tmpFile);

/* Run monitor Command */
sprintf(cmdBuffer, "monitor 0 >> %s 2>&1", tmpFileName);
system(cmdBuffer);

/* Append audit log data */
if (Long == Long_True)
{
    /* Run monitor Command */
#ifdef __UNIX__
    sprintf(cmdBuffer, "print >> %s 2>&1", tmpFileName);
    system(cmdBuffer);
    sprintf(cmdBuffer, "print Last %d lines of audit.log:. >> %s 2>&1", auditInterval,
tmpFileName);
    system(cmdBuffer);
    sprintf(cmdBuffer, "tail -10 %s/audit.log >> %s 2>&1", auditInterval,
getenv("TC_DBPATH"), tmpFileName);
    system(cmdBuffer);
#else
    sprintf(cmdBuffer, "@echo. >> %s 2>&1", tmpFileName);
    system(cmdBuffer);
    sprintf(cmdBuffer, "@echo Last %d lines of audit.log:. >> %s 2>&1", auditInterval,
tmpFileName);
    system(cmdBuffer);
    sprintf(cmdBuffer, "@tail -%d %s\\audit.log >> %s 2>&1", auditInterval,
getenv("TC_DBPATH"), tmpFileName);
    system(cmdBuffer);
#endif
}

/* Reopen file and finish writing html */
tmpFile = fopen(tmpFileName, "ab");
fprintf(tmpFile, "</pre>\n");
fprintf(tmpFile, "<h2>End of output</h2>\n");
fprintf(tmpFile, "</body>\n");
fprintf(tmpFile, "</html>\n");
fflush(tmpFile);
fclose(tmpFile);
printf("Finished updating %s at %s\n", tmpFileName, asctime(newtime));
/* Replace current file */
remove(argv[1]);
rename(tmpFileName, argv[1]);

#ifdef __OS2__
    DosSleep(monInterval*1000);
#endif
#ifdef __NT__
    Sleep(monInterval*1000);
#endif
#ifdef __UNIX__
    sleep(monInterval);
#endif
}
exit(0);
}
```

(page 4 of 4)

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Final Thoughts

The VisualAge TeamConnection build facility is a general purpose facility. That means you can automate any process that is associated with the parts in your Releases. We found that using build in conjunction with the VisualAge TeamConnection shadow facility and the VisualAge TeamConnection web interface allowed us to create a customized web based solution that is applicable for many development efforts. It has allowed us to create a more automated, more efficient and more pleasant development process.

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Bibliography

VisualAge TeamConnection Publications

For more information on how to use VisualAge TeamConnection, you can consult the following manuals:

SC34-4551 TeamConnection, Administrator's Guide
SC34-4552 Getting Started with the TeamConnection Clients
SC34-4499 TeamConnection, User's Guide
SC34-4501 TeamConnection, Commands Reference
SC34-4500 TeamConnection, Quick Commands Reference

Related Redbooks

The following IBM redbooks provide practical advice about VisualAge TeamConnection from software specialists:

SG24-4648 Introduction to the IBM Application Development
Team Suite
SG26-2008 TeamConnection Family and Application Development
SG24-4610 TeamConnection Workframe Integration Survival Guide

Related Technical Reports

The following technical reports provide hints for using VisualAge TeamConnection:

29.2357 Evolution of a new TeamConnection Family:
Taking advantage of automation (second in a series)
29.2267 TeamConnection frequently asked questions: How to do
routine operating system tasks
29.???? Comparison between CMVC 2.3.1 and VisualAge
TeamConnection Enterprise Server 3

Evolution of a VisualAge TeamConnection family:
Using the Web and Shadowing to Build and Distribute

Copyrights, Trademarks and Service marks

The following terms used in this technical report, are trademarks or service marks of the indicated companies:

TRADEMARK, REGISTERED TRADEMARK OR SERVICE MARK	COMPANY
AIX, OS/2, IBM, DB2, CMVC, VisualAge TeamConnection, TeamConnection	IBM Corporation
PDF	Adobe Systems Incorporated
Intel	Intel Corporation
UNIX	UNIX System Laboratories, Inc.
Microsoft, Windows, Windows NT	Microsoft Corporation
Netscape	Netscape Communications Corp.
Info-ZIP	Info-ZIP Group