## talkdata

# Document Attachments & e-Mail Process Links

Insurance Claim Processing Example Showing a Step-by-step Approach to Document Attachment and e-Mail Integration

*Examples illustrated on: TeamWorks Version 6n on JBoss with SQL-Server 2005* 

By

**Gary Samuelson** 

| OVERVIEW                               | 1    |
|--|------|
| ARCHITECTURE                           | 2    |
| PROCESS OVERVIEW                       | 3    |
| BUSINESS PROCESS                       | 4    |
| ORGANIZATION & ROLES                   | 4    |
| BUSINESS MODEL                         | 5    |
| Business Data                          | 6    |
| Services – User Oriented with TW Coach | 6    |
| System Services – Automated            | 7    |
| IMPLEMENTATION DETAILS                 | 8    |
| EMAIL SERVER                           | 8    |
| Installation and Setup                 | 8    |
| EMAIL CLIENT                           | 9    |
| Outlook Express Configuration          | 9    |
| TEAMWORKS EMAIL INTEGRATION            | . 13 |
| TEAMWORKS TIMER EVENTS                 | . 15 |
| HREF Process Link                      | . 16 |
| PARTICIPANT GROUPS                     | . 16 |
| RUNNING THE PROCESS                    | . 17 |
| ABOUT THE AUTHOR                       | . 21 |

## **OVERVIEW**

The key point in this solution is that documents are managed through procedure. It's the business process that defines:

- **Business Context** The business process model defines exactly when, why, and how documents are consumed and processed. This context gives us a vocabulary for describing metadata; so, business terminology then describes business resources (documents). This methodology establishes clarity as we now have business words for what the business has (resources) and does (process).
- Security Process implies usage requirements as in "who" needs to do "what" to our managed documents. What follows then are meaningful definitions (aka: understood) for authentication and authorization in that individuals and groups have their business place, function, and accessibility to resources and sensitive/controlled information.



## ARCHITECTURE

The architecture for this example illustrates integration between three major system components:

- 1) Teamworks Process Server Business model runtime environment and driver.
- 2) Database Shown here as our document storage sub-system.
- 3) Email server Manages email messages between Teamworks and participants. Messages contain HREF links that point to business process. Users access documents through process (via these links) as apposed to opening physically attached binary (actual) documents.



## **PROCESS OVERVIEW**

Claims processing serves as a good example for illustrating document attachments, forwarding, and email generation.

A/A1) The Customer or Customer Service Representative (CSR) fills out the necessary claim forms and attaches required pictures, police reports, etc..

**B**) Clerk's office receives the claim and associated attachments. Clerk reviews and certifies the claim before handing off to the Adjuster's office. The associated attachments follow the claim process instance into the Adjuster's task list or "inbox".

**C**) The Adjuster reviews the claim information and attachments during the investigation process. A settlement is negotiated before payment authorization is forwarded to the Accounting office.

**D**) Accounting receives the payment authorization and prints a check for the Insured.



## **BUSINESS PROCESS**

### **Organization & Roles**



Often neglected, though critical to all BPM projects, is our organization. Roles define responsibility which provides a roadmap on what people (roles) do or need within the process. Without this we cannot begin because we honestly don't yet know our process until we understand role-to-task relationships.

#### **Organization**:

• **Claims Office** – Group who is responsible for receiving, validating, processing, and paying claims.

#### Roles:

- **Customer** Policy holder. In relation to the claims office, the Customer is a policy holder requesting payment on insured coverage basically, a person who wants a check to cover insured damages (i.e. money to fix a wrecked car). Listing the Customer as a member of the Claims Office is odd; however (for technical reasons) we list this role so that we correctly assign system access rights as they relate to both application function and managed business/system data.
- **Customer Service Representative (CSR)** This person is responsible for both maintaining customer relations and providing support for customers seeking payments on claims. A majority of customer interaction is managed by the CSR.
- **Claims Clerk** The Claims Clerk is responsible for verifying accuracy and completeness on new insurance claims, documents and attachments.
- Adjuster The Adjuster's role is to investigate claims, negotiate settlements, and authorize payments to claimants.
- **Payments Clerk** This role is responsible for receiving payment authorizations, writing (releasing) checks and mailing them to customers.



#### **Business Data**

This model contains the following data items:

- 1) Customer The Customer type contains three attributes: First name, Last name, and policy number.
- 2) Task ID The Task ID is a system value generated for uniquely identifying task instances. The ID is embedded within the email process-link (href) and directs the user into the correct process instance when "clicked". Task ID value is pulled directly from Teamwork's JavaScript API.
- 3) Document Object Teamworks provides this component in the coach development palette. I simply dragged this into my service-coach designer and configured.

#### Services – User Oriented with TW Coach

#### **Customer Details and Attachments**

This service contains a basic coach with an enclosed Teamworks document object.

|   | 🖲 Claim Processing - Mozilla Firefox   | K |
|---|--|---|
|   | Eile Edit View History Bookmarks Iools Help  |   |
|   | 🕜 💽 - C 🚇 - X 🔏 - 🦑 🏠 ( 🖬 http://otto:8081/ 😭 - 💽 - Google 🖋   | D |
|   | Claim Processing   |   |
|   | Customer Details and Attachments   | Ī |
|   | Customer   |   |
|   | First Name: Bob  |   |
|   | Last Name: Smith   |   |
|   | Policy Number: 1234  |   |
| his section is<br>rovided as a "control"<br>rom the TW Coach<br>Designer palette. | Documents<br>No Documents Attached<br>Upload Document<br>Type File<br>File C:la_soratohlaaa_sample_attach.doc Browse<br>Title Sample Attach<br>OK Cancel |   |
|   | Ok   |   |
|   | Done 💩 🦑   |   |

## **Other Services**

The rest of the services re-use the Customer Details coach except for Accounting, which is almost identical except that the data is "display only" and I changed the "Ok" button text to read "print check".

### System Services – Automated

Both "email Clerk" and "email Adjuster" services follow the same pattern. I hard-coded data values for demonstration purposes.



- Email setup This is a nested service I created to initialize required input parameters for the "Isw SMTP Send Email" integration definition.
- 2) Address email This service sets email address and body content. It also generates a link pointing back to the specific task associated to the runtime instance.
- Lsw SMTP Email This integration definition is included with Teamworks and listed under the System library.

## **IMPLEMENTATION DETAILS**

#### **Email Server**

I recommend Java Email Server (aka JES) as your desktop development email environment. This server is lightweight and easy to both configure and monitor.

http://www.ericdaugherty.com/java/mailserver

#### Installation and Setup

- 1) Download JES to your local hard drive.
- 2) Extract to into a directory

Example: c:\jes-1.6.1

3) Add a user to "user.conf" configuration file

Example: user.bob@mydomain.com=1234

Pattern: user.[user name]@[domain]=[password in plain text]

The password is converted to a hash when the server is started.

I only added one user – Bob. He will play all the roles within our business model. Using one email account simplifies the initial development and testing process since you only require one email client and user-account for monitoring messages.

4) Start the email server

Run "mail.bat" (under \bin directory) to start the server.

I chose not to run this as a service so that I can more easily monitor console messages.

## **Email Client**

I chose MS Outlook Express. This client is free and included on most Windows XP installations.

## **Outlook Express Configuration**

1) Select Tools->Accounts from the top-level menu

The following dialog opens:

Note: the "localhost" mail account shown below is what we will create in this example.

| Internet A          | ccounts   |                                  | ? 🗙   |
|---------------------|---|----------------------------------|---|
| All Mail<br>Account | News Directory Servic<br>Type<br>mail (default) | e<br>Connection<br>Any Available | Add →<br><u> A</u> dd →<br><u> </u> <u> </u> <u></u> |
|                     |   |                                  | <u>S</u> et Order<br>Close  |

2) Click the "Add" button (top right hand corner) and select "mail"

Type in "bob" as the display name and click "next".

| Internet Connectio                                    | on Wizard  | ×     |
|---|--|-------|
| Your Name   |  | ž     |
| When you send e-mail, you<br>Type your name as you wo | ur name will appear in the From field of the outgoing messag<br>udd like it to appear. | je.   |
| <u>D</u> isplay name:                                 | Bob<br>For example: John Smith   |       |
|   |  |       |
|   |  |       |
|   |  |       |
|   | < <u>B</u> ack <u>N</u> ext > Ca   | ancel |

3) Internet E-mail Address

Type in: <u>bob@mydomain.com</u>

NOTE: You MUST use "mydomain.com" unless you want to add an additional "local" domain to your JES email server. See file "mail.conf" for more information.

Click "next"

| Internet Connecti          | on Wizard  | ×     |
|----------------------------|--|-------|
| Internet E-mail Address    |  | ×     |
| Your e-mail address is the | address other people use to send e-mail messages to you. |       |
| <u>E</u> -mail address:    | bob@mydomain.com<br>For example: someone@microsoft.com   |       |
|                            |  |       |
|                            | < <u>B</u> ack <u>N</u> ext > C                          | ancel |

4) E-mail Server Names

Use "localhost" for both incoming and outgoing mail.

Click "next"

| Internet Connection Wizard  |        |
|---|--------|
| E-mail Server Names   | ž.     |
| My incoming mail <u>s</u> erver is a POP3 verver.                   |        |
| Incoming mail (POP3, IMAP or HTTP) server:                          |        |
| An SMTP server is the server that is used for your outgoing e-mail. |        |
| localhost   |        |
|   |        |
| < <u>B</u> ack <u>N</u> ext >                                       | Cancel |

5) Internet Mail Logon

Use the following Account name: bob Password: 1234 Note that this is the same password you added to the "user.conf" file above.

6) Congratulations

Click the "Finish" button when you see the silly "Congratulations" dialog.

| Internet Connection Wizard   |       |
|--|-------|
| Congratulations  | N.    |
| You have successfully entered all of the information required to set up your account.<br>To save these settings, click Finish. |       |
| < <u>B</u> ack Finish C.   | ancel |

7) Send a test message via Outlook Express

Create and send an email addressed to "bob@mydomain.com

| 🗅 test n      | nessag          | e                 |                  |                  |                     |          |                     |                |          |
|---------------|-----------------|-------------------|------------------|------------------|---------------------|----------|---------------------|----------------|----------|
| <u>Eile</u> d | it <u>V</u> iew | <u>I</u> nsert Fi | ormat <u>T</u> o | ools <u>M</u> es | sage <u>H</u> e     | elp      |                     |                | <b>.</b> |
| 📑<br>Send     | Cut             | Copy              | Paste            | <b>У</b><br>Undo | Sy<br>Check         | Spelling | <b>()</b><br>Attach | ↓!<br>Priority | •        |
| 📴 To:         | bob@myc         | lomain.com        |                  |                  |                     |          |                     |                |          |
| 📴 Cc:         |                 |                   |                  |                  |                     |          |                     |                |          |
| Subject:      | test mess       | age               |                  |                  |                     |          |                     |                |          |
| Arial         |                 | <b>v</b> 1        | .0 🔽 I.          | B I              | <u>u</u> <u>A</u> , |          | €                   | ± 1 🔳          |          |
| This is a t   | est             |                   |                  |                  |                     |          |                     |                | <        |
| -             |                 |                   |                  |                  |                     |          |                     |                |          |

8) Receive test message via Outlook Express

Click the "Send/Recv" button. You should receive the message in a few moments. NOTE: JES usually takes a few seconds, roughly 3, to process mail. This value can be adjusted in JES configuration, "mail.conf".

Example of received mail via JES:

| 🛋 test m                          | iessage  |         |                |
|-----------------------------------|--|---------|----------------|
| <u>Eile</u> dit                   | t ⊻iew <u>T</u> ools <u>M</u> essage <u>H</u> elp                |         | 27             |
| See Reply                         | Reply All Forward Print Delete Previou                           | Js Next | M<br>Addresses |
| From:<br>Date:<br>To:<br>Subject: | bob<br>Friday, December 12, 2008 11:54 AM<br>bob<br>test message |         |                |
| This is a te                      | est  |         | <              |
|                                   |  |         | <u> </u>       |
|                                   |  |         | .:!            |

## **Teamworks Email Integration**

1) Create a test Teamworks service to test email integration.

| Start |  |
|-------|--|
|-------|--|

The "test email" server script contains the following:

Note: I set and initialize all required SMTP adapter values in this script.

```
tw.local.smtpHost = 'localhost';
tw.local.to = 'bob@mydomain.com';
tw.local.from = 'bob@mydomain.com';
tw.local.replyTo = 'bob@mydomain.com';
tw.local.subject = 'test message';
tw.local.contentType = 'text/html';
tw.local.contentType = 'text/html';
tw.local.importance = ''; // doesn't seem to work
tw.local.cc = '';
tw.local.bcc = '';
tw.local.bcc = '';
tw.local.attachmentFileNames = '';
var html = '';
html += ' this is a test message ';
html += ' this is a test message ';
html += '<a href=http://www.google.com>Test link to Google</a>';
tw.local.messageText = html;
```

2) Debug the script to see if it's able to connect to JES and send email

If everything is working correctly you should get the following email, via JES, in Outlook Express:

| 🛎 test m                              | nessage  |          |
|---------------------------------------|--|----------|
| <u>Eile</u> dit                       | it <u>V</u> iew <u>T</u> ools <u>M</u> essage <u>H</u> elp                                 | <b>*</b> |
| See Reply                             | Reply All Forward Print Delete Previous  | *        |
| From:<br>Date:<br>To:<br>Subject:     | bob@mydomain.com<br>Friday, December 12, 2008 12:20 PM<br>bob@mydomain.com<br>test message |          |
| this is a te:<br><u>Test link t</u> e | st message<br>10 Google  | <        |
|                                       |  |          |

## **Teamworks Timer Events**

Since we "fire" the timer event immediately after the task starts, this Teamworks (TW) timer event is essentially used to "split" or "fork" the token down an additional path towards the email creation activity. What's special though about its usage is that we are able to grab the task's instance ID the moment it's created. In this way we are able to use HREF links to reference task instances.



#### **Timer Event Implementation Example**

It's important that "close attached activity" is NOT Checked so that the Investigate and Authorize task (itself) is allowed to run.

| 🗑 Attached Event Deta                   | ils                 |            |
|---|---------------------|------------|
| Close Attached Activity:<br>Repeatable: |                     |            |
| <ul> <li>Timer Properties</li> </ul>    |                     |            |
| Trigger On:                             | After Start of Step | *          |
| Custom Date:                            |                     | <b>B</b> . |
| Before/After Difference:                | 0                   | 🔩 Hours 🖌  |
| Tolerance Interval:                     | 0                   | 🔍 Hours 🔽  |
| Use Business Calendar:                  |                     |            |
|   |                     |            |
| Step Simulation Implement               | ntation Pre & Post  |            |

The "pre assignments" are used to catch the system's taskId. We use this value within the mailed HREF link to direct the user into the correct process instance.

| 🔳 Properties 🗙                                  | ▽ □ □                     |
|---|---------------------------|
| 🖻 Pre Assignments                               | +                         |
| tw.local.taskID 🔍 🔍 🗄                           | tw.system.step.taskId 🔍 🍇 |
|   |                           |
|   |                           |
| ▼ Post Assignments                              | +                         |
|   |                           |
|   |                           |
|   |                           |
| Step   Simulation   Implementation   Pre & Post |                           |
|   |                           |

## **HREF Process Link**

The HREF link for a task instance follows this pattern:

http://[server]:[port]/teamworks/process.lsw?zWorkflowState=1&zTaskId=[taskID]

Example:

http://localhost:8081/teamworks/process.lsw?zWorkflowState=1&zTaskId=152

This link can be used as follows:

```
var html = '';
html += ' Claim is ready for review ';
html += '<br/><br/>';
html += '<a
href=http://localhost:8081/teamworks/process.lsw?zWorkflowState=1&zTa
skId='+tw.local.taskID+'>Link to Claim</a>';
tw.local.messageText = html;
```

#### **Participant Groups**

Don't forget to create participant groups per each of the roles defined (above) in the organization. I created four groups:

- 1) Adjuster
- 2) ClaimsClerk
- 3) CSR
- 4) PaymentsClerk

## **RUNNING THE PROCESS**

NOTE: The entire example can be found at Lombardi's Teamworks Wiki. See article, "Document Attachments and email with HTML process HREF links".

- 1) Start the business process definition (BPD) within Teamworks Process Inspector.
- 2) The process waits on Customer Details and Attachments user task.



3) Start the Customer Details and Attachments task. The following browser window should open:

(Note: you might be prompted with a login request prior to seeing this web-page)

| 🖲 Claim Processing - Mozilla Firefox                           |          |
|--|----------|
| Eile Edit Yiew History Bookmarks Tools Help                    |          |
| < 🗩 - 🗶 🥠 - 🗶 🦂 🏕 🏠 (🖬 http://otto:8081/teamworks/f. 😭 - 💽 Goo | igle 🔎   |
| Claim Processing 🛛 🖓 View Process H                            | lelp   🎯 |
| Customer Details and Attachments                               |          |
| Customer   |          |
| First Name: Bob  |          |
| Last Name: Smith   |          |
| Policy Number: 1234  |          |
| Documents  |          |
| No Documents Attached  |          |
| Add Document   |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
| Done   | ۰ 🖈 🙂    |

4) Select a sample document as an attachment and click "OK" in the upload document section.

| 🖲 Claim Processing - Mozilla Firefox   |          |
|--|----------|
| Eile Edit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp |          |
| < 🔊 - 🗶 🦧 - 🧩 🏠 (🖬 http://otto:8081/teamworks/f. 🕁 - 💽 - Google                      | P        |
| Claim Processing   | <b>)</b> |
| Customer Details and Attachments   |          |
| Customer   | _        |
| First Name: Bob  |          |
| Last Name: Smith   |          |
| Policy Number: 1234  |          |
| _ Documents  | $\neg$   |
| No Documents Attached  |          |
| Upload Document-   | ٦        |
| Type File 💌  |          |
| File C:\a_soratoh\aaa_sample_attaoh.doo Browse                                       |          |
| Title sample_attach  |          |
| OK Cancel  |          |
|  | -        |
|  | )k       |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
|  |          |
| Done   | *        |

5) Click "OK" to complete CSR's "attach document" task.



6) Check your Outlook Express email client. You should see the Teamworks generated email message



7) Click the link to launch the referenced task. This link should take us into the Clerk's "Review and Certify Claim" task. Since I re-used the same service/coach for CSR, Clerk, and Adjuster tasks – this form will look exactly the same as the one seen for CSR (above).



8) Run the rest of the BPD and tasks through to completion. The other tasks simply repeat the above...

## **ABOUT THE AUTHOR**

Gary Samuelson is the Director of Systems Architecture at TalkData Consulting. Somewhat a jack-of-all-trades with over 19 years experience in information technologies, he has turned his focus on BPM - the new catchphrase for integrated business/operations support systems and ERP (aka BSS/OSS). In addition to his BPM consulting work he spends his free time writing on technology and building custom desktop and server machines for use in video post-production and computer game development.

His website:

http://Garysamuelson.com