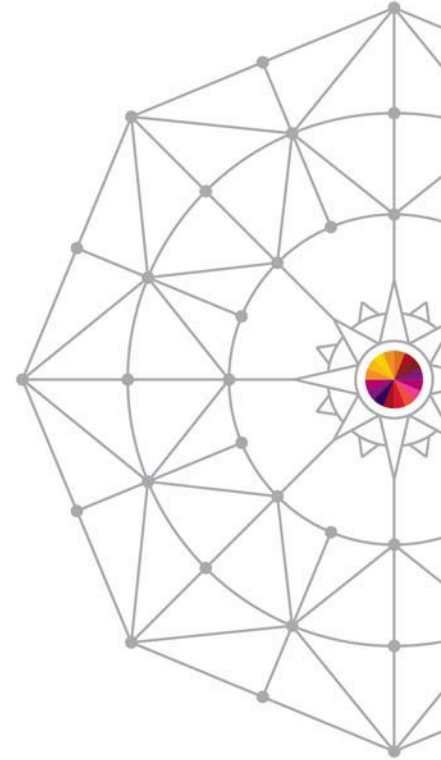


# Where Copybooks Go and Rational Developer for System z and Rational Team Concert Implementation Questions

Venkat Balabhadrapatruni  
*venkatu@us.ibm.com*

August 6<sup>th</sup>, 2014  
Session: 15474




#SHAREorg



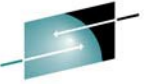
SHARE is an independent volunteer-run information technology association  
that provides **education, professional networking and industry influence.**

Copyright (c) 2014 by SHARE Inc.

 Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

 Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

# Please note



SHARE  
in Pittsburgh 2014

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.



# Abstract

Consider that you have a bunch of existing source code on IBM z/OS. But, how do you organize it in Rational Developer for System z projects and Rational Team Composer components, team areas, and project areas so that you get the most from each tool? Where should you put your COPY files? How about all those COBOL source files? And, what about the various other configuration, setup, and source file types that comprise your application? In this session, you learn about the project types, component types and elements in Rational Developer for System z and Rational Team Composer. The session starts with a simple application and progresses to more complex constructions.

# Why are we here?

- Everybody has lots of code to manage
  - struggle with the best way to do so
  - Code may be scattered in many places
  - Or consolidated in a small number of places
- Code structure impacts delivery speed
  - Education and learning
  - Build speed
  - Test speed
- Learn how to approach a source base and bring structure to it
  - Identify starting points
  - Use these to suggest groupings
  - Adjust based on team knowledge and experience

# Agenda

- Definitions
- Motivation for identifying component structure
- High Level approach
- A small example
- Scaling up
- Using analysis tools to help
- Additional resources

# Definitions

- Stream
  - A repository object in RTC that contains one or more components – these collect team's work
- Component
  - A grouping of related artifacts (files) in a stream.
- zComponent Project
  - A container into which buildable artifacts can be placed
- zFolder
  - A container to group artifacts within a zComponent Project – aligns with PDS type of dataset
- zFile
  - An artifact that represents a PDS member in z/OS

# Motivation for structure

- Structure matters
  - Helps organize modules
  - Helps team members learn about the source base
  - Helps administrators understand what is associated with what
  - Helps define the bounds of a set of programs
- What if there is no apparent structure?
  - Use some analysis tools to suggest a structure to follow
  - Break up into smaller sets to simplify handling
  - Ease the processing of long lists of file names

# High Level Approach

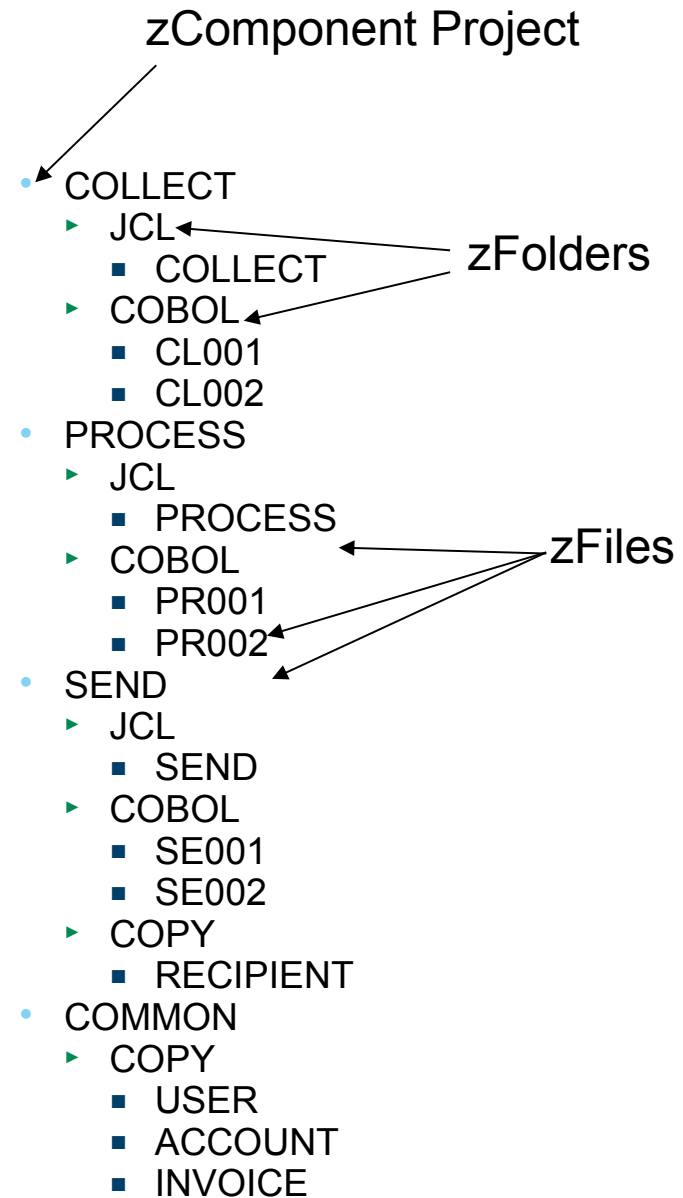
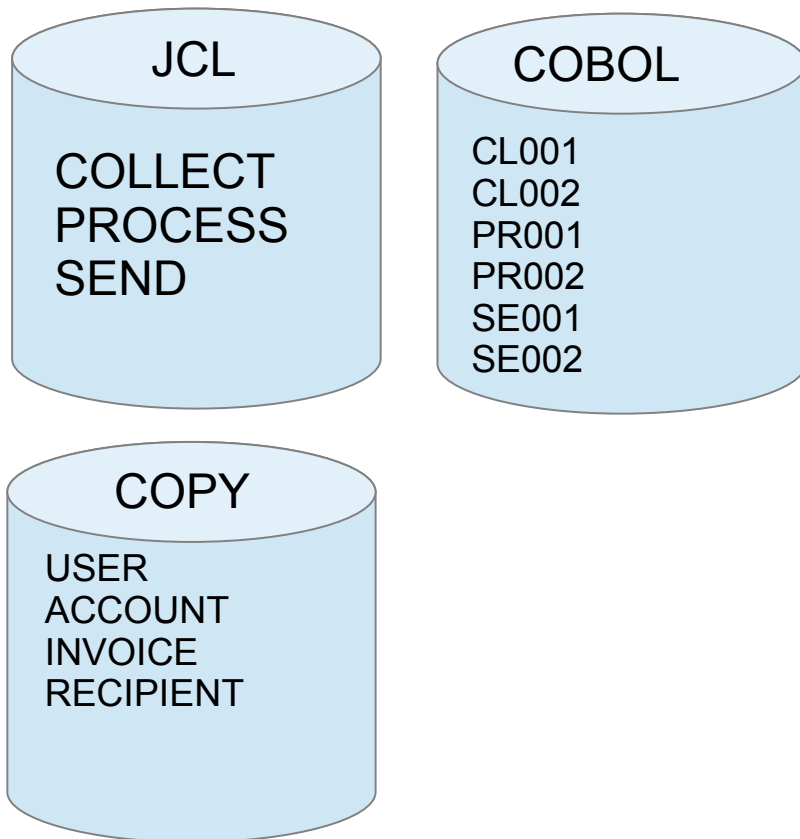
- Start with known entry points into your programs
  - Batch jobs
  - CICS transactions
  - IMS transactions ... and so on
- Analyze the programs which these reference, including COPY files, called programs, all included screen definitions, and so forth
  - Identify this as a group
- Identify files which are included in more than one group
  - Extract from the base groups those files which are common across groups into a separate group



# High Level Approach

- Repeat this process until you have a groups which cover the set of files analyzed and do not have any common files
- Re-combine the subset/extracted groups based on knowledge and experience to minimize the number of groups
- Create zComponent Projects for each of the groups
- Create zFolders in each zComponent Project for each file type
- Create zFiles in the appropriate zFolder for each file in the group

# A Small Example



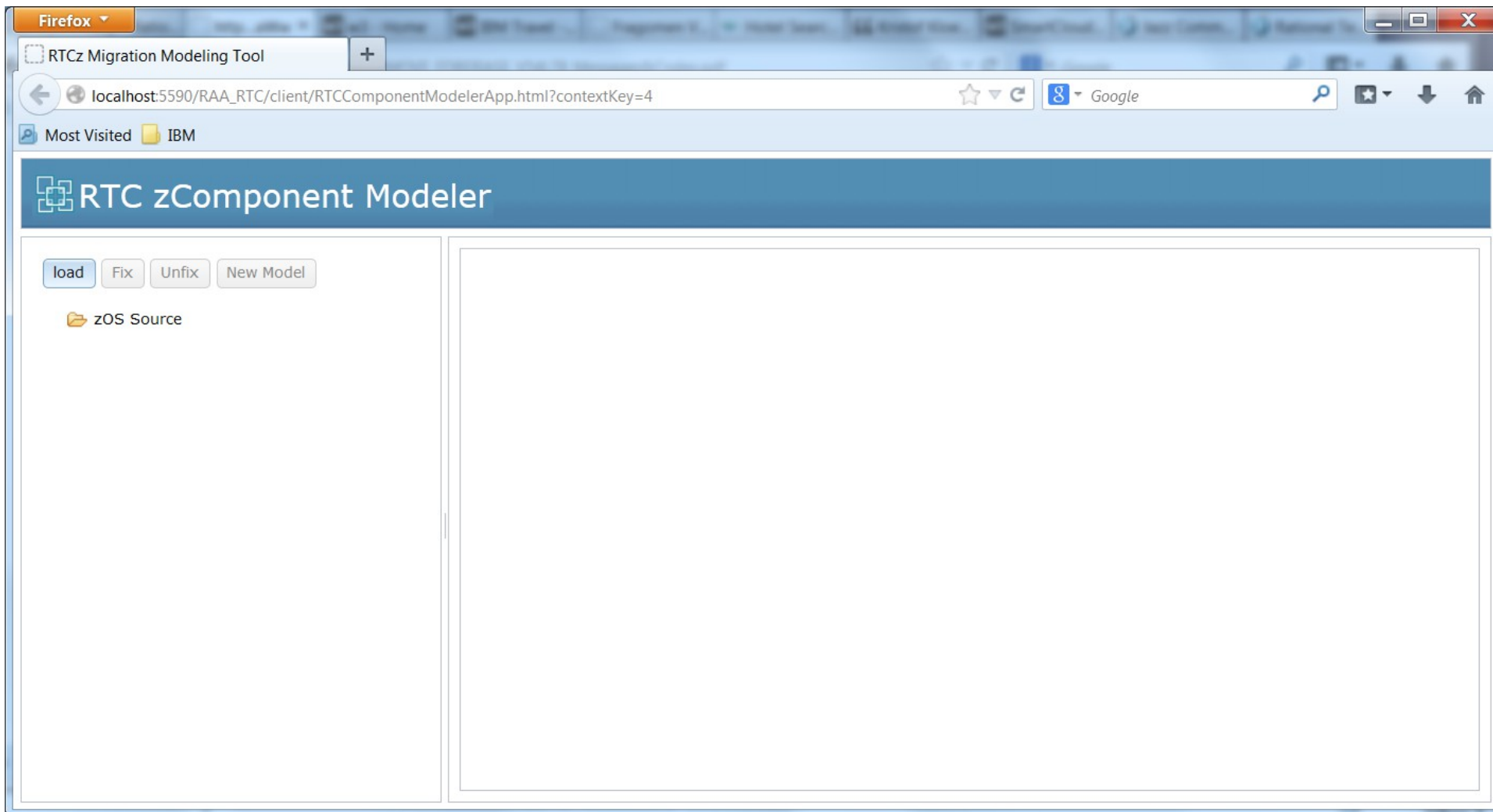
# Scaling Up

- This looks easy with a small example
- For thousands of files in a small set of datasets this can be very difficult
- The number of sub-sets generated can get very large, very quickly
  - Not collapsing down results in too many components
- Sometimes it is better to re-combine subsets just to cut down on the number of distinct sub-sets are defined – within reason
  - Collapsing too far results in components that are too large

# Using Analysis Tools

- Rational Asset Analyzer
  - Graphical views of program structure
  - Reports and queries to identify included and referenced files
- Rational Developer for System z
  - Graphical views of program structure
  - Code navigation to navigate to referenced files
  - Dataset search filters to guide common collections

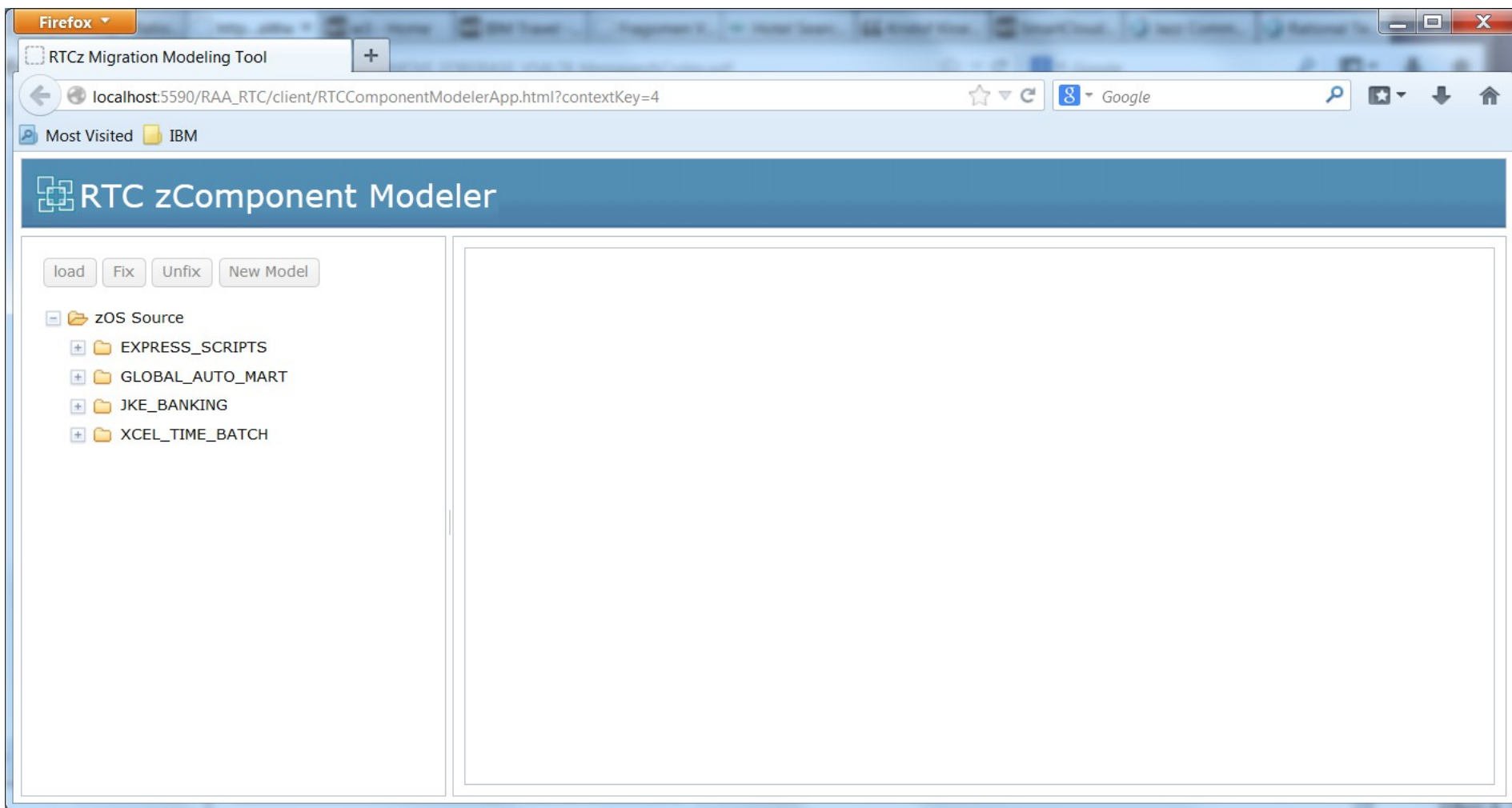
# Going a step further ...



The screenshot shows a Firefox browser window with the following details:

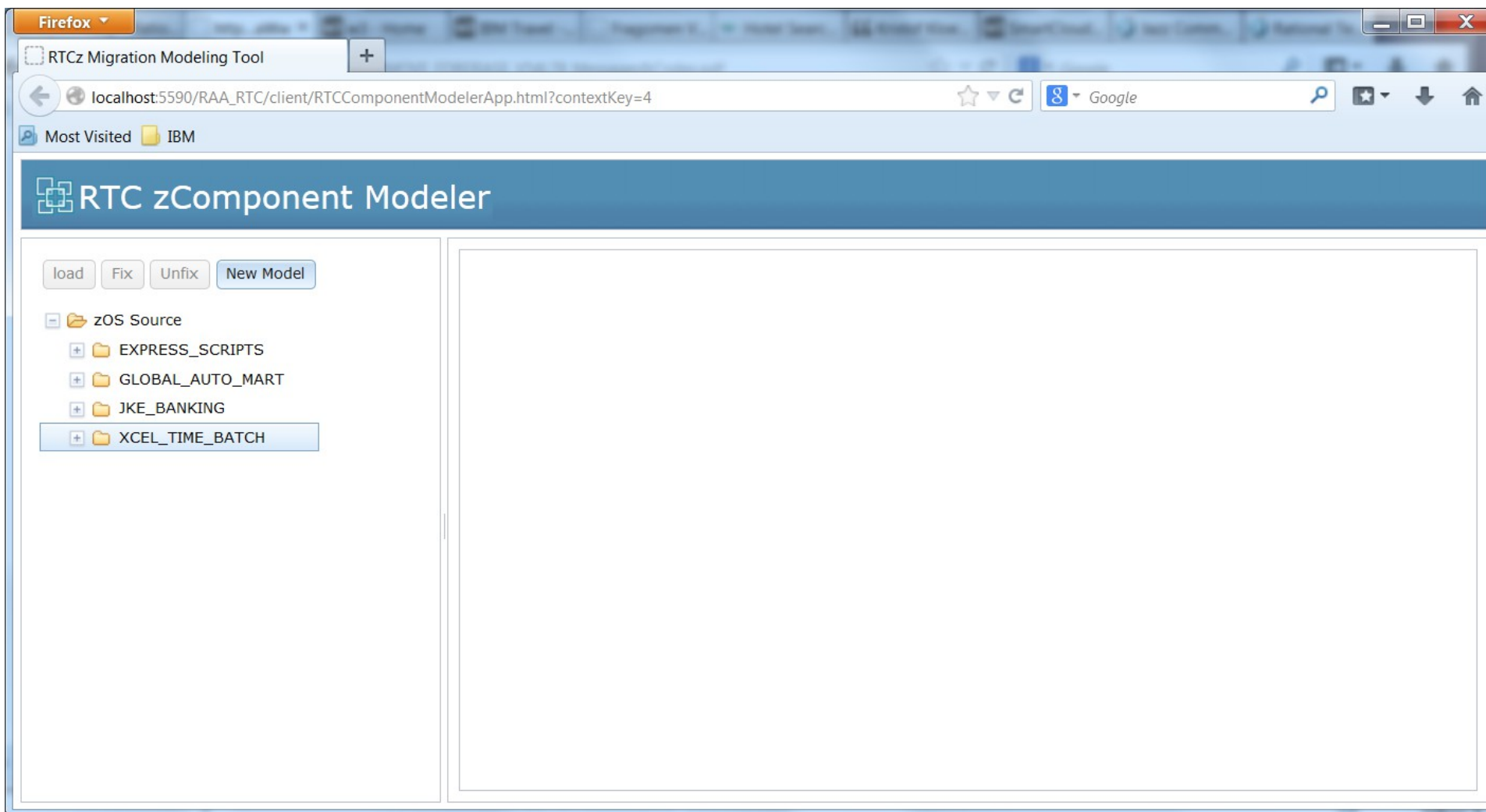
- Browser Tab:** RTCz Migration Modeling Tool
- Address Bar:** localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4
- Page Title:** RTC zComponent Modeler
- Left Panel:** Contains buttons for 'load', 'Fix', 'Unfix', and 'New Model'. Below these buttons is a folder icon labeled 'zOS Source'.
- Main Content Area:** A large, empty white rectangular area.

# Going a step further ...



The screenshot shows a Firefox browser window displaying the RTC zComponent Modeler application. The address bar shows the URL: localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4. The application interface includes a header with the title "RTC zComponent Modeler" and a sidebar with a file tree structure. The file tree shows a "zOS Source" folder containing subfolders: EXPRESS\_SCRIPTS, GLOBAL\_AUTO\_MART, JKE\_BANKING, and XCEL\_TIME\_BATCH. The main content area is currently empty.

# Going a step further ...



Firefox

RTCz Migration Modeling Tool

localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4

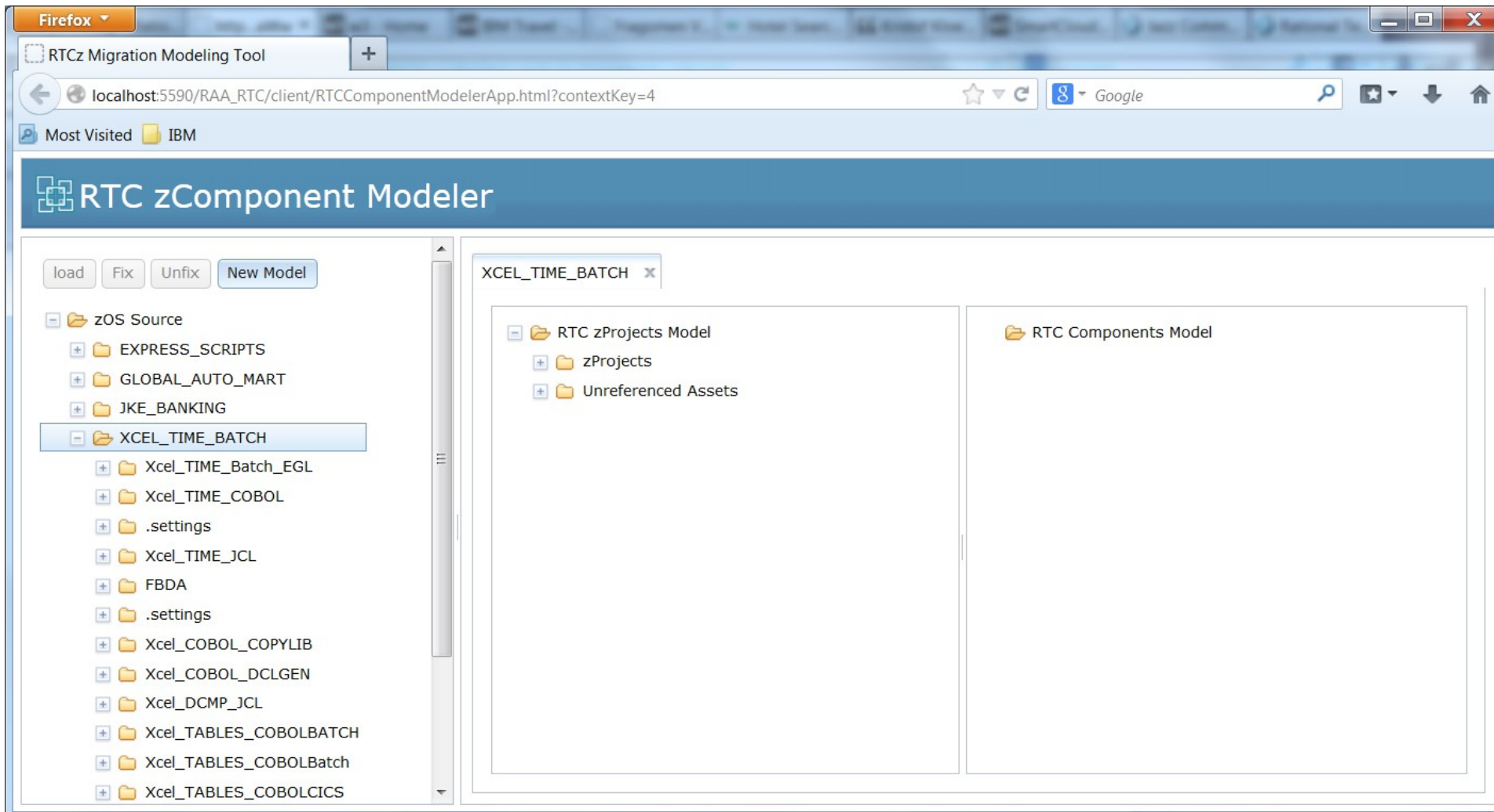
Most Visited IBM

## RTC zComponent Modeler

load Fix Unfix New Model

- zOS Source
  - EXPRESS\_SCRIPTS
  - GLOBAL\_AUTO\_MART
  - JKE\_BANKING
  - XCEL\_TIME\_BATCH

# Going a step further ...



The screenshot shows a Firefox browser window displaying the RTC zComponent Modeler application. The browser's address bar shows the URL `localhost:5590/RAA_RTC/client/RTCComponentModelerApp.html?contextKey=4`. The application interface features a top navigation bar with the title "RTC zComponent Modeler". Below this, there is a toolbar with buttons for "load", "Fix", "Unfix", and "New Model". The main workspace is divided into two panels. The left panel shows a tree view of the project structure, with "XCEL\_TIME\_BATCH" selected. The right panel shows the details of the selected project, including "RTC zProjects Model" and "RTC Components Model".

Firefox

RTCz Migration Modeling Tool

localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4

Most Visited IBM

## RTC zComponent Modeler

load Fix Unfix New Model

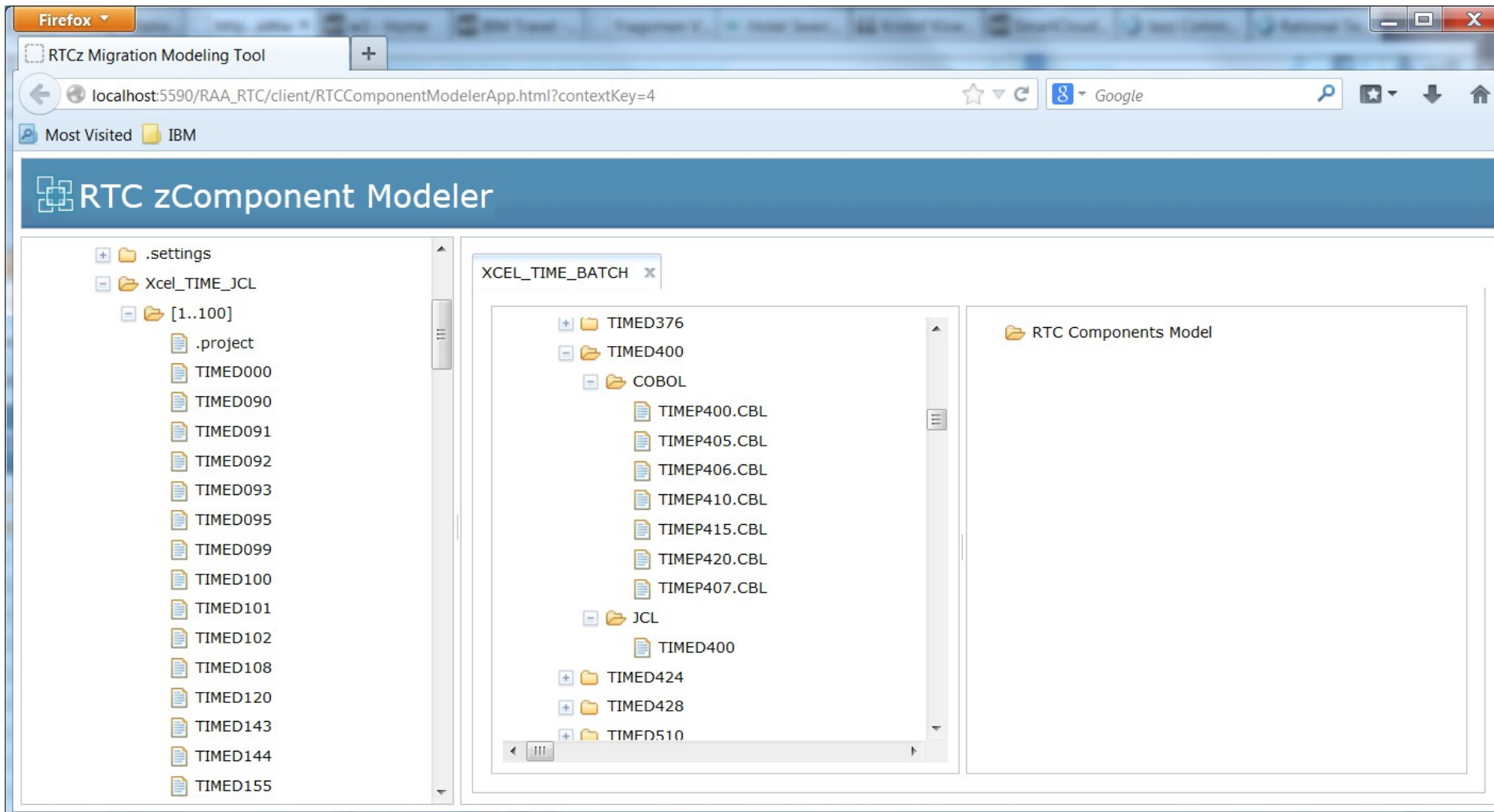
- [-] ZOS Source
  - [+] EXPRESS\_SCRIPTS
  - [+] GLOBAL\_AUTO\_MART
  - [+] JKE\_BANKING
  - [+] XCEL\_TIME\_BATCH
    - [+] Xcel\_TIME\_Batch\_EGL
    - [+] Xcel\_TIME\_COBOL
    - [+] .settings
    - [+] Xcel\_TIME\_JCL
    - [+] FBDA
    - [+] .settings
    - [+] Xcel\_COBOL\_COPYLIB
    - [+] Xcel\_COBOL\_DCLGEN
    - [+] Xcel\_DCMP\_JCL
    - [+] Xcel\_TABLES\_COBOLBATCH
    - [+] Xcel\_TABLES\_COBOLBatch
    - [+] Xcel\_TABLES\_COBOLCICS

XCEL\_TIME\_BATCH

- [-] RTC zProjects Model
  - [+] zProjects
  - [+] Unreferenced Assets
- [-] RTC Components Model



# Going a step further ...



Firefox

RTCz Migration Modeling Tool

localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4

Most Visited IBM

## RTC zComponent Modeler

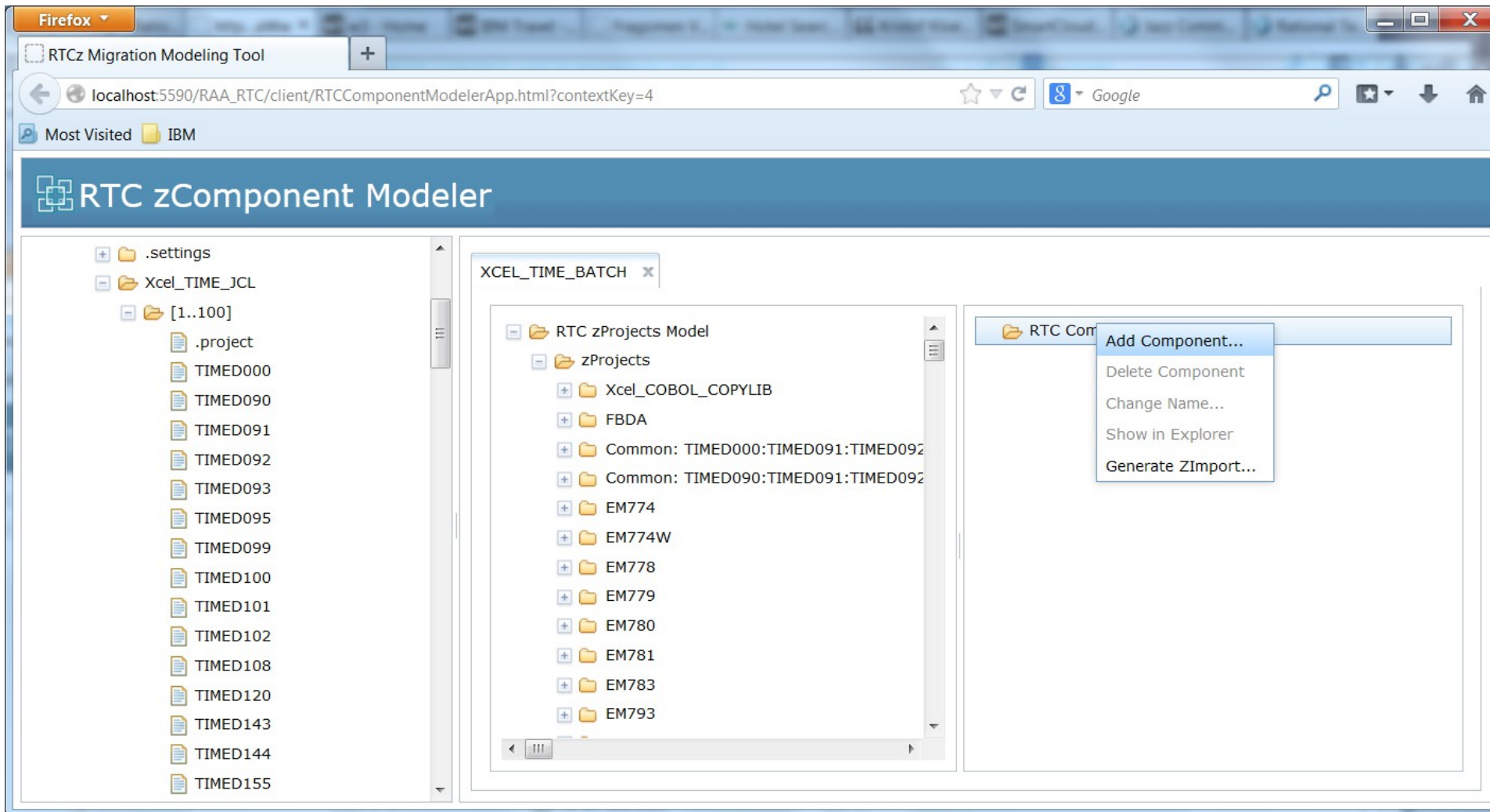
- .settings
- Xcel\_TIME\_JCL
  - [1..100]
    - .project
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099
    - TIMED100
    - TIMED101
    - TIMED102
    - TIMED108
    - TIMED120
    - TIMED143
    - TIMED144
    - TIMED155

XCEL\_TIME\_BATCH

- TIMED376
- TIMED400
  - COBOL
    - TIMEP400.CBL
    - TIMEP405.CBL
    - TIMEP406.CBL
    - TIMEP410.CBL
    - TIMEP415.CBL
    - TIMEP420.CBL
    - TIMEP407.CBL
  - JCL
    - TIMED400
- TIMED424
- TIMED428
- TIMED510

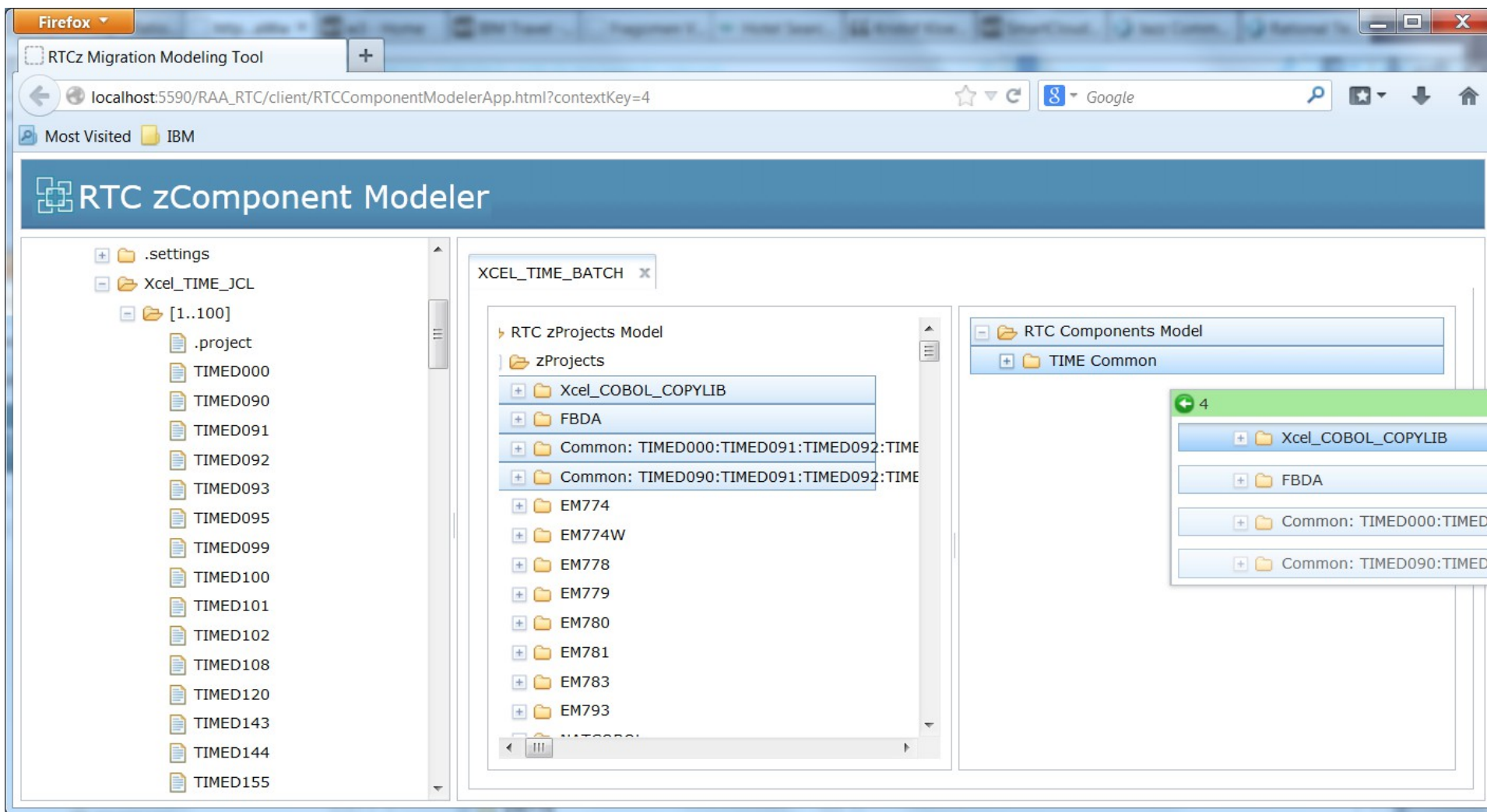
RTC Components Model

# Going a step further ...



The screenshot shows a Firefox browser window displaying the RTC zComponent Modeler application. The browser's address bar shows the URL `localhost:5590/RAA_RTC/client/RTCComponentModelerApp.html?contextKey=4`. The application interface has a dark blue header with the title "RTC zComponent Modeler". On the left, a file explorer shows a tree structure: `.settings`, `Xcel_TIME_JCL`, and a sub-folder `[1..100]` containing files `.project` and `TIMED000` through `TIMED155`. The main workspace is titled "XCEL\_TIME\_BATCH" and contains a tree view for "RTC zProjects Model" with sub-folders: `zProjects`, `Xcel_COBOL_COPYLIB`, `FBDA`, `Common: TIMED000:TIMED091:TIMED092`, `Common: TIMED090:TIMED091:TIMED092`, `EM774`, `EM774W`, `EM778`, `EM779`, `EM780`, `EM781`, `EM783`, and `EM793`. A context menu is open over the "RTC Com" folder, listing options: "Add Component...", "Delete Component", "Change Name...", "Show in Explorer", and "Generate ZImport...".

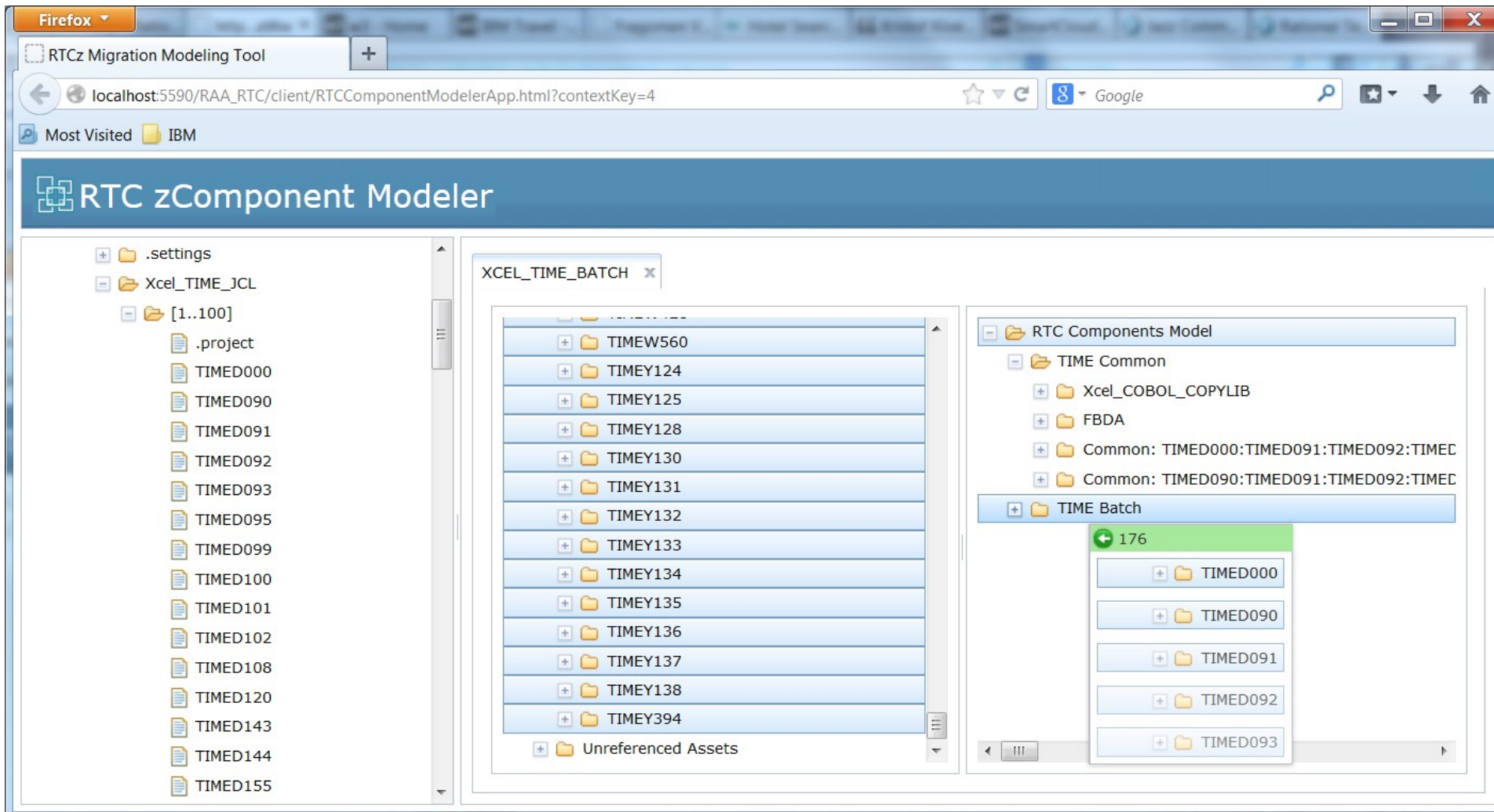
# Going a step further ...



The screenshot shows a Firefox browser window with the address bar displaying `localhost:5590/RAA_RTC/client/RTCComponentModelerApp.html?contextKey=4`. The page title is "RTC zComponent Modeler". The application interface is divided into several panes:

- Left Pane:** A file explorer showing a tree structure. The root is `.settings`, followed by `Xcel_TIME_JCL` and a sub-folder `[1..100]`. Under `[1..100]`, there is a `.project` file and a list of files named `TIMED000` through `TIMED155`.
- Top Pane:** A tab titled `XCEL_TIME_BATCH` is active.
- Center Pane:** Displays the "RTC zProjects Model" tree. It includes a `zProjects` folder containing:
  - `Xcel_COBOL_COPYLIB`
  - `FBDA`
  - `Common: TIMED000:TIMED091:TIMED092:TIME`
  - `Common: TIMED090:TIMED091:TIMED092:TIME`
  - `EM774`
  - `EM774W`
  - `EM778`
  - `EM779`
  - `EM780`
  - `EM781`
  - `EM783`
  - `EM793`
- Right Pane:** Displays the "RTC Components Model" tree, which includes a `TIME Common` folder. A tooltip is visible over this pane, showing a list of components with a green header containing a plus sign and the number "4". The components listed are:
  - `Xcel_COBOL_COPYLIB`
  - `FBDA`
  - `Common: TIMED000:TIMED`
  - `Common: TIMED090:TIMED`

# Going a step further ...



The screenshot shows a web browser window with the title "RTCz Migration Modeling Tool". The address bar shows the URL "localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4". The browser's "Most Visited" list includes "IBM".

The application interface is titled "RTC zComponent Modeler". It features a file explorer on the left with a tree view showing the following structure:

- .settings
- Xcel\_TIME\_JCL
  - [1..100]
    - .project
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099
    - TIMED100
    - TIMED101
    - TIMED102
    - TIMED108
    - TIMED120
    - TIMED143
    - TIMED144
    - TIMED155

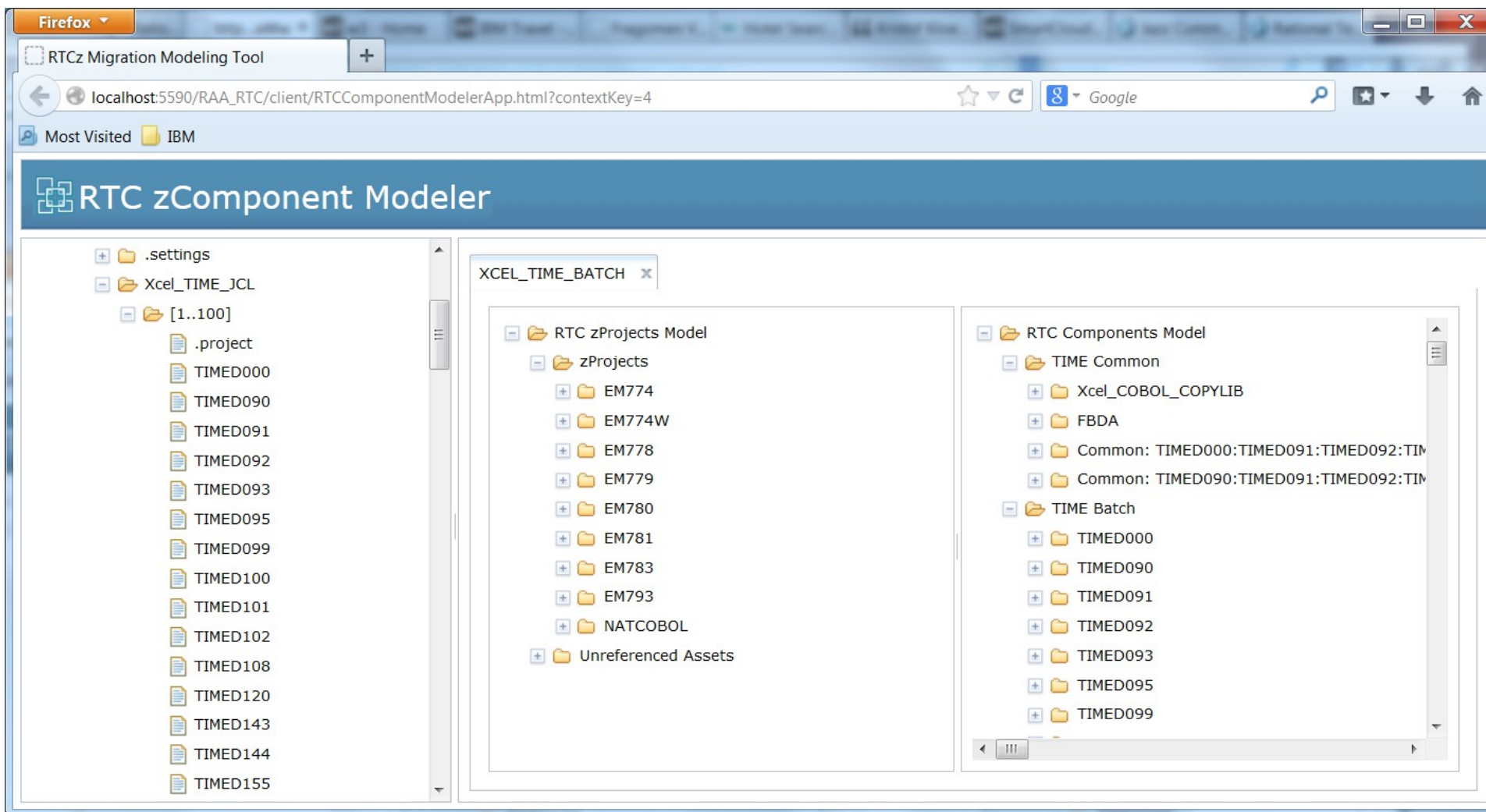
The main workspace is divided into two panes. The left pane, titled "XCEL\_TIME\_BATCH", displays a list of folders:

- TIMEW560
- TIMEY124
- TIMEY125
- TIMEY128
- TIMEY130
- TIMEY131
- TIMEY132
- TIMEY133
- TIMEY134
- TIMEY135
- TIMEY136
- TIMEY137
- TIMEY138
- TIMEY394
- Unreferenced Assets

The right pane, titled "RTC Components Model", shows a hierarchical view:

- RTC Components Model
  - TIME Common
    - Xcel\_COBOL\_COPYLIB
    - FBDA
    - Common: TIMED000:TIMED091:TIMED092:TIMEC
    - Common: TIMED090:TIMED091:TIMED092:TIMEC
    - TIME Batch
      - 176
        - TIMED000
        - TIMED090
        - TIMED091
        - TIMED092
        - TIMED093

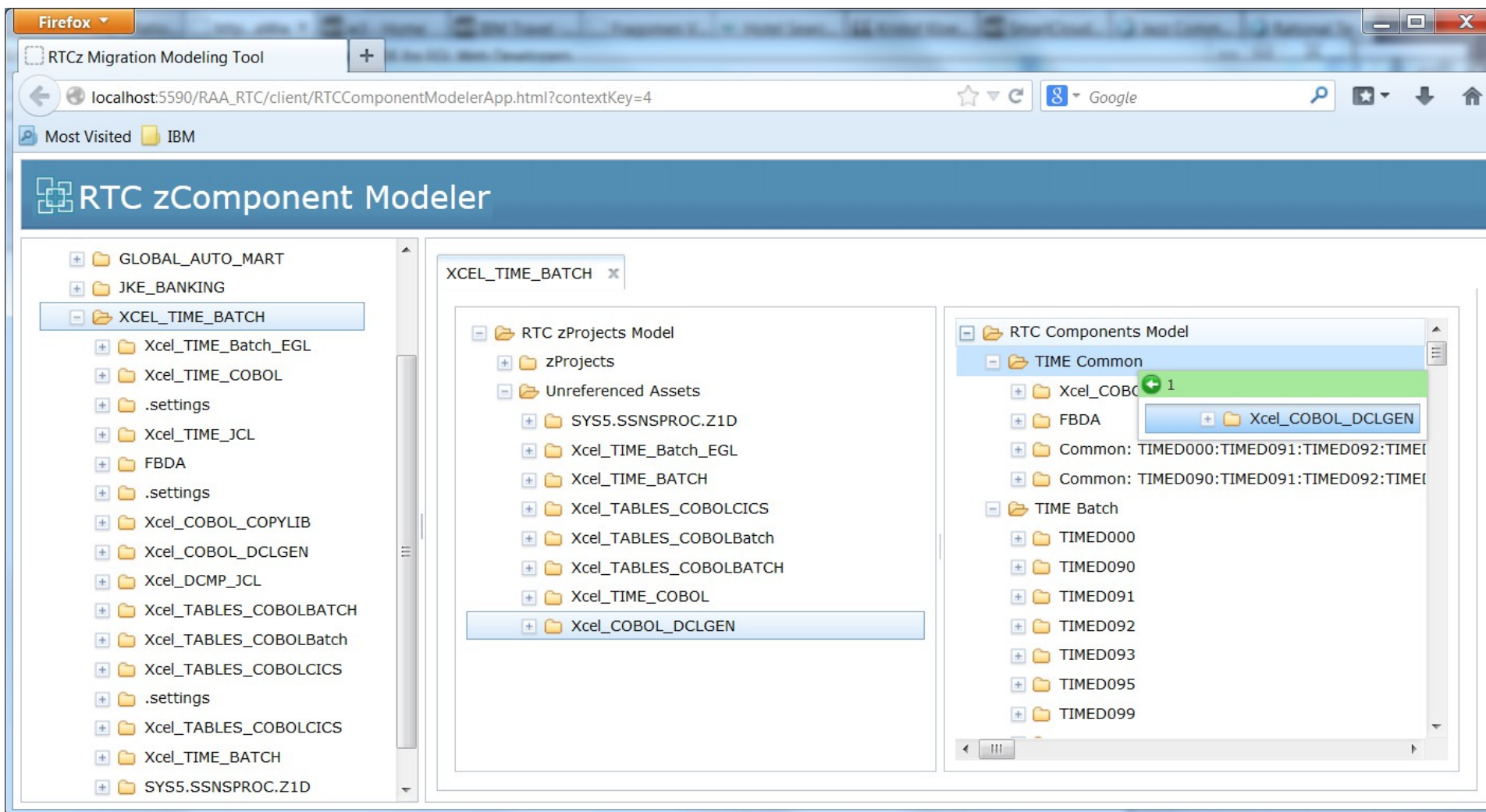
# Going a step further ...



The screenshot shows a Firefox browser window displaying the RTC zComponent Modeler application. The browser's address bar shows the URL: localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4. The application interface has a blue header with the title "RTC zComponent Modeler". Below the header, there are three main panels:

- Left Panel:** A file tree showing the project structure. It includes a ".settings" folder, an "Xcel\_TIME\_JCL" folder, and a "[1..100]" folder containing a ".project" file and a list of "TIMED" folders from TIMED000 to TIMED155.
- Center Panel:** A window titled "XCEL\_TIME\_BATCH" showing a tree view of the "RTC zProjects Model". It contains a "zProjects" folder with sub-folders: EM774, EM774W, EM778, EM779, EM780, EM781, EM783, EM793, NATCOBOL, and Unreferenced Assets.
- Right Panel:** A window titled "RTC Components Model" showing a tree view of the "RTC Components Model". It contains a "TIME Common" folder with sub-folders: Xcel\_COBOL\_COPYLIB, FBDA, and two "Common" folders (one for TIMED000:TIMED091:TIMED092:TIMED093 and one for TIMED090:TIMED091:TIMED092:TIMED093). Below this is a "TIME Batch" folder containing individual "TIMED" folders from TIMED000 to TIMED099.

# Going a step further ...



Firefox

RTCz Migration Modeling Tool

localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4

Most Visited IBM

## RTC zComponent Modeler

- GLOBAL\_AUTO\_MART
- JKE\_BANKING
- XCEL\_TIME\_BATCH**
  - Xcel\_TIME\_Batch\_EGL
  - Xcel\_TIME\_COBOL
  - .settings
  - Xcel\_TIME\_JCL
  - FBDA
  - .settings
  - Xcel\_COBOL\_COPYLIB
  - Xcel\_COBOL\_DCLGEN
  - Xcel\_DCMP\_JCL
  - Xcel\_TABLES\_COBOLBATCH
  - Xcel\_TABLES\_COBOLBatch
  - Xcel\_TABLES\_COBOLCICS
  - .settings
  - Xcel\_TABLES\_COBOLCICS
  - Xcel\_TIME\_BATCH
  - SYS5.SSNSPROC.Z1D

XCEL\_TIME\_BATCH

- RTC zProjects Model
  - zProjects
  - Unreferenced Assets
    - SYS5.SSNSPROC.Z1D
    - Xcel\_TIME\_Batch\_EGL
    - Xcel\_TIME\_BATCH
    - Xcel\_TABLES\_COBOLCICS
    - Xcel\_TABLES\_COBOLBatch
    - Xcel\_TABLES\_COBOLBATCH
    - Xcel\_TIME\_COBOL
    - Xcel\_COBOL\_DCLGEN**

RTC Components Model

- TIME Common
  - Xcel\_COBOL\_DCLGEN** 1
  - FBDA
  - Common: TIMED000:TIMED091:TIMED092:TIMEI
  - Common: TIMED090:TIMED091:TIMED092:TIMEI
- TIME Batch
  - TIMED000
  - TIMED090
  - TIMED091
  - TIMED092
  - TIMED093
  - TIMED095
  - TIMED099

# Going a step further ...

The screenshot shows the RTC zComponent Modeler application running in a Firefox browser. The browser address bar shows the URL: localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4. The application interface has a header with the title "RTC zComponent Modeler". On the left, there is a tree view of project components. The selected component is "XCEL\_TIME\_BATCH", which is expanded to show sub-components like "Xcel\_TIME\_Batch\_EGL", "Xcel\_TIME\_COBOL", ".settings", "Xcel\_TIME\_JCL", "FBDA", ".settings", "Xcel\_COBOL\_COPYLIB", "Xcel\_COBOL\_DCLGEN", "Xcel\_DCMP\_JCL", "Xcel\_TABLES\_COBOLBATCH", "Xcel\_TABLES\_COBOLBatch", "Xcel\_TABLES\_COBOLCICS", ".settings", "Xcel\_TABLES\_COBOLCICS", "Xcel\_TIME\_BATCH", "SYS5.SSNSPROC.Z1D", and ".settings".

In the center, there is a "Change Name" dialog box with the following fields and buttons:

- Dialog Title: Change Name
- Field: Name: (with an empty text input box)
- Buttons: OK, Cancel

On the right side of the application, there is another tree view showing "RTC Components Model". The "TIME Batch" folder is selected, showing sub-components: "TIMED000", "TIMED090", "TIMED091", "TIMED092", "TIMED093", "TIMED095", "TIMED099", and "TIMED100".

# Going a step further ...

Firefox

RTCz Migration Modeling Tool

localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4

Google

Most Visited IBM

## RTC zComponent Modeler

- EXPRESS\_SCRIPTS
- GLOBAL\_AUTO\_MART
- JKE\_BANKING
- XCEL\_TIME\_BATCH**
  - Xcel\_TIME\_Batch\_EGL
  - Xcel\_TIME\_COBOL
  - .settings
  - Xcel\_TIME\_JCL
  - FBDA
  - .settings
  - Xcel\_COBOL\_COPYLIB
  - Xcel\_COBOL\_DCLGEN
  - Xcel\_DCMP\_JCL
  - Xcel\_TABLES\_COBOLBATCH
  - Xcel\_TABLES\_COBOLBatch
  - Xcel\_TABLES\_COBOLCICS
  - .settings
  - Xcel\_TABLES\_COBOLCICS
  - Xcel\_TIME\_BATCH
  - SYS5.SSNSPROC.Z1D
  - .settings

XCEL\_TIME\_BATCH

- RTC zProjects Model
  - zProjects
  - Unreferenced Assets
    - SYS5.SSNSPROC.Z1D
    - Xcel\_TIME\_Batch\_EGL
    - Xcel\_TIME\_BATCH
    - Xcel\_TABLES\_COBOLCICS
    - Xcel\_TABLES\_COBOLBatch
    - Xcel\_TABLES\_COBOLBATCH
    - Xcel\_TIME\_COBOL

RTC Components Model

- TIME Common
  - Xcel\_COBOL
  - FBDA
  - Common JCL
  - Common COBOL
  - Xcel\_COBOL\_DCLGEN
- TIME Batch
  - TIMED000
  - TIMED090
  - TIMED091
  - TIMED092
  - TIMED093
  - TIMED095
  - TIMED099
  - TIMED100
  - TIMED101

- Add Component...
- Delete Component
- Change Name...
- Show in Explorer
- Generate ZImport...**



# Going a step further ...

The screenshot shows the RTC zComponent Modeler application running in a Firefox browser. The browser address bar shows the URL: localhost:5590/RAA\_RTC/client/RTCComponentModelerApp.html?contextKey=4. The application interface displays a tree view of project folders on the left, including EXPRESS\_SCRIPTS, GLOBAL\_AUTO\_MART, JKE\_BANKING, and XCEL\_TIME\_BATCH. The XCEL\_TIME\_BATCH folder is expanded, showing sub-folders like Xcel\_TIME\_Batch\_EGL, Xcel\_TIME\_COBOL, .settings, Xcel\_TIME\_JCL, FBDA, .settings, Xcel\_COBOL\_COPYLIB, Xcel\_COBOL\_DCLGEN, Xcel\_DCOMP\_JCL, Xcel\_TABLES\_COBOLBAT, Xcel\_TABLES\_COBOLBat, Xcel\_TABLES\_COBOLCICS, .settings, Xcel\_TABLES\_COBOLCICS, Xcel\_TIME\_BATCH, SYS5.SSNSPROC.Z1D, and .settings.

A ZImport dialog box is open in the foreground, displaying a list of JCL jobs. The jobs listed are:

```
C:E:/Xcel/Xcel_COBOL_COPYLIB=TIME Common
P:E:/Xcel/Xcel_COBOL_COPYLIB(*)=Xcel_COBOL_COPYLIB
C:E:/Xcel/FBDA=TIME Common
P:E:/Xcel/FBDA(*)=FBDA
C:Common JCL=TIME Common
P:E:/Xcel/SYS5.SSNSPROC.Z1D(EGLPRD)=Common JCL:PROC
C:Common COBOL=TIME Common
P:E:/Xcel/Xcel_TIME_BATCH(AD506A.cb)=Common COBOL:COBOL
C:E:/Xcel/Xcel_COBOL_DCLGEN=TIME Common
P:E:/Xcel/Xcel_COBOL_DCLGEN(*)=Xcel_COBOL_DCLGEN
C:TIMED000=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED000)=TIMED000:JCL
C:TIMED090=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED090)=TIMED090:JCL
C:TIMED091=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED091)=TIMED091:JCL
C:TIMED092=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED092)=TIMED092:JCL
C:TIMED093=TIME Batch
P:E:/Xcel/Xcel_TIME_BATCH(TIMEP093.cb)=TIMED093:COBOL
P:E:/Xcel/Xcel_TIME_JCL(TIMED093)=TIMED093:JCL
C:TIMED095=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED095)=TIMED095:JCL
C:TIMED099=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED099)=TIMED099:JCL
C:TIMED100=TIME Batch
P:E:/Xcel/Xcel_TIME_JCL(TIMED100)=TIMED100:JCL
C:TIMED101=TIME Batch
P:E:/Xcel/Xcel_TIME_BATCH(TIMEP101.cb)=TIMED101:COBOL
P:E:/Xcel/Xcel_TABLES_COBOLBAT(Xcel_TABLES_COBOLBAT(AV4000.CB))=TIMED101:COBOL
```

# Summary

- z/OS application development benefits from RTC project, stream, and component structure
- Identifying the appropriate groupings is a combination of algorithm and heuristics (knowledge and experience)
- Application development will improve
  - Program understanding
  - Faster project load
  - Faster build
  - Faster test identification
- More productive development enabling faster turn-around on fixes, changes, and enhancements

# Questions

# Additional Resources

- IBM Integrated Solution for System z Development test drive:  
[http://www.ibm.com/developerworks/downloads/rdtsandbox\\_systemz/index.html](http://www.ibm.com/developerworks/downloads/rdtsandbox_systemz/index.html)
- To learn more about the solution offering, see the IBM Integrated Solution for System z Development
- To learn more about the EM offerings – see the individual product pages:
  - [Rational Developer for zEnterprise](#) or [Rational Team Concert](#) or [Rational Asset Analyzer](#) or [Rational Development and Test Environment for System z](#)