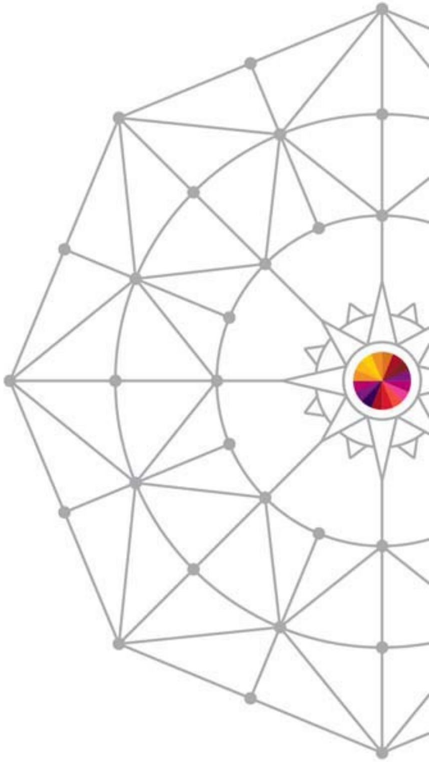
Where Copybooks Go and Rational Developer for System z and Rational Team Concert Implementation Questions



Venkat Balabhadrapatruni

venkatu@us.ibm.com

August 6th, 2014

Session: 15474

#SHAREorg

Copyright (c) 2014 by SHARE Inc.

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

Please note

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

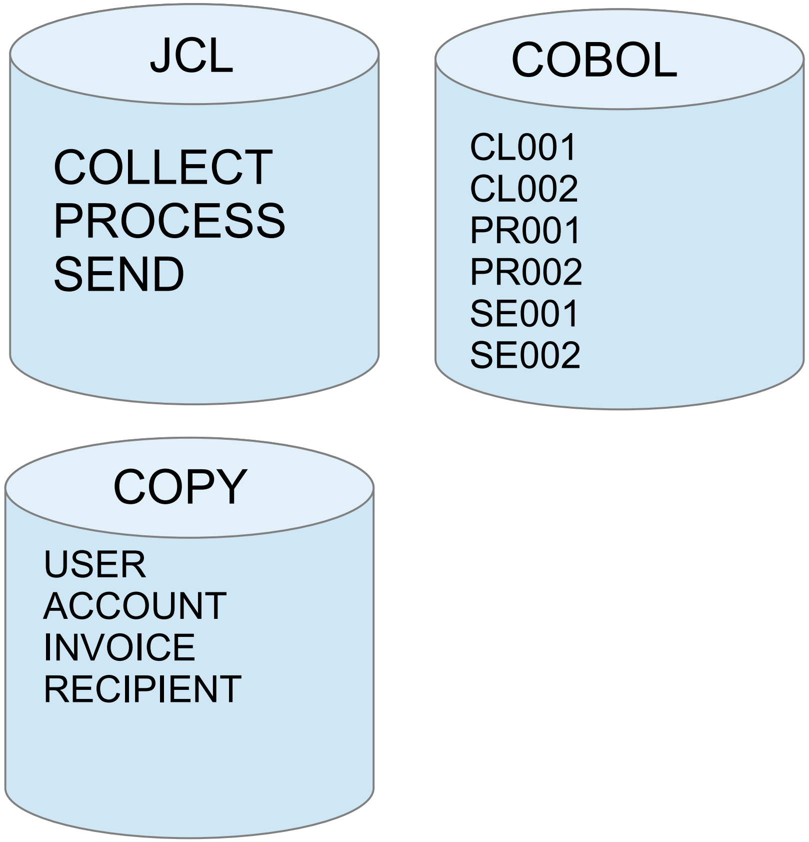
High Level Approach

- 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



JCL

COLLECT
PROCESS
SEND

COBOL

CL001
CL002
PR001
PR002
SE001
SE002

COPY

USER
ACCOUNT
INVOICE
RECIPIENT

zComponent Project

zFolders

zFiles

- COLLECT
  - JCL
    - COLLECT
  - COBOL
    - CL001
    - CL002
- PROCESS
  - JCL
    - PROCESS
  - COBOL
    - PR001
    - PR002
- SEND
  - JCL
    - SEND
  - COBOL
    - SE001
    - SE002
  - COPY
    - RECIPIENT
- COMMON
  - COPY
    - USER
    - ACCOUNT
    - INVOICE

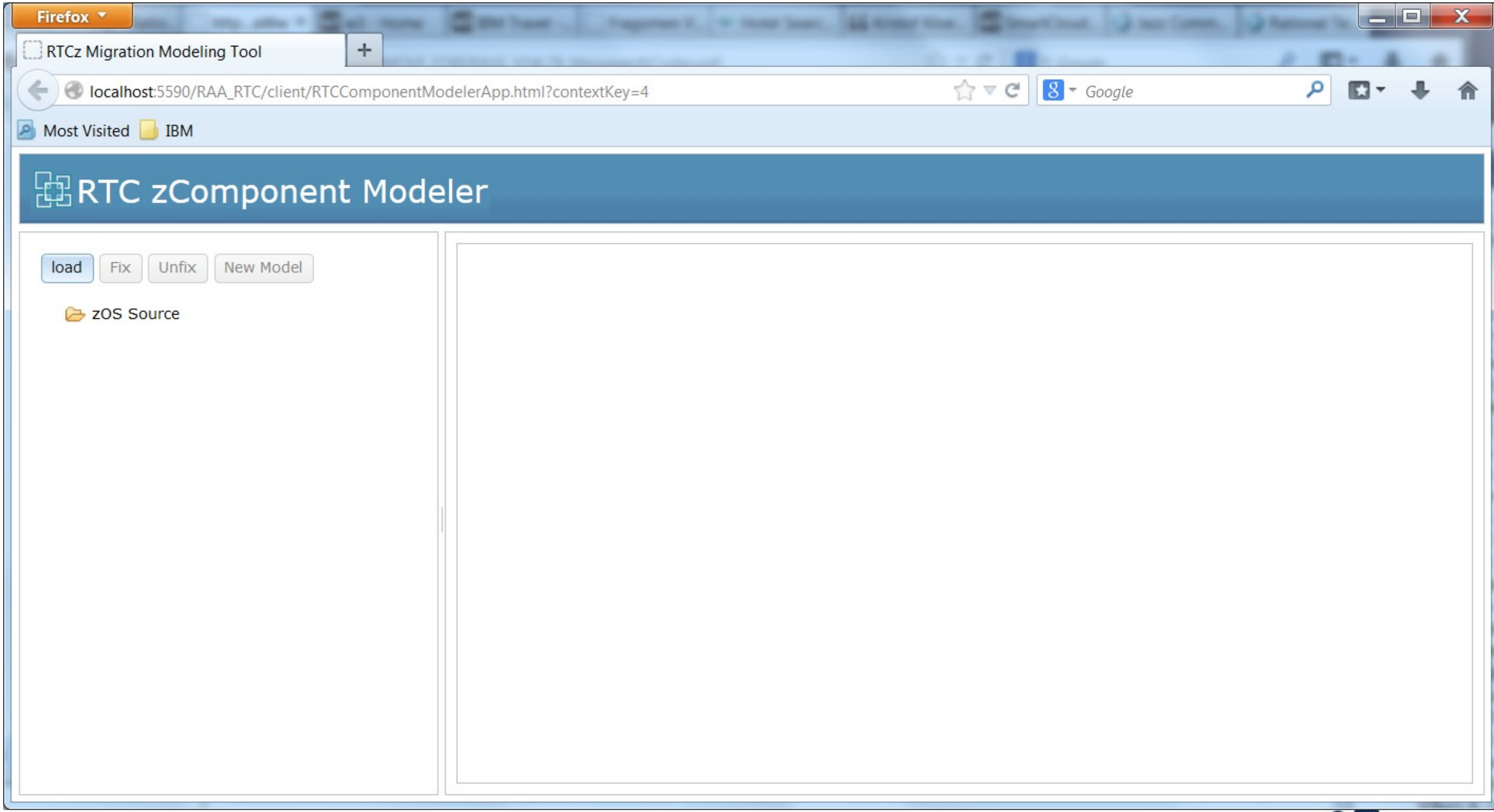
Scaling Up

- 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

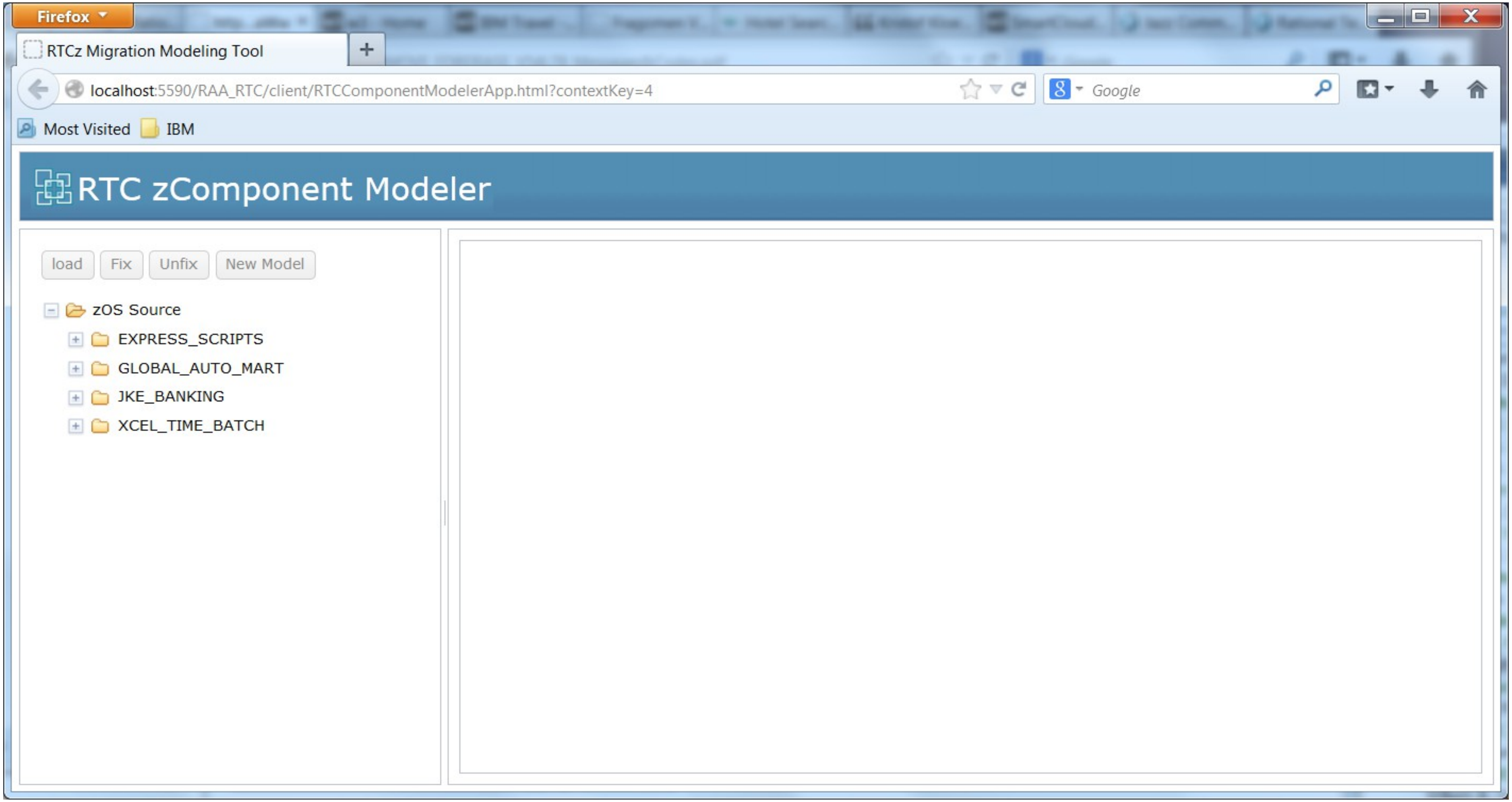
- Rational Asset Analyzer
  - Graphical views of program structure
  - Reports and queries to identify included and referenced files
- Rational Developer for System z
  - Graphical views of program structure
  - Code navigation to navigate to referenced files
  - Dataset search filters to guide common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

Going a step further …

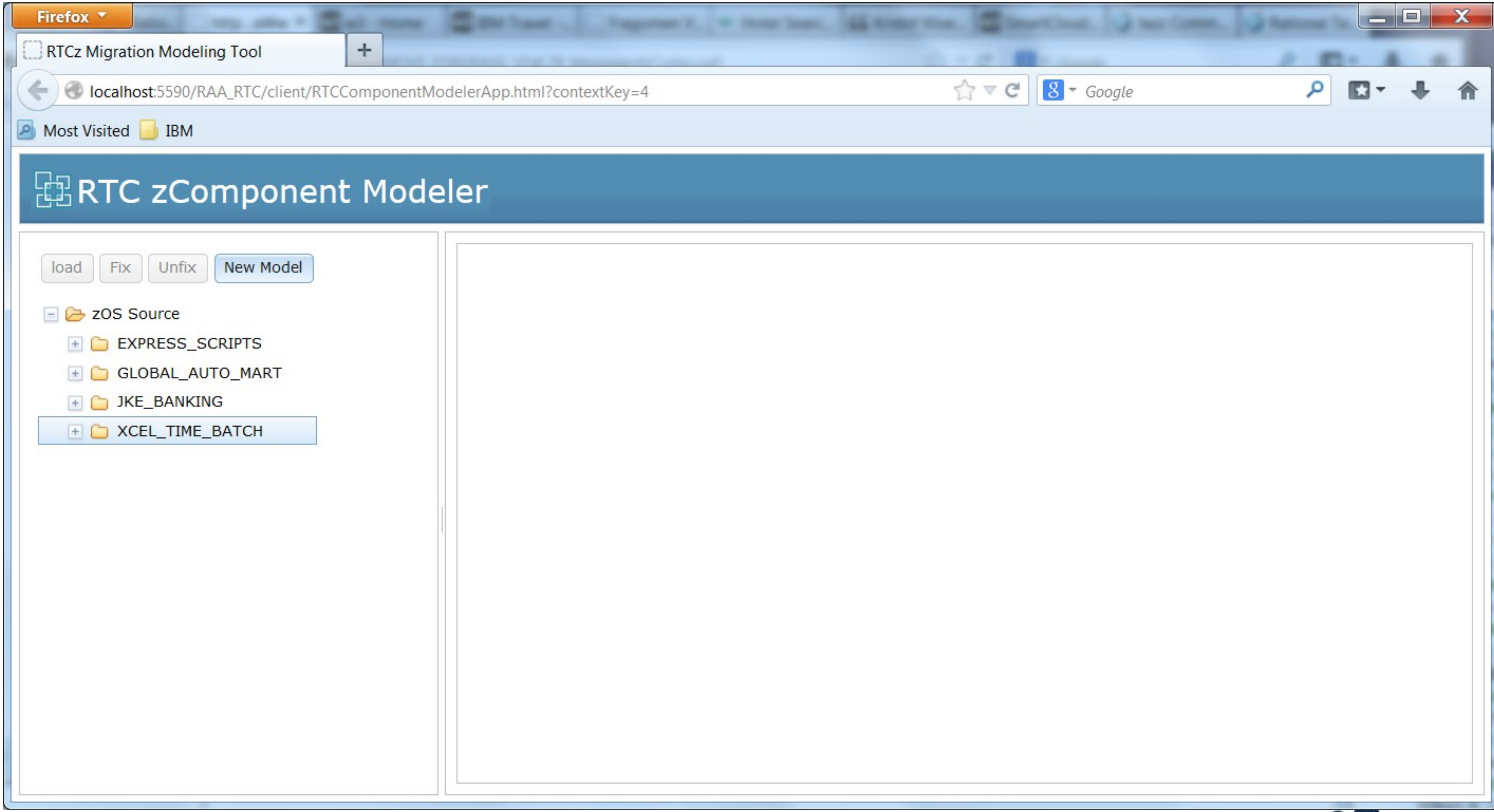


Firefox
RTCz Migration Modeling Tool
localhost:5590/RAA_RTC/client/RTCComponentModelerApp.html?contextKey=4
Google
Most Visited
IBM
RTC zComponent Modeler
load
Fix
Unfix
New Model
zOS Source

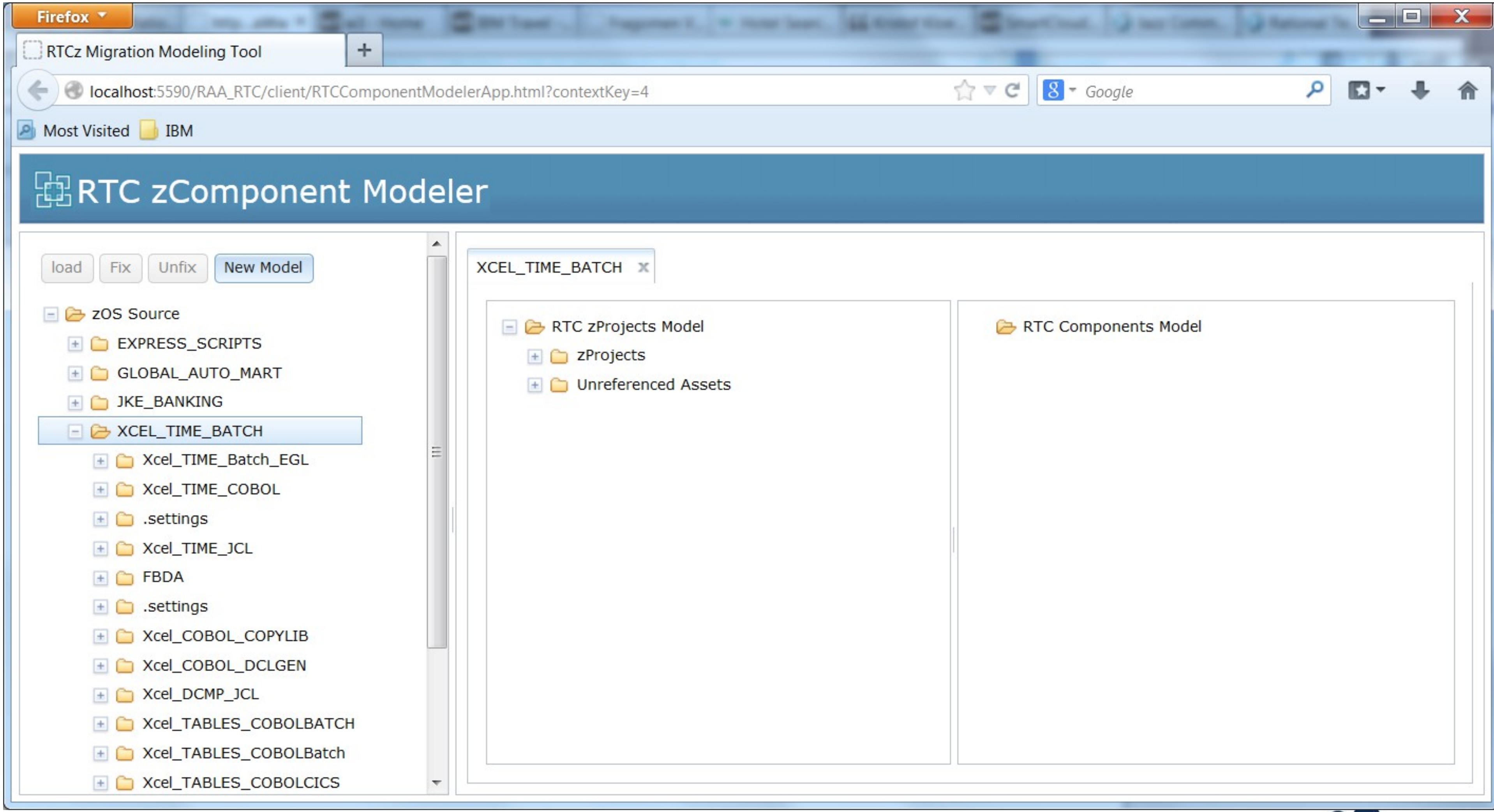
Going a step further …



Going a step further …



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

- 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

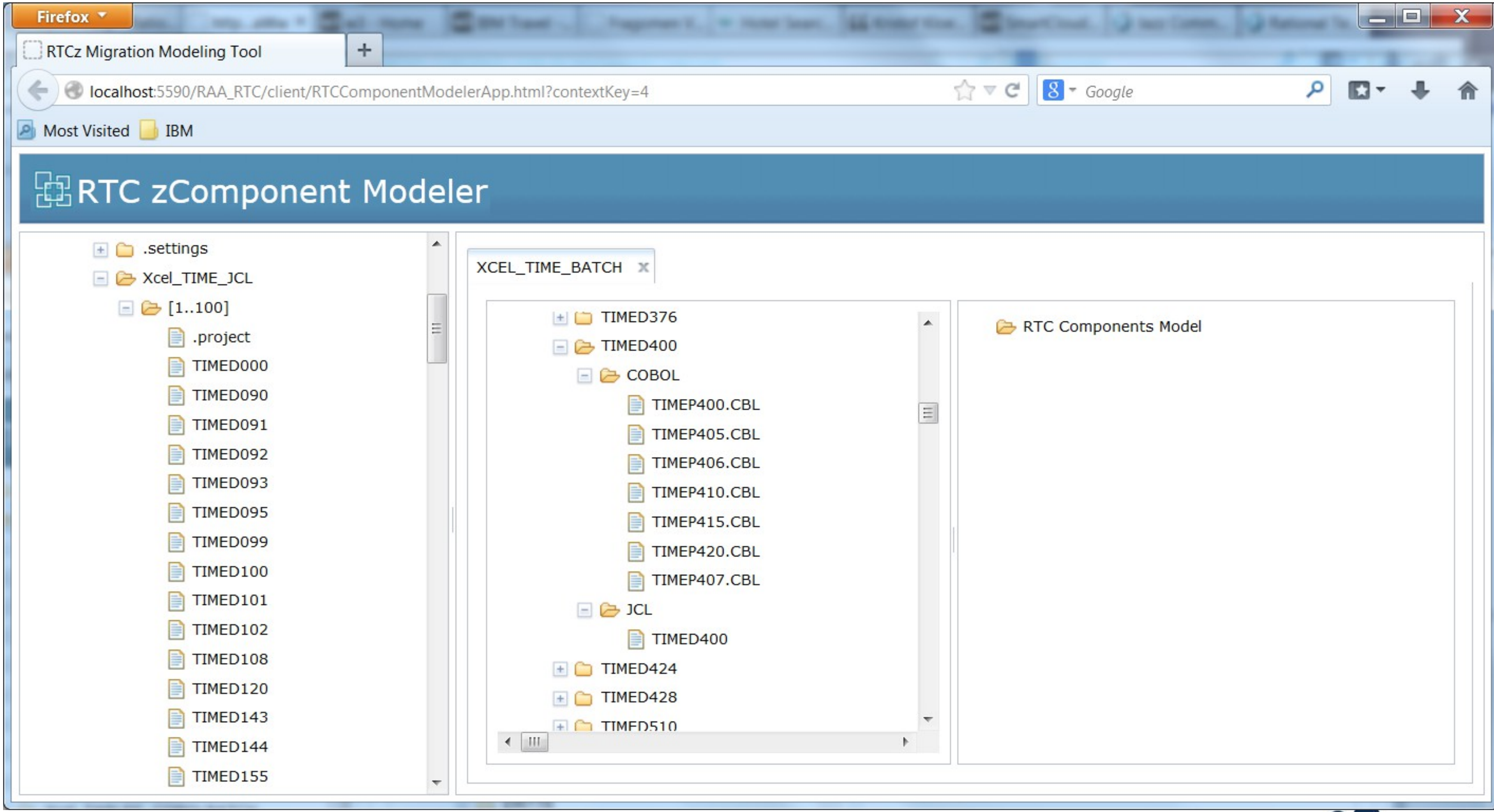
XCEL_TIME_BATCH

- RTC zProjects Model
  - zProjects
  - Unreferenced Assets

- RTC Components Model

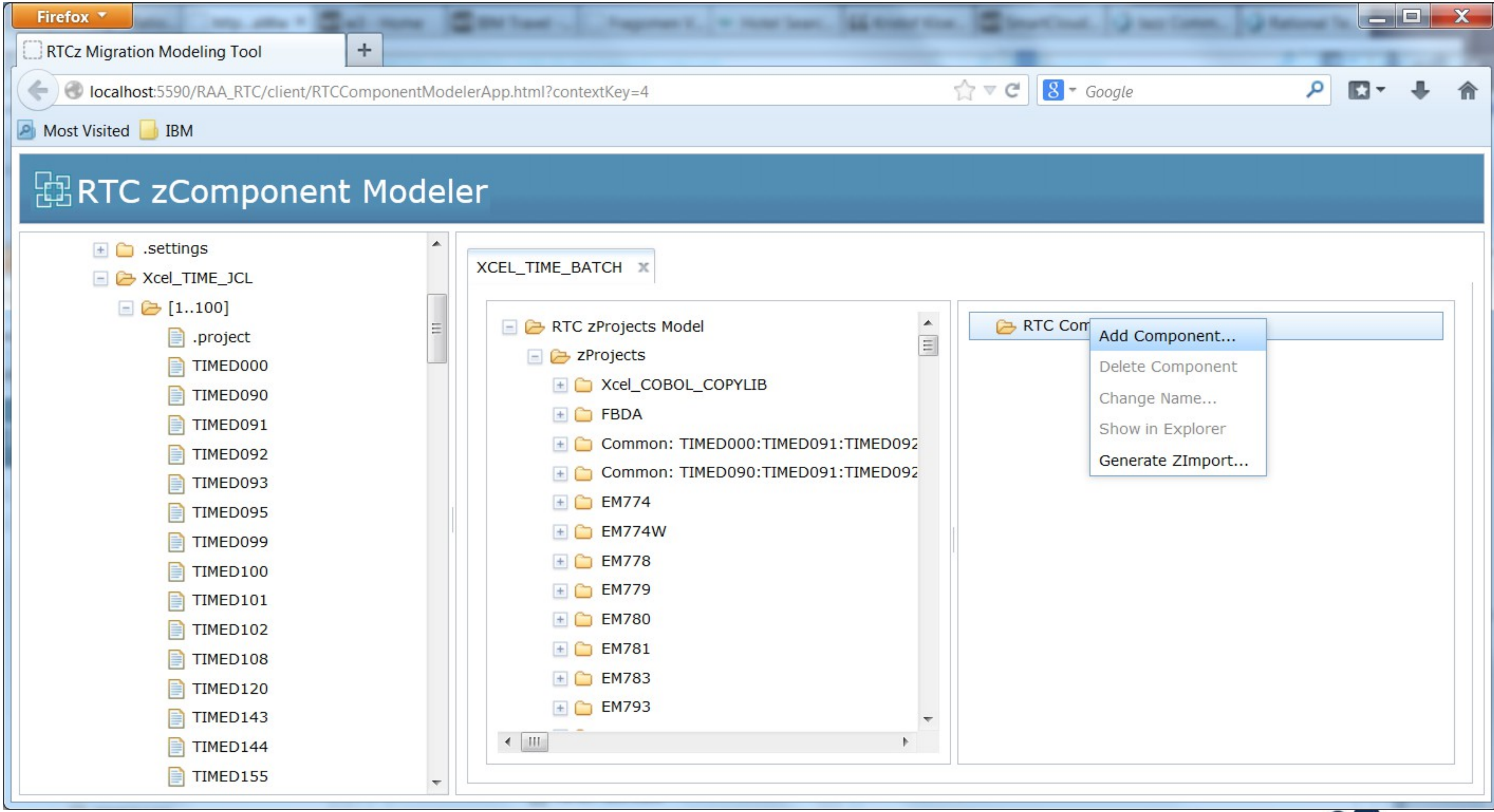
Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Going a step further …



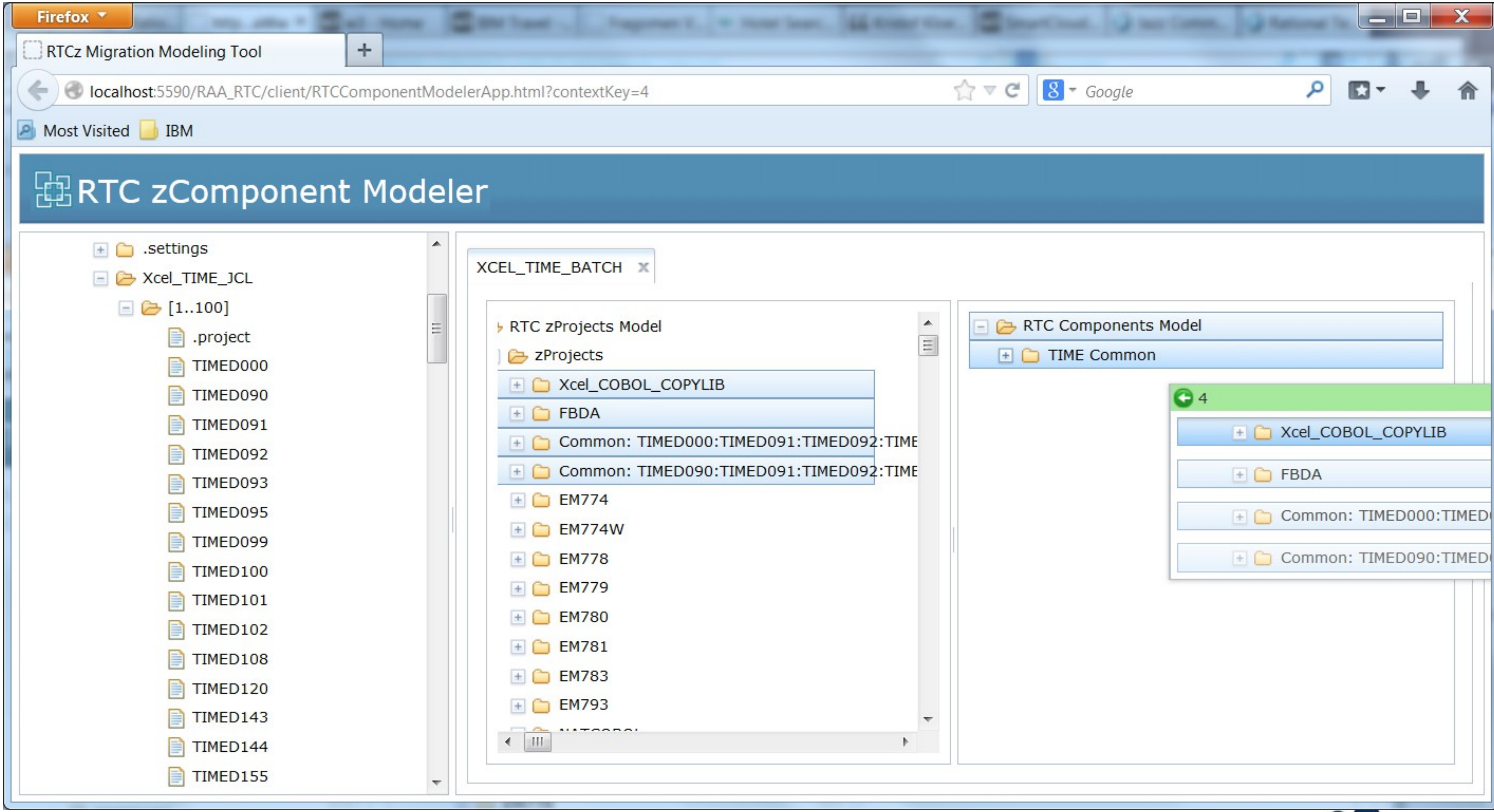
Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Going a step further …



Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

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

- .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

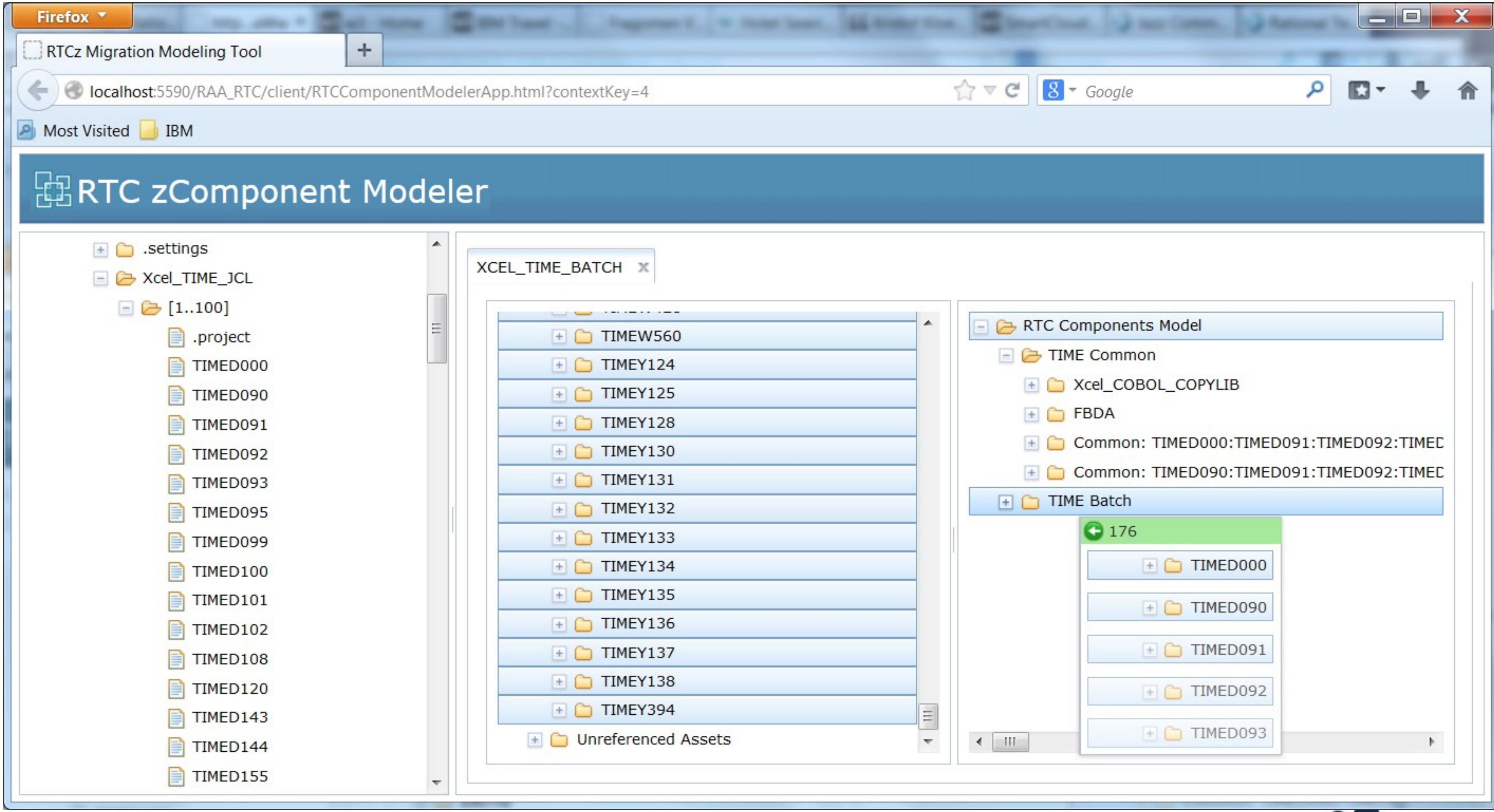
- RTC zProjects Model
  - zProjects
    - Xcel_COBOL_COPYLIB
    - FBDA
    - Common: TIMED000:TIMED091:TIMED092:TIME
    - Common: TIMED090:TIMED091:TIMED092:TIME
    - EM774
    - EM774W
    - EM778
    - EM779
    - EM780
    - EM781
    - EM783
    - EM793

- RTC Components Model
  - TIME Common

4

- Xcel_COBOL_COPYLIB
- FBDA
- Common: TIMED000:TIMED
- Common: TIMED090:TIMED

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

- .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

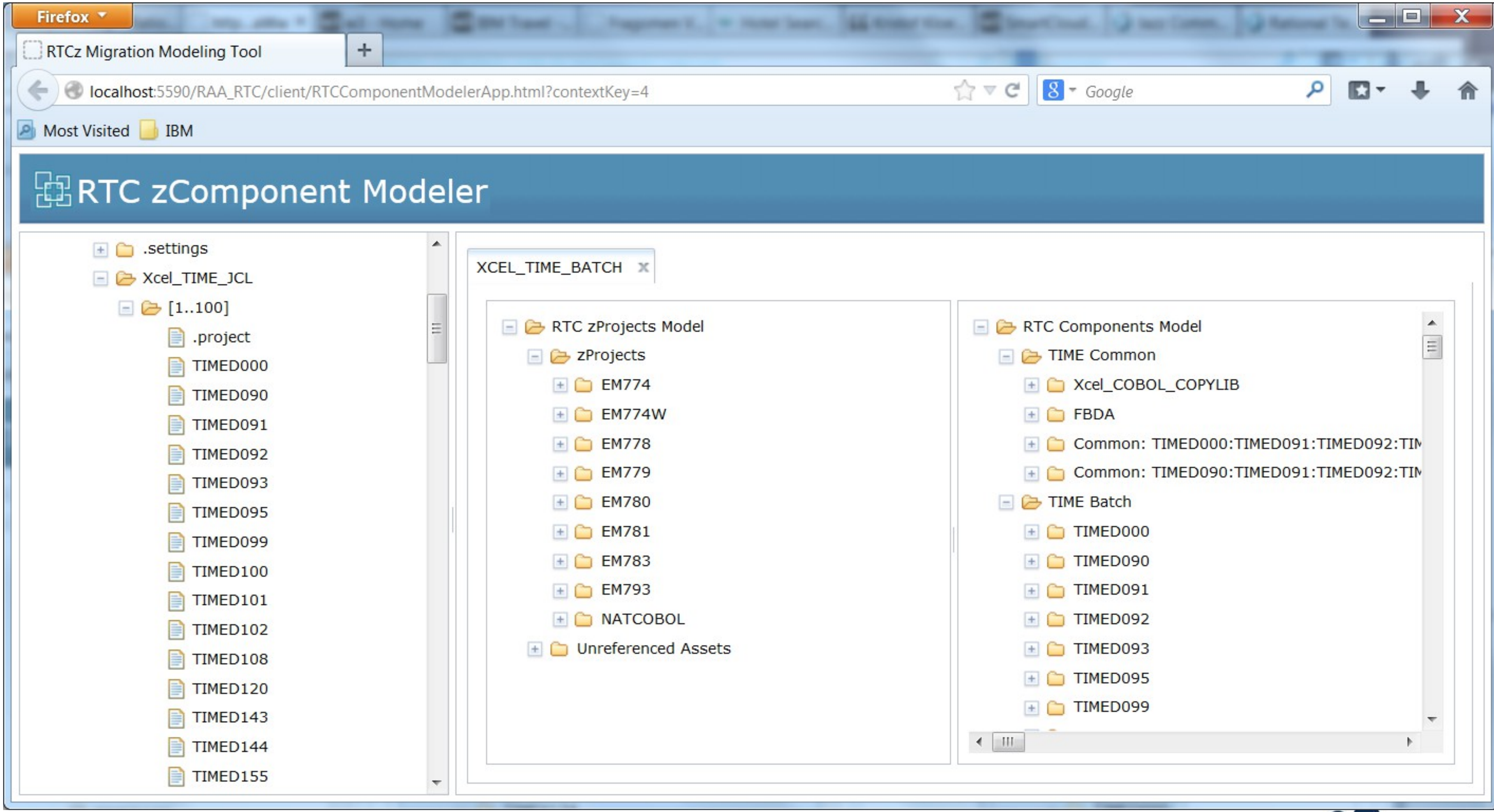
- TIMEW560
- TIMEY124
- TIMEY125
- TIMEY128
- TIMEY130
- TIMEY131
- TIMEY132
- TIMEY133
- TIMEY134
- TIMEY135
- TIMEY136
- TIMEY137
- TIMEY138
- TIMEY394
- Unreferenced Assets

- RTC Components Model
  - TIME Common
    - Xcel_COBOL_COPYLIB
    - FBDA
    - Common: TIMED000:TIMED091:TIMED092:TIMED
    - Common: TIMED090:TIMED091:TIMED092:TIMED
  - TIME Batch

176

- TIMED000
- TIMED090
- TIMED091
- TIMED092
- TIMED093

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

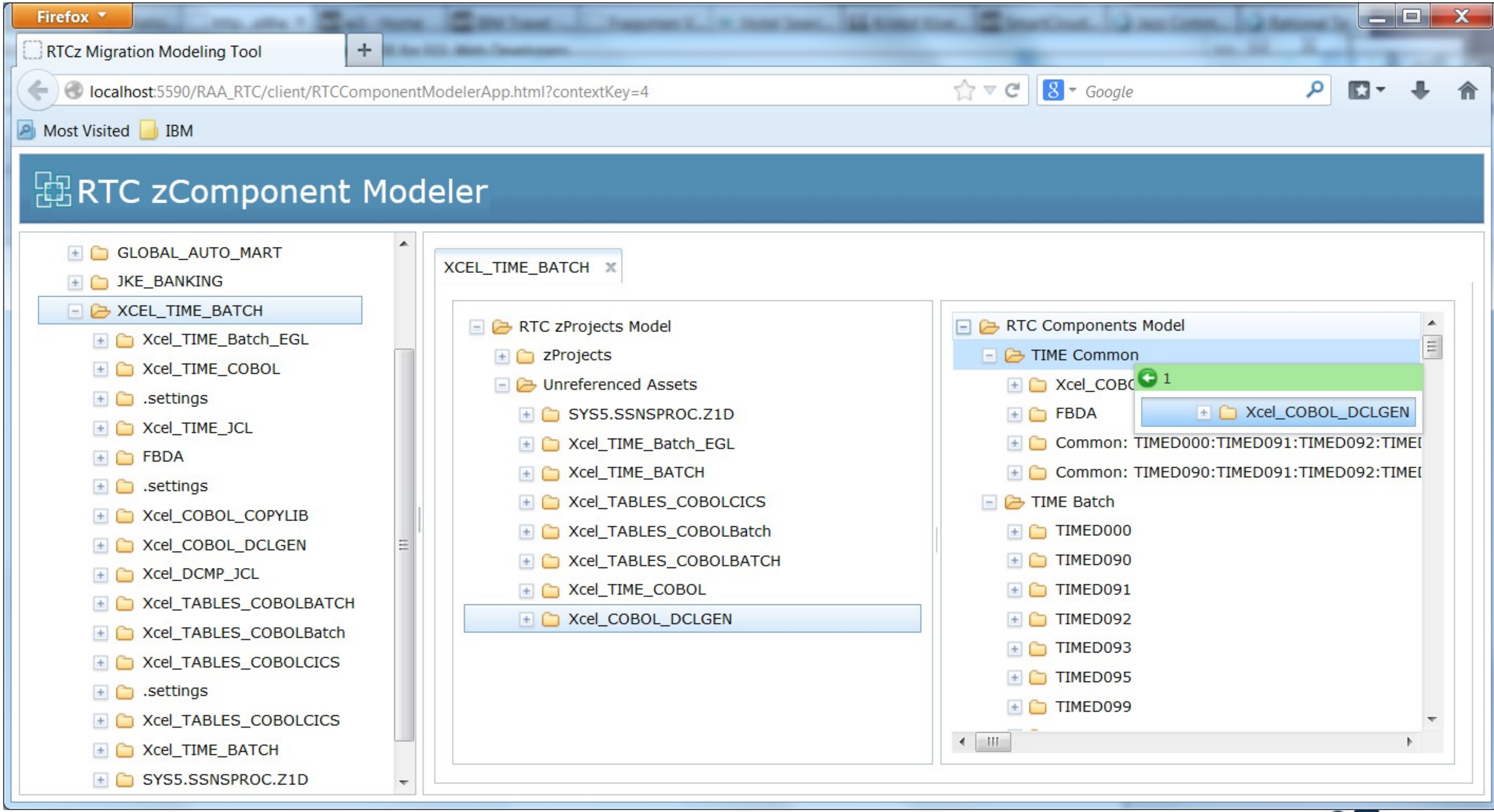
- .settings
- Xcel_TIME_JCL
  - [1..100]
    - .project
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099
    - TIMED100
    - TIMED101
    - TIMED102
    - TIMED108
    - TIMED120
    - TIMED143
    - TIMED144
    - TIMED155

XCEL_TIME_BATCH ×

- RTC zProjects Model
  - zProjects
    - EM774
    - EM774W
    - EM778
    - EM779
    - EM780
    - EM781
    - EM783
    - EM793
    - NATCOBOL
  - Unreferenced Assets

- RTC Components Model
  - TIME Common
    - Xcel_COBOL_COPYLIB
    - FBDA
    - Common: TIMED000:TIMED091:TIMED092:TIM
    - Common: TIMED090:TIMED091:TIMED092:TIM
  - TIME Batch
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099

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

- 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

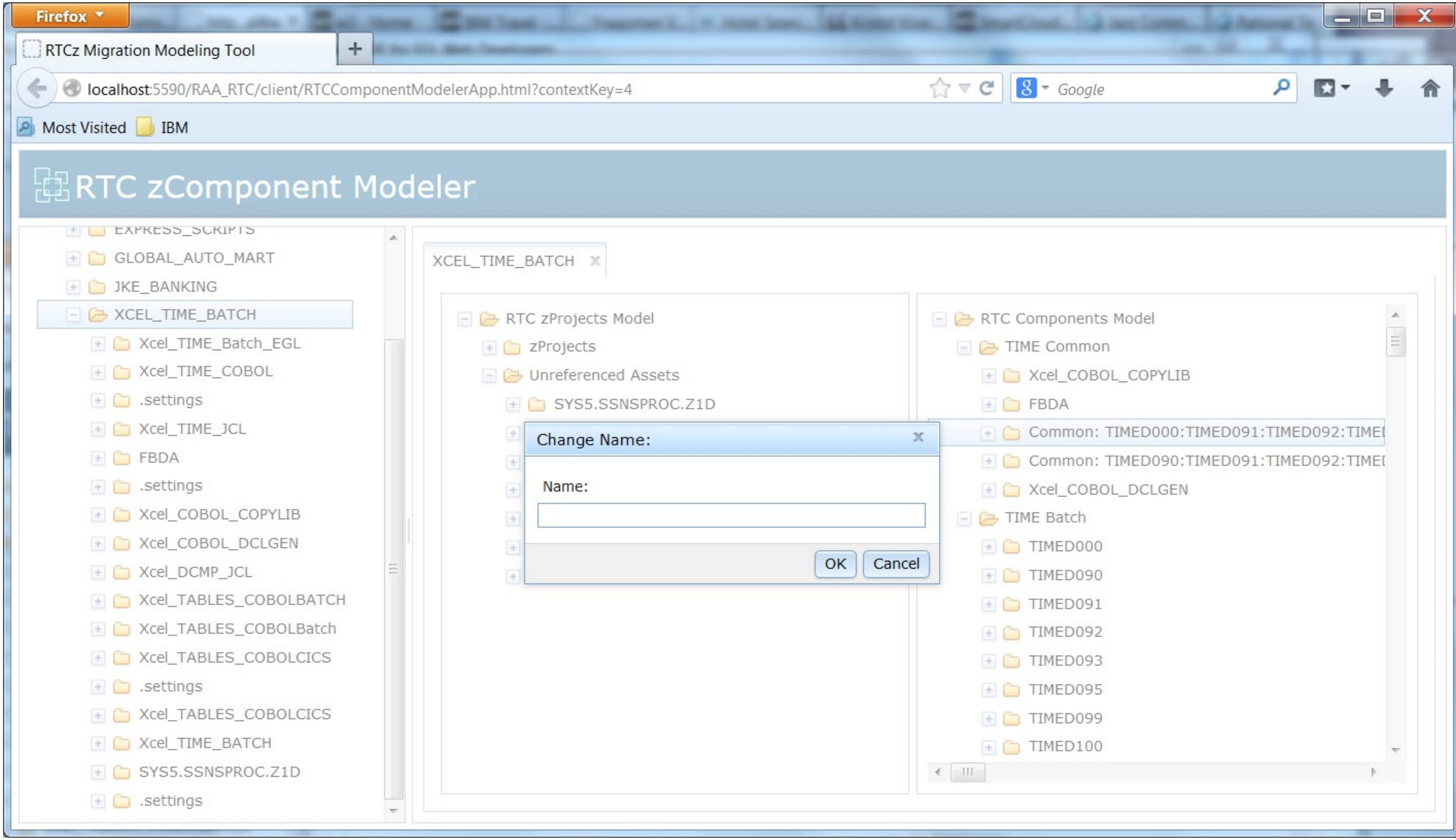
- RTC Components Model
  - TIME Common
    - Xcel_COBC
    - FBDA
    - Common: TIMED000:TIMED091:TIMED092:TIMEI
    - Common: TIMED090:TIMED091:TIMED092:TIMEI
  - TIME Batch
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099

1

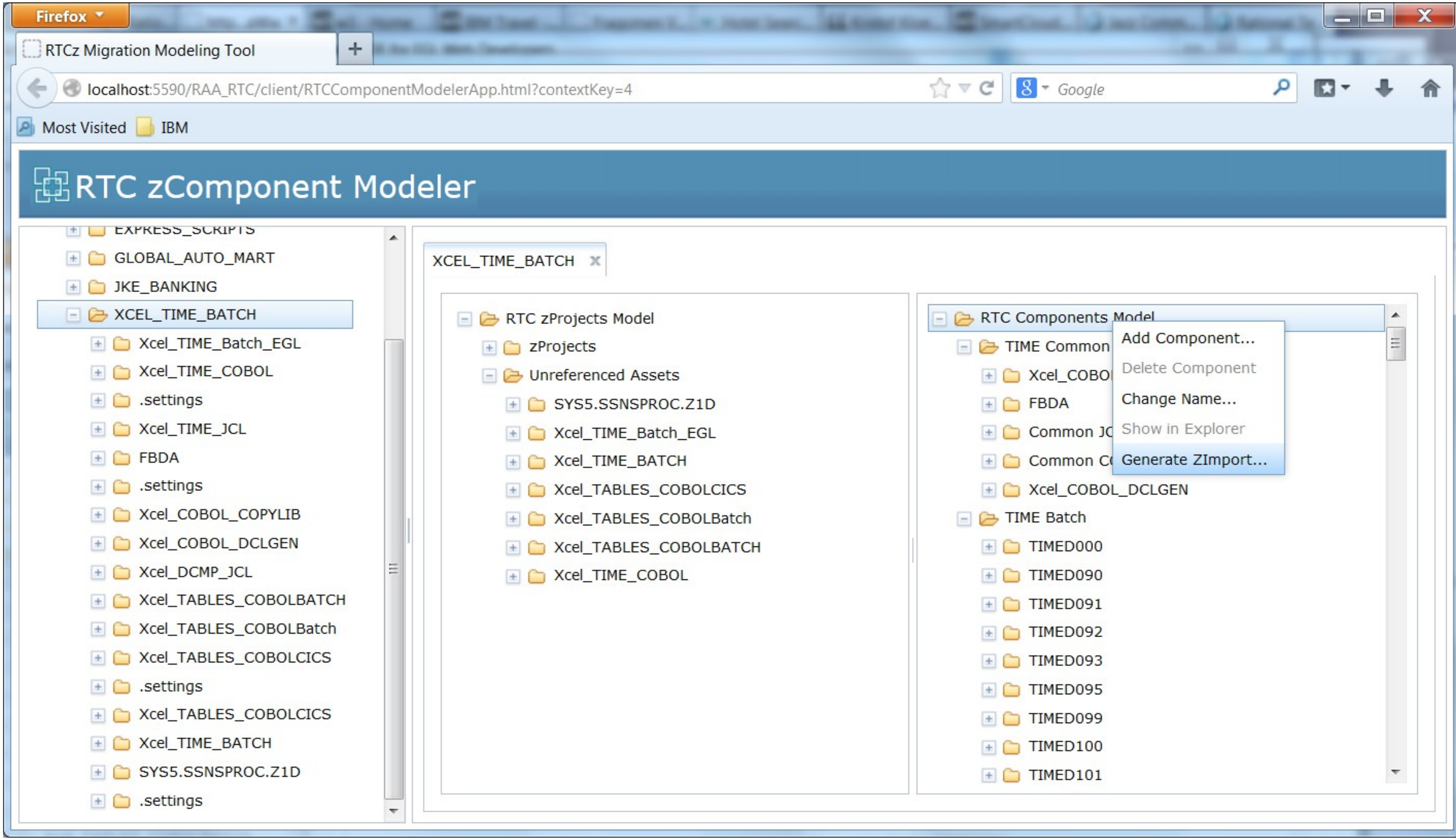
Xcel_COBOL_DCLGEN

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval

Going a step further …



Going a step further …



RTC zComponent Modeler

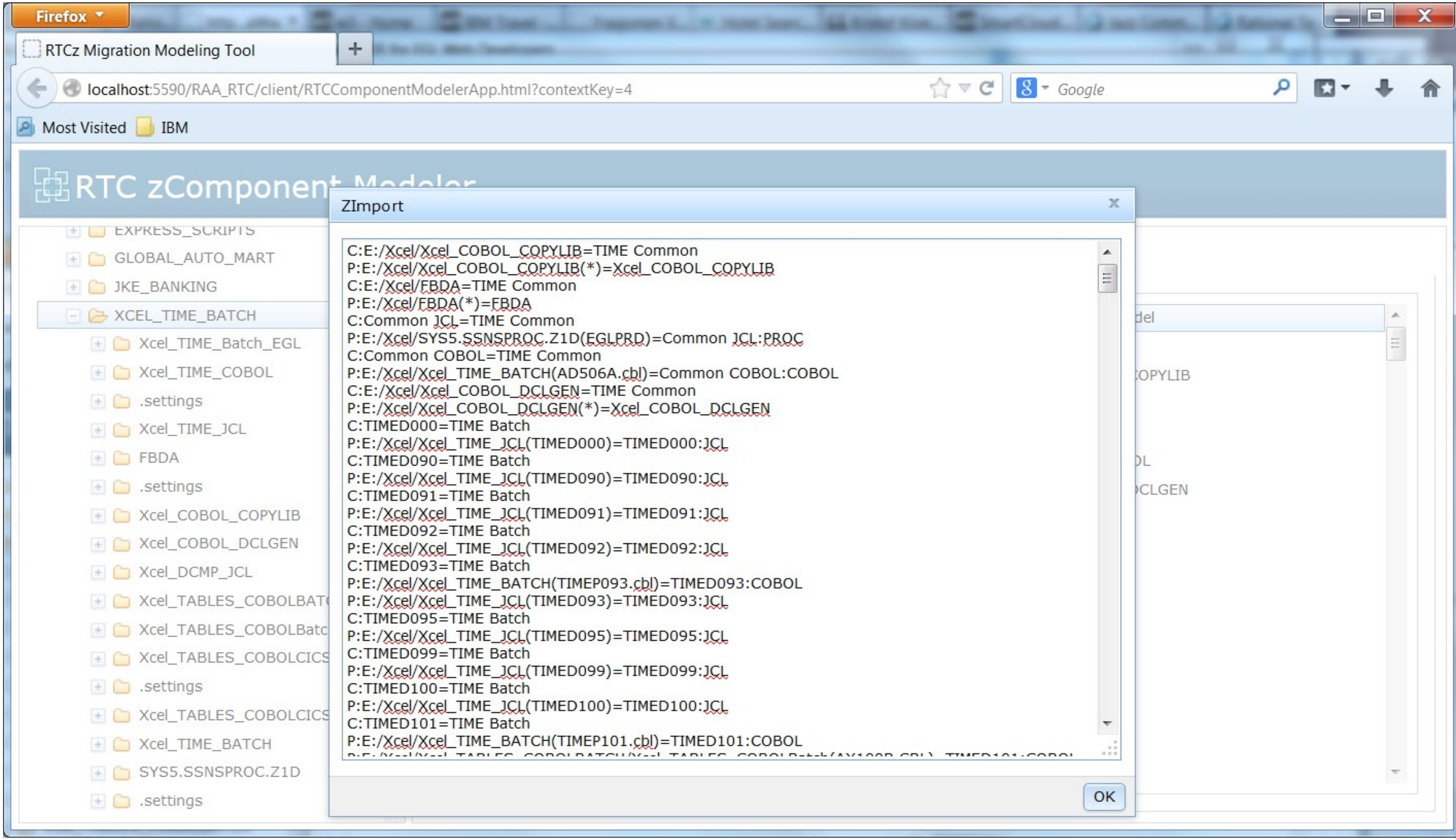
- EXPRESS_SCRIPTS
- GLOBAL_AUTO_MART
- JKE_BANKING
- XCEL_TIME_BATCH
  - Xcel_TIME_Batch_EGL
  - Xcel_TIME_COBOL
  - .settings
  - Xcel_TIME_JCL
  - FBDA
  - .settings
  - Xcel_COBOL_COPYLIB
  - Xcel_COBOL_DCLGEN
  - Xcel_DCMP_JCL
  - Xcel_TABLES_COBOLBATCH
  - Xcel_TABLES_COBOLBatch
  - Xcel_TABLES_COBOLCICS
  - .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_COBO
    - FBDA
    - Common JC
    - Common C
    - Xcel_COBOL_DCLGEN
  - TIME Batch
    - TIMED000
    - TIMED090
    - TIMED091
    - TIMED092
    - TIMED093
    - TIMED095
    - TIMED099
    - TIMED100
    - TIMED101

Going a step further …



```
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.cbl)=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.cbl)=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.cbl)=TIMED101:COBOL
P:E:/Xcel/Xcel_TABLES_COBOLBATCH/Xcel_TABLES_COBOLBatch(AX100R.CBL)=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
- 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