BizTalk + SharePoint: 1+1=3

Integration Best Practices

Technical White Paper

Table of Contents

Audience

This white paper is intended for architects, consultants, developers and knowledge workers who know about BizTalk Server and/or SharePoint and would like to know more about the benefits of integrating these two great Microsoft products.

For managers who just want to get an idea of what this paper is all about and find out how this can apply to current company operations, reading the Management Summary (next page) will suffice.

For people who already know *everything* about BizTalk, just skip the BizTalk Server Overview chapter. Perhaps the best practices chapter is still a good read.

People who already know all SharePoint *ins-and-outs* can safely skip the SharePoint Overview chapter. The best practices chapter is probably still a good read though.

This white paper has been written based on the latest version of each Microsoft product, including:

- BizTalk Server 2009
- SharePoint 2007
- System Center Operations Manager 2007 R2
- Windows Server 2008
- SQL Server 2008

About the authors

Gijsbert "Gijs" in 't Veld is the CTO & Principal consultant at motion10.
Gijsbert is a three-time BizTalk Server MVP (Most Valuable Professional) awardee.
He blogs about his BizTalk experiences at: http://www.indafield.com/BizTalkNotes and can be contacted at gijsbert.intveld@motion10.com.

Wesley Bakker is a lead SharePoint developer at motion10.
Wesley is a SharePoint MCT (Microsoft Certified Trainer).
He blogs about his SharePoint experiences at: http://weblogs.asp.net/wesleybakker and can be contacted at wesley.bakker@motion10.com.

Management Summary

Microsoft's BizTalk Server is a product used by more than 8000 companies world-wide for addressing the various application and business partner integration needs. It is a well known product for building SOA architectures.

SharePoint – Microsoft's fastest growing product - is well known for its document management, collaboration, workflow, portal and search capabilities.

The integration of the two products actually brings a whole new set of capabilities to end-users. SharePoint is able to give BizTalk Server a "face" and provide human work-flow features and dashboard functionality. BizTalk + SharePoint: 1+1=3.

This white paper describes in detail what results can be achieved by combining the two products and how to configure that. It also includes best practices with regard to setting up an architecture with these products.

This is what is possible:

A typical process flow would be: Receive Purchase Orders via different communication protocols and using different standards and formats (EDI, XML, Flat File, etc.) in BizTalk Server. The purchase orders are translated by a BizTalk map to a canonical format, the canonical purchase order is handled by a BizTalk Orchestration and a rule in the BizTalk Rule Engine determines that manual intervention by a manager is needed (for example, if total order value > $100,000.-). This means that a review request using the same XML schema, but with a reference to an InfoPath template will be handed off to SharePoint through BizTalk's SharePoint Adapter. SharePoint can be configured to automatically start a human workflow based on this event, adding a task entry to the manager's Outlook inbox. As soon as the manager opens the task, this will open the review request in a form rendered by SharePoint's Forms Services and the approve or deny button can be clicked. This will feed the request back to the waiting BizTalk business process instance again which will handle it to completion. The manager's task is completed for this order. During processing of transactions, BizTalk feeds the BAM database, where SQL analysis services takes care of rendering this information into understandable Business Activity information shown in the SharePoint portal through Excel Services. Meanwhile, any error or other exception occurring will feed the operators' dashboard in SharePoint with useful information to be able to solve the issues.

BizTalk Server Overview

Introducing BizTalk Server

BizTalk Server is a complete integration platform that can be used to tie together line of business systems (using EAI technologies) as well as connecting business partners through B2B connections. BizTalk Server can sit at the heart of your SOA architecture and can be used to orchestrate services and provide insight in the transactions' flows by means of Business Activity Monitoring (BAM).

How to position BizTalk Server

BizTalk Servers functions at the heart of the integration services environment of an enterprise. It acts as the "middle man" between business partners and the line of business systems, but it also brokers between Line of Business systems.

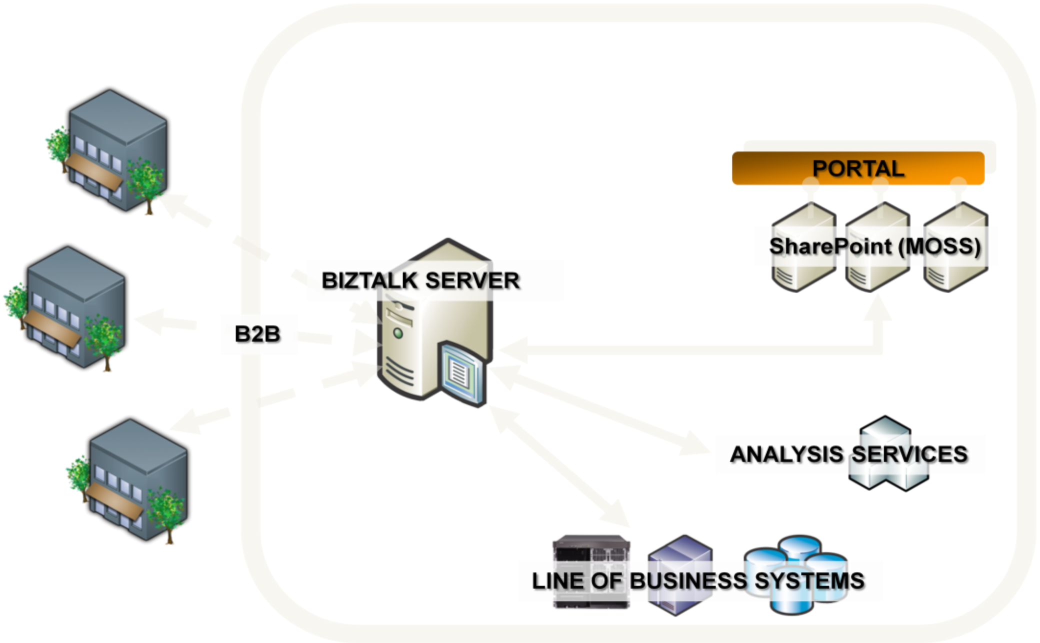


Figure 1

What does BizTalk Server have to offer

BizTalk offers a number of key functionalities that combined serves as an "integration broker" or "business process management" tool. The following lists the most important features.

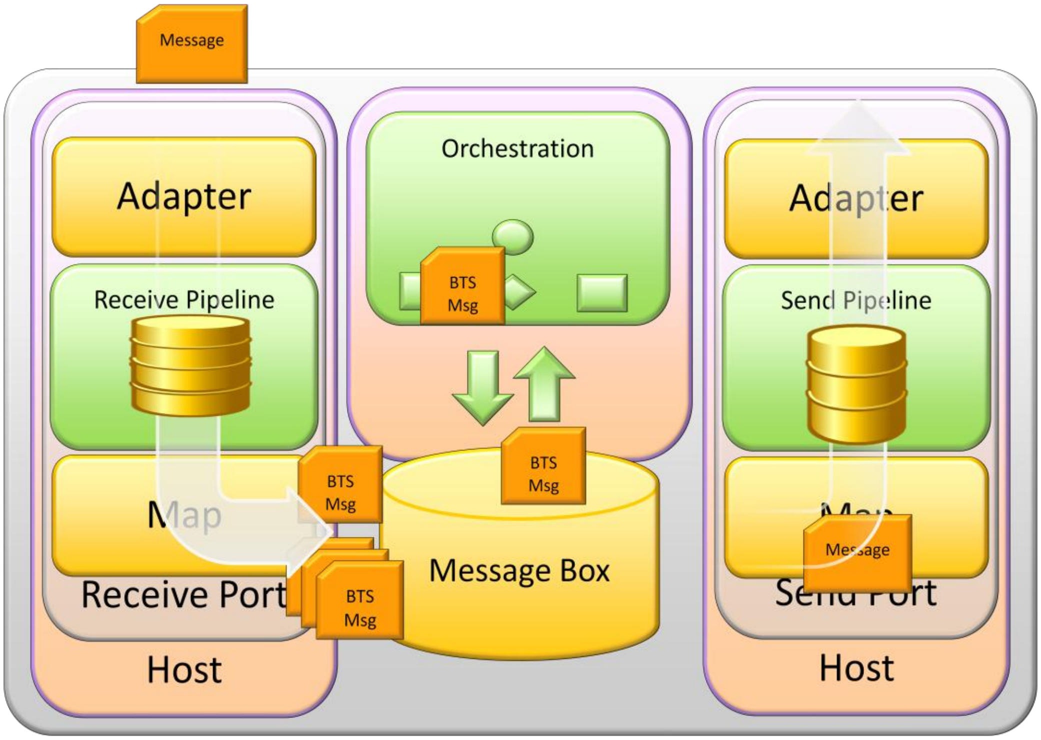


Figure 2

Publish & Subscribe

At the core of BizTalk Service is its powerful publish and subscribe architecture using the Message Box (a SQL Service Database) for persistence. Each message entering the system will, after processing by the receive adapter responsible for the communication protocol handling and the pipeline responsible for decryption, decoding and disassembling, end up in the Message Box along with a set of context properties. One or more subscribers will pick up these messages for further processing. The messages are picked up by subscribing to certain properties of these messages. Subscribers can be Send Ports and orchestrations. The messages always leave the system through Send Ports, which have a pipeline and send adapter configured on them that are responsible for assembling, encoding and encrypting and the transmission according to the configured communication protocol.

Orchestration

Orchestration is responsible for executing business processes within the integration environment. A business analyst can design the business process by using visual shapes. Developers can then provide the necessary coding and administrators can bind the process to physical Send and Receive Ports to facilitate the actual sending and receiving of messages by the process.

Business Rule Engine

Orchestrations are a powerful way of creating business processes without needing to code procedural programs. In orchestrations it is often needed to execute logical paths based on decisions being made. If criteria for making these decisions are not static, the use of the business rule engine makes life easier. The decision criteria are removed from the orchestration and placed in the rule engine, where they can be maintained by a business analyst. The rules are then simply "called" from within the orchestrations.

Transformation

Transformation between different schemas is needed to adjust inbound information to the needs of the destination applications. Information arriving in BizTalk Server can be described by XML schemas and the same applies to the information leaving BizTalk Sever. By using the visual mapper a business analyst or integration specialist can create the transformation rules between these two (or more) schemas that will be executed by the XSLT engine at runtime. Maps can be executed as part of an orchestration, but they can also be configured on the Receive Port (it will then be executed just after pipeline processing) and Send Port (it will be executed just before pipeline processing).

Adapters and Pipelines

Adapters are needed to consume and deliver messages according to the protocols handled by the source and target systems. BizTalk Server comes with a set of adapters such as Line of Business Adapters for SAP, Oracle, Siebel and Dynamics but also database adapters and more generic adapters such as WCF (web services), SMTP/POP3 and FTP. The system also comes with a set of pipelines for handling EDI, Flat Files, XML and other standard types of messages format handling, but developers can also build their own pipeline components and pipelines. The combination of adapter and pipeline makes it possible to connect to almost any system out-of-the-box.

Business Activity Monitoring

During executing of the business processes in BizTalk the system can capture business data (data from events and payloads handled by the orchestrations) and store that in SQL Server OLAP cubes that can be used to present business analysts with information they can use to start understanding the business better and help them improve business processes.

BizTalk Server Best Practices

Introducing Best Practices

There are so many ways BizTalk Server can be used and there are quite some best practices and patterns guides available on the Internet. The best practices that we will focus on in this white paper are related to management and monitoring of BizTalk Server in combination with SharePoint and the other related Microsoft technologies.

Usage of canonical formats

A key design decision that has to be made in every integration project is to use an intermediate format or Canonical Format for every business transaction to be handled by the system. This means that as soon as messages arrive they have to be converted to the canonical format and just before messages are sent out again they have to be converted from the canonical format to the destination format. The use of such a design has many advantages, because the exception handling, business activity monitoring, use of the rule engine and the creation of transformation maps is so much easier and straightforward when using this concept. It also makes it possible to create a more natural separation between development and analyst jobs.

Automated exception handling

BizTalk Server is at the center of the universe in lots of enterprises, which automatically means that lots can go wrong as well. We have seen numerous companies using the system in a very reactive manner, meaning that a group of operators continuously are scanning for errors and handling them accordingly. Using BizTalk Server technology itself and related software from Microsoft can easily create a situation where management by exception can be implemented. Using concepts like Failed Message Routing can help implement such an environment.

Giving BizTalk Server a "face"

BizTalk Server is often used as a black-box and as a result end users typically blame BizTalk for everything that goes wrong. Using technologies like SharePoint, InfoPath and Excel gives end users the capabilities to interact with the black-box and actually give them the insight needed to make better decisions. Human workflow and business activity monitoring, but also more technical monitoring of the system can be achieved rather easily with the combination of products. This white paper show how to best implement such integrations between these great Microsoft server products.

SharePoint Overview

Introducing SharePoint

Microsoft Windows SharePoint Services 3.0 (WSS) is a complete application framework for developing web based collaboration portals that comes with Windows Server 2003 and Windows Server 2008. Most of its strength comes from the huge amount of extensibility points and its flexibility, not to mention the fact that it is built on top of the well known ASP.Net 2.0 Framework and Windows Workflow Foundation (WF).

Microsoft Office SharePoint Services 2007 (MOSS) is an extension on top of WSS. It adds a lot of features, but especially the tight integration with MS Office 2007 makes the installation of MOSS on a collaboration portal worthwhile.

How to position SharePoint

A SharePoint portal can be seen as the central command post for information workers. Information workers can search information, collaborate on information and create information. And all they need is a standards compliant browser. This information can be in the form of records or documents.

What does SharePoint have to offer

SharePoint offers a combination of features which make it valuable inside any organization. We will list just a few to give you some insight.

Search

The SharePoint search indexing service is capable of indexing SharePoint intranet sites, external websites and files system locations. It has the possibility to define scopes and to combine multiple metadata properties into one searchable property. It can even index file content with the help of filters. Information workers do not have to work their ways through large file shares or several websites to get to the information they need any longer.

Collaboration

SharePoint offers collaboration in the form of document libraries. Document libraries are centralized locations where documents of all kinds can be stored and accessed by information workers. Document versioning, workflow, checkout, alerts, it changes the way people look at documents. Information workers do not have to email a multiple of documents waiting for response and then combining all comments manually.

Content Management

Both Web Content Management as Document Content Management is integrated into SharePoint. SharePoint offers the ability to create web page templates and document templates. This enables the information workers to create their own content. Approval workflows give managers a last hand in what actually gets published or not. Information workers no longer need to have their own document templates or come up with their own page design.

There are a lot of ways data can be presented at the end users. One of the key features of SharePoint is that it can deliver this content in a huge variety of views. You can think of:

- Reports
- Dashboards
- Excel Workbooks
- Analytic Views
- Scorecards
- Projects

About SharePoint integration

One of the strongest features of SharePoint is its integration capabilities. There is a seamless integration with the complete Microsoft Office 2007 suite. Clients can publish their Microsoft Office documents to SharePoint libraries right from within the Microsoft Office applications. Task lists agendas in SharePoint can be connected to Microsoft Outlook with a simple click of a button.

SharePoint Best Practices

Introducing Best Practices

SharePoint has a lot of applications and each application has another set of best practices to be applied. SharePoint also has a lot of features and with each feature comes some considerations and best practices. In this white paper we will stick to authentication, authorization and storage recommendations.

Authentication

Authentication in WSS is not really different than with any other ASP.Net web application. You can use both forms and windows authentication and that is it. There are some considerations however to keep in mind when choosing a form of authentication.

While a web browser has no problem at all to show you a login form, other applications such as Microsoft Office Word 2007, Microsoft Office Excel 2007 and Microsoft Office Access 2007 cannot work with forms authentication. Considering that a very large part of the files you collaborate on are Microsoft Office Suite files it would be a very bad choice to use forms authentication for your intranet zone. For an internet zone however forms authentication could be the perfect choice. Although your internet clients will miss the client integration with their Microsoft Office files, you do not have to maintain their accounts in the Active Directory on your windows domain. With forms authentication you can still download the files, update them using your program of choice and upload the files accordingly. You should use Windows Authentication internally and Forms Authentication externally.

Authorization

Use roles if you can. And that advice basically applies to everything Microsoft. Use roles in Microsoft Windows authorization, Microsoft SQL Server authorization and WSS authorization whenever you can. There are situations where you cannot use roles. In a timesheet list you might want everybody in the "HR department" role to have read rights and just one person, the owner of the timesheets in that list folder, to have contributor rights.

One of the great pitfalls you can step into with authorization is to get to granular with it. Although WSS gives you the option to set authorization up to the list item level you do not have a centralized administration of user rights. Things can get very complicated by the time you reach the 10,000 users and 5000 sites with 1000 lists containing 100,000 list items. Stick to roles whenever you can and do not go any more granular than folder level authorization.

Storage

Sites and Site Collections

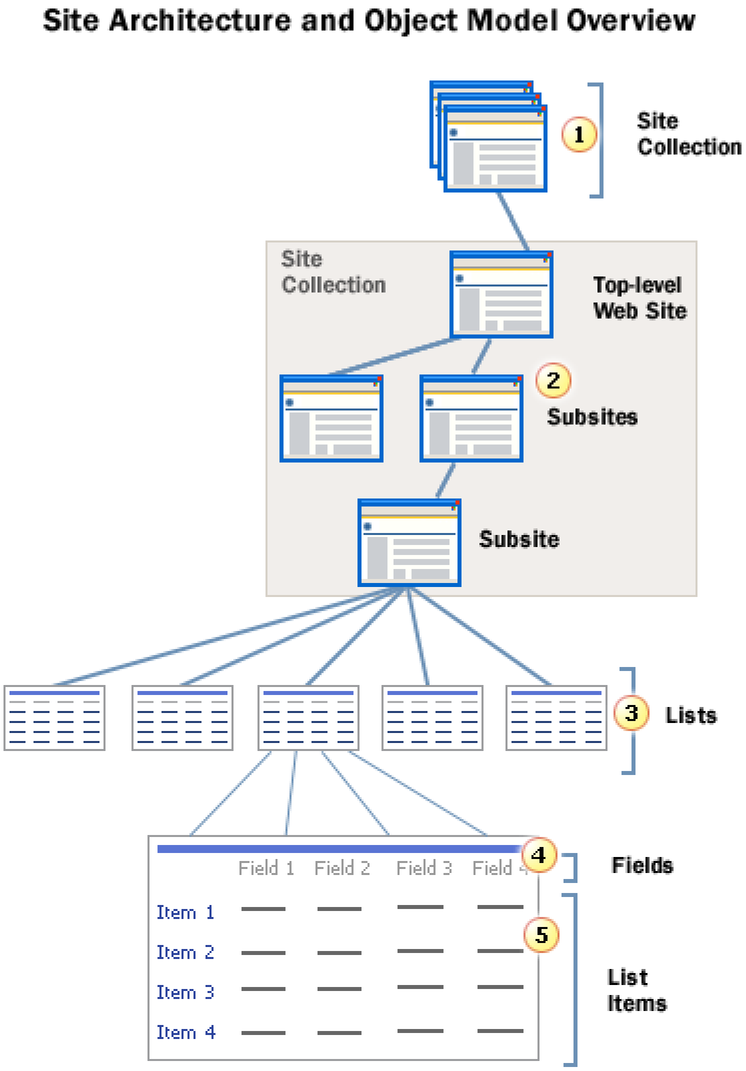


Figure 3

As shown in Figure 3 we can create one Site Collection in a WSS enabled web application and simply add all of our sub sites, with a recommended maximum of 250.000

sites. You must be aware however that WSS uses a so called “content database” for storing the data that is contained within a site and that backup and restore of huge databases can be somewhat cumbersome. To save you from long running backup- and restore procedures WSS supports the distribution of content over multiple content databases. You must note however, that a site collection cannot be divided over multiple content databases.

It is best practice to group your sites into multiple site collections, that each has its own content database, in such a way that each site collection will not grow any larger than 100GB. You can also group multiple smaller site collections to be stored in one content database. It is possible to move sites later on with the “stsadm” command line tool but it is a lot easier to take this limit into account during the architectural fase.

WSS content database usage can be configured on the Central Administration pages. It is good to know that whenever you create a new site collection, WSS will add this collection to the content database which has the greatest difference between the current number of sites and the maximum number of sites. To make sure a single site will not grow out of bounds you can enable quota templates as well to restrict excessive growth.

Lists

SharePoint lists contain all your records and document data and thus built for both flexibility and performance. You must keep certain things in mind when working with lists.

The most commonly used interface to view list items is through the WSS default web interface. Unfortunately the web interface gets sluggish when you try to display more than 2000 items. You can question however if it is ever useful to retrieve more than 2000 items. Better it is to create filtered views to reduce the amount of returned items to anything less than 2000. Another way to solve this “2000 items ceiling” problem is by using folders. Lists can contain folders to group list items and each folder can contain folders again to create a hierarchical tree. In that way you can create a well performing list that contains a total of no less than 5 million list items, by restricting the item count of each folder to 2000 items, so that each subsequent request will retrieve no more than 2000 items.

Three useful integration examples

Introduction

The Integration of BizTalk Server and SharePoint in the way described in this white paper relies on a number of technologies such as BAM, SQL Server Analysis Services, InfoPath, Excel Services and Forms Services. The key to the integration however, is the possibility to exchange messages between the two products by means of the BizTalk WSS (Windows SharePoint Services) Adapter. The WSS Adapter can be used to send and receive messages between BizTalk and SharePoint. Under the covers, this is based on a web services layer added to the SharePoint environment during configuration of BizTalk Server. Architecturally, it looks like this:

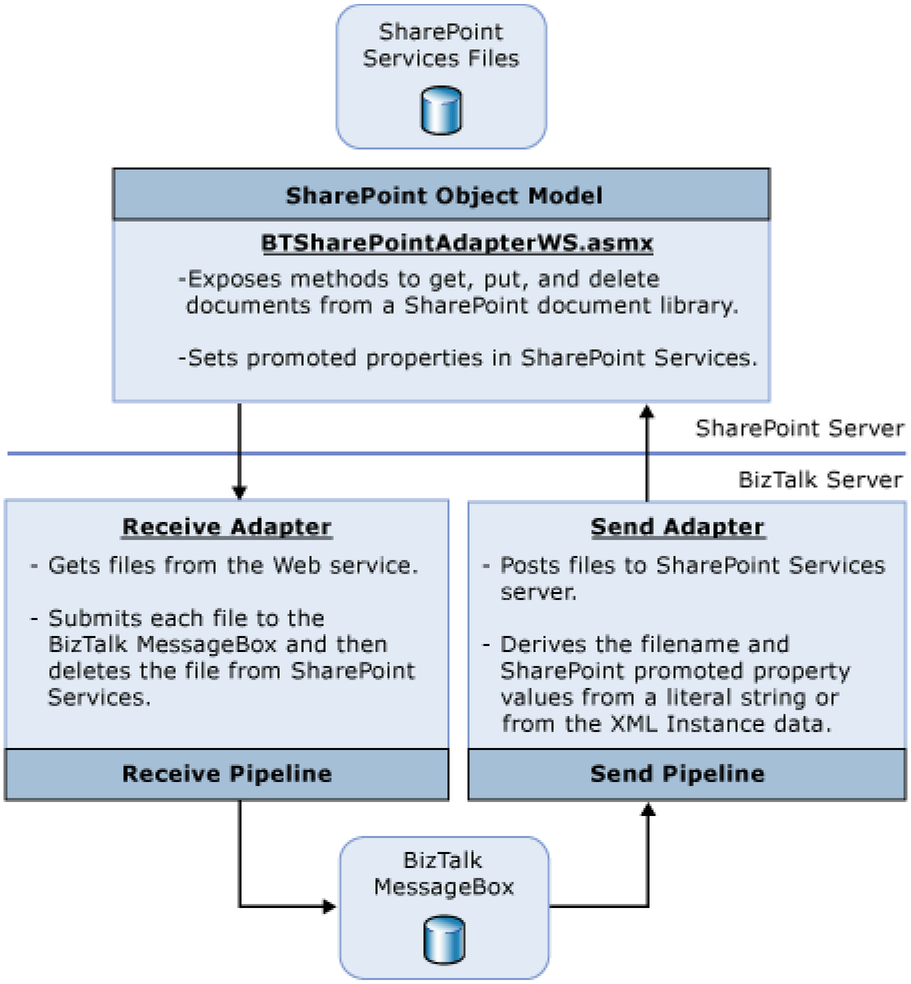
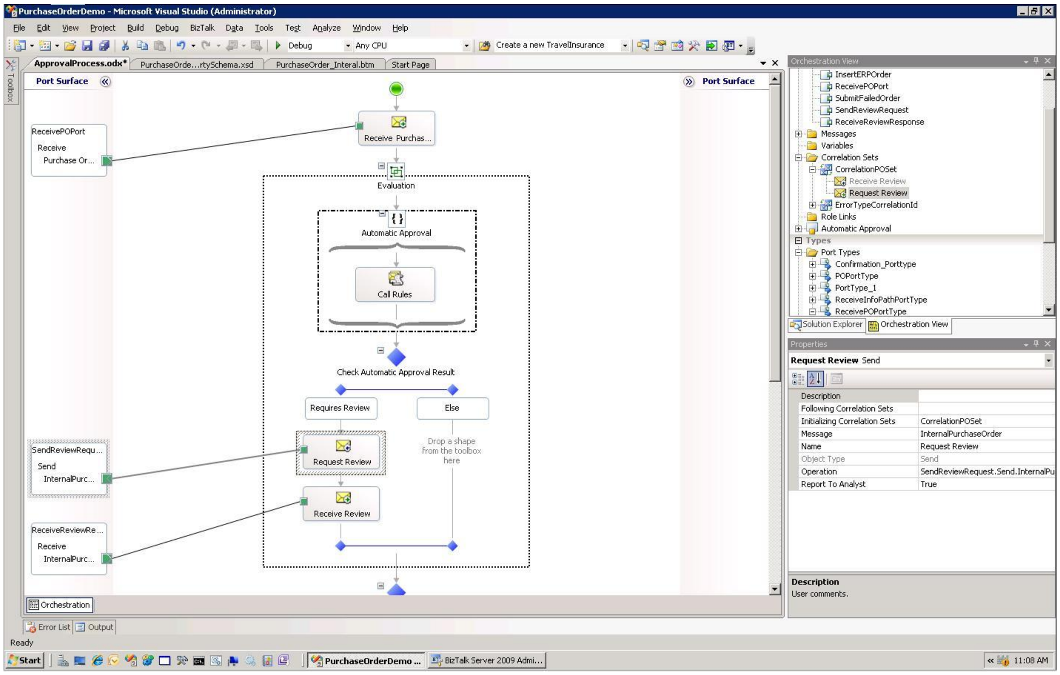

SharePoint Services Files
SharePoint Object Model
BTSharePointAdapterWS.asmx
-Exposes methods to get, put, and delete documents from a SharePoint document library.
-Sets promoted properties in SharePoint Services.
SharePoint Server
BizTalk Server
Receive Adapter
- Gets files from the Web service.
- Submits each file to the BizTalk MessageBox and then deletes the file from SharePoint Services.
Send Adapter
- Posts files to SharePoint Services server.
- Derives the filename and SharePoint promoted property values from a literal string or from the XML Instance data.
Receive Pipeline
Send Pipeline
BizTalk MessageBox

Figure 4

The three integration examples discussed in this chapter are all based on the same “Purchase Order Demo” solution. This solution receives inbound purchase orders in several ways, executes the business logic and hands off the orders to the back-end ERP system. It is a simplified solution but can be used effectively to convey the core propositions of this white paper.

The core of the solution is the Orchestration that handles the purchase orders. Key to the solution is that this orchestration can only handles the internal (canonical) purchase orders. The internal purchase order is based on an XML schema that will be used in other areas of the total solution as well, as described further on.



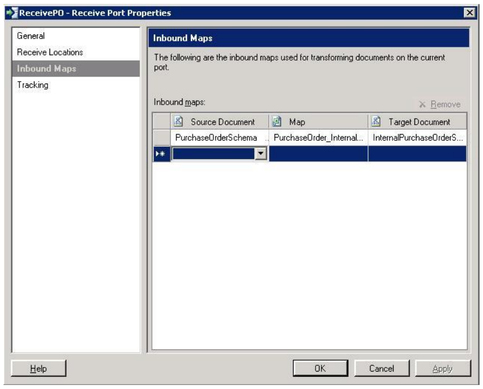
Request Review Send

Description	
Following Correlation Sets	
Initializing Correlation Sets	CorrelationPOSet
Message	InternalPurchaseOrder
Name	Request Review
Object Type	Send
Operation	SendReviewRequest.Send.InternalPu
Report To Analyst	True

Description
User comments.

Figure 5

This means that directly after receiving purchase orders from the outside world, they have to be converted to this internal format. This is done at the Receive Port level. It is common knowledge that you can have multiple Receive Locations per Receive Location. However, not many people know that you can also have multiple Maps configured on a Receive Port. At runtime, the system simply executes the correct Map by selecting the one that has the right source schema describing the inbound message.



Inbound Maps

The following are the inbound maps used for transforming documents on the current port.

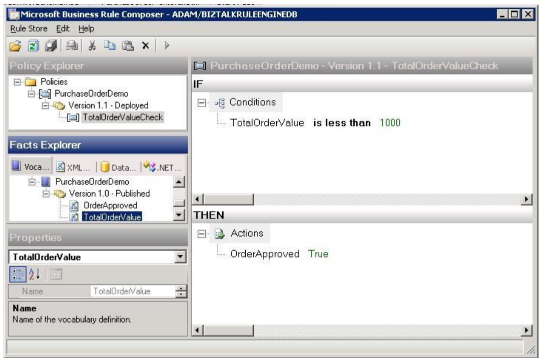
Inbound maps:

Source Document	Map	Target Document
PurchaseOrderSchema	PurchaseOrder_Internal...	InternalPurchaseOrderS...

Figure 6

So, the Receive Pipelines configured on the Receive Locations are responsible for converting the wire format to the XML format (for example EDI or Flat File to XML) and the output from the Pipeline will be fed into the selected Map. The output from the Map will be published in the Message Box.

Once the Orchestration starts processing an inbound order, the first thing that it does is invoke the Rule Engine to determine if manual intervention of the purchase order is needed. In our example, a simple rule on the total order value determines this:



Name: TotalOrderValue

Name
Name of the vocabulary definition.

PurchaseOrderDemo - Version 1.1 - TotalOrderValueCheck

IF
Conditions
TotalOrderValue **is less than** 1000

THEN
Actions
OrderApproved True

Figure 7

Once a purchase order is approved (either automatically or manually) it will be inserted in the back-end ERP system.

This concludes the short intro on the solution used throughout this white paper.
Now, it is time to dive into the three examples.

Exception Handling

Introduction

When BizTalk Server processes messages, many things can go wrong in many places! Protocol violations can occur in transport adapters and during pipeline processing there are many stages in the pipeline that can result in errors, such as: decryption fails (certificate not found or expired?), unpacking the envelope fails, conversion from wire format to XML fails (because the schema cannot be found or validation fails?). Next, orchestration processing can go wrong because rule execution fails, an XSLT map cannot be fully executed or the invocation of a service fails or a SQL lookup cannot find the desired row. And at the send side, many of the same things can go wrong.

By default, BizTalk suspends a message when an error occurs and writes an error record in the application event log. Sometimes these messages are resumable (generally when the context of the message was causing the failure). By using Failed Message Routing (a feature available since the introduction of BizTalk Server 2006) this behavior can be changed. Instead of suspending messages, the failing messages can be subscribed to just like any other message handled by BizTalk. This can be used to feed these messages to your SharePoint portal based on error categories and thus streamline your exception management.

There can of course also be a way back from SharePoint in exception handling patterns. Consider a scenario where messages that fail validation can be “repaired” (through an InfoPath form) and then be re-submitted to BizTalk. This can be set up easily as well through a “message repair orchestration”. This scenario is however not discussed further in this white paper.

BizTalk Configuration

In BizTalk’s administration console you have to turn on Failed Message Routing for each Send Port and Receive Port separately. The setting is called “Enable routing for failed messages”. On the Send Port, this setting has been “hidden” somewhat on the Transport Advanced Options section. On the Receive Port, it is directly available in the General section.

Failed Message Routing

Whenever a message fails during processing by one of the send or Receive Ports that have this setting checked, the following happens:

1. De original promoted properties of the messages will be demoted (this prevents the subscribers to the original message from picking up this message again);
2. New properties regarding the error will be promoted, all in the *ErrorReport* category: *ErrorType* (this will always contain the value “Failed Message”), *FailureCode, FailureCategory*, *Description*, *MessageType*, *ReceivePortName* and *InboundTransportLocation*;
3. The message will be published to the Message Box again;
4. One or more subscribers will pick up the new message.

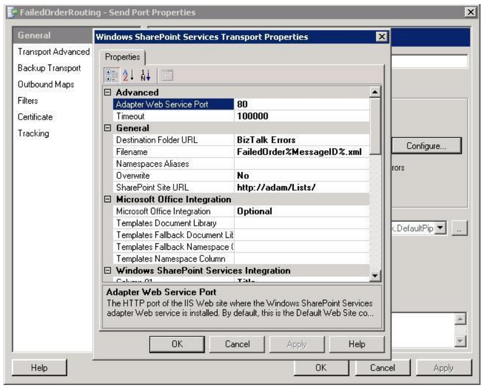
Failed Message Routing can also be used in orchestrations. For this to work a little bit more than just checking some boxes has to be done. Basically you should set all the error properties from within a Message Assignment shape yourself. Example:

```
InternalPOInError = InternalPurchaseOrder;
InternalPOInError(ErrorReport.ErrorType) = "FailedMessage";
InternalPOInError(ErrorReport.FailureCode) = "ORDLT100";
InternalPOInError(ErrorReport.InboundTransportLocation) = "N/A";
InternalPOInError(ErrorReport.ReceivePortName) = "N/A";
InternalPOInError(ErrorReport.Description) = "Total order amount less than $100";
```

Because filtering by the Send Ports or orchestrations for the failed messages will take place on the property *ErrorReport.ErrorType* this property must get promoted. This can be done by creating a new Correlation Type with the property *ErrorReport.ErrorType*. Next is the creation of a Correlation Set with the type just created. On the Send Shape you should indicate that the Correlation Set has to be initialized. The message created should be written directly to the Message Box.

Next, at least one subscription to these failed messages should be created by means of a Send Port or orchestration. Building a “failed messages” orchestration gives far more flexibility in handling the failed messages. But a simple form could be created by configuring one or more Send Ports filtering on the right error properties. Like this, filters can be created that look at the value of *ErrorReport.ErrorCategory* for example.

By specifying the WSS Adapter on the Send Port you can make sure that the failed messages get handed off to the right SharePoint folder:



Advanced	
Adapter Web Service Port	80
Timeout	100000
General	
Destination Folder URL	BizTalk Errors
Filename	FailedOrder%MessageID%.xml
Namespaces Aliases	
Overwrite	No
SharePoint Site URL	http://adam/Lists/
Microsoft Office Integration	
Microsoft Office Integration	Optional
Templates Document Library	
Templates Fallback Document Lit	
Templates Fallback Namespace (	
Templates Namespace Column	
Windows SharePoint Services Integration	

Adapter Web Service Port
The HTTP port of the IIS Web site where the Windows SharePoint Services adapter Web service is installed. By default, this is the Default Web Site co...

Figure 8

By creating more than one Send Port with different filters, you can make sure that the right category of failed messages end up with the right BizTalk operator.

Processing instructions for InfoPath

By using technologies such as InfoPath, the failed message can actually be presented in a nice form within the SharePoint portal. Just specify the right processing instruction for InfoPath in the *XMLAsmProcessingInstructions* on the XMLTransmit Send Pipeline configured on the Send Port. For example:

```
<?mso-infoPathSolution productVersion="12.0.0"
PIVersion="1.0.0.0" href="https://adam/InfoPath
Templates/Forms/ReviewPO.xsn" name="urn:schemas-microsoft-
comffice:infopath:Forms:" solutionVersion="1.0.0.1" ?>.
```

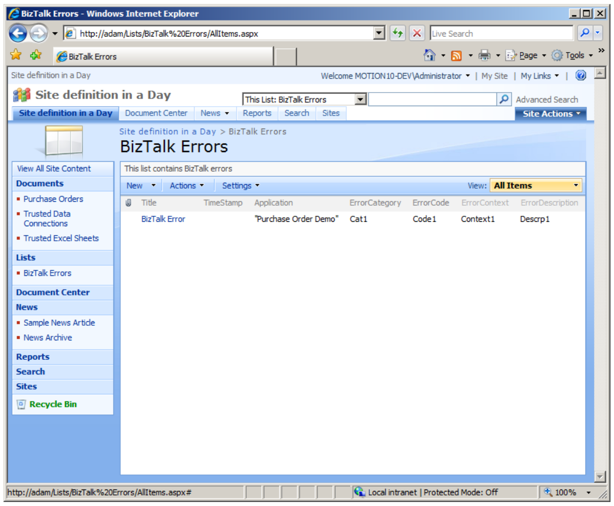
In the next chapter, you will learn how to configure SharePoint to handle and present the failed messages.

SharePoint Configuration

The BizTalk server account (the account under which the BizTalk Service runs) needs a List where it has contributor rights to store the items. If you would like to display the error messages in a more user friendly way you can use a Forms Library as well and XML messages as well. You can design the form template of the Forms Library with Microsoft InfoPath. We will show you how to use XML messages and InfoPath in the 'Manager Approval' example.

Results

The result is a BizTalk Errors list to which we can subscribe to receive email notifications when new items get added. We could also start a Workflow to add a new task to an administrator's task list as soon as a new error is added to the list. It is also possible to have different error lists per BizTalk Error Category, thus enabling the involvement of the right people in your organization responsible for operations in an automated fashion.



BizTalk Errors

This list contains BizTalk errors

Title	TimeStamp	Application	ErrorCategory	ErrorCode	ErrorContext	ErrorDescription
BizTalk Error		"Purchase Order Demo"	Cat1	Code1	Context1	Descrp1

Figure 9

Manager Approval

Introduction

During processing of messages by BizTalk, most of the transactions can and should be handled automatically. It is however possible that in some circumstances a human being has to be involved in the process. Usually this is because of exceptions occurring in the business processes. Examples are:

- An order message has been received with a total order value greater than a certain threshold.
- An invoice is about to be sent to a customer with non-standard payment terms

Typically, these kind of exceptions are based on company rules. This best practice describes an end-to-end example of involving human workflow for manager approval of inbound purchase orders that exceed a certain value. The technologies used are BizTalk, BizTalk's Rule Engine, SharePoint adapter, SharePoint, Outlook integration and lastly InfoPath.

The XML format

The XML format is widely accepted as a standard for the transport of data. Although XML is a so called "human readable" format this does not mean it is a "human friendly" format in ways of editing and reviewing data. There are multiple ways to aid users reviewing XML data. Microsoft Excel 2007 can both read and produce XML data very well, but it does not necessarily help in providing insight into the data and it might be a little to flexible in allowing changes.

Sometimes you want the end users to focus only on those parts of the data that are important and want to make sure they cannot change the XML schema. That is when InfoPath comes around as the one stop solution. InfoPath has both a very familiar user interface for power users and enough advanced options for the developer to create customizations. The tight integration with Microsoft Office SharePoint Services makes InfoPath the tool of choice for working with XML documents.

In this walkthrough we will implement a complete solution for creating and reviewing XML files in SharePoint.

InfoPath

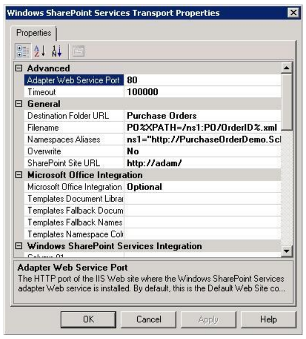
InfoPath displays template based forms to the end user. The user can use these forms to edit XML documents without any knowledge of the particular XML file or schema. Once the user is finished and decides to save the changes, an XML document is created with the correct XML format applied automatically.

If we create a form template and start with a blank form and no data source, we let InfoPath define the XML schema. Unfortunately InfoPath is not very flexible with the namespaces it creates. Your data columns all end up in a "my:" namespace for example which is not necessarily a good thing. It is almost always better to start off with an already existing XML or XSD (XML Schema Definition) file. If we do so, we insure ourselves that the XML documents our form will produce can be processed by other applications that expect that same XML schema.

In our example we will create a form to enable the end user to review a purchase order. This purchase order is generated by BizTalk Server and has a defined XML schema (our canonical or internal purchase order schema) which is also used throughout the BizTalk solution we have built. We will use InfoPath to create a "Purchase Order Review" form based on this schema.

BizTalk Configuration

At the heart of the BizTalk solution is the orchestration that is responsible for handling the inbound purchase orders. After receiving an inbound purchase order the rule engine is invoked to check if manual intervention is needed. If true, a side step is made by handing off the internal purchase order to the SharePoint portal by means of the WSS Adapter.



Windows SharePoint Services Transport Properties

Advanced	
Adapter Web Service Port	80
Timeout	100000
General	
Destination Folder URL	Purchase Orders
Filename	PO%XPATH=/ns1:PO/OrderID%.xml
Namespaces Aliases	ns1="http://PurchaseOrderDemo.Scl
Overwrite	No
SharePoint Site URL	http://adam/
Microsoft Office Integration	
Microsoft Office Integration	Optional
Templates Document Librai	
Templates Fallback Docum	
Templates Fallback Names	
Templates Namespace Col	
Windows SharePoint Services Integration	

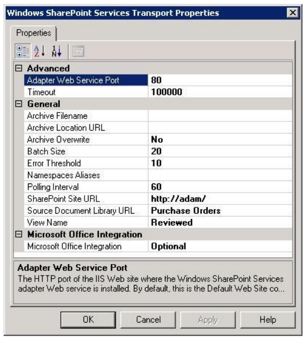
Adapter Web Service Port
The HTTP port of the IIS Web site where the Windows SharePoint Services adapter Web service is installed. By default, this is the Default Web Site co...

Figure 10

By initializing a correlation set (using the order id of the purchase order) on the Send Port in the Orchestration the process can wait for the right reviewed purchase order to come back from SharePoint. This can actually be a long running process, where BizTalk can decide to dehydrate the orchestration for the time being, freeing up resources to other processes. Details of this Orchestration can be seen in the Introduction chapter. The review process in the SharePoint environment actually makes sure that the elements *Reviewed* and *Approved* will be set. Depending on the value of *Approved* the process will continue the approved or denied path in the orchestration.

Note that in the “Windows SharePoint Services Integration” section we have not entered any configuration information. The SharePoint library we are posting this to-be-reviewed-order to has been created by means of an InfoPath template based on the internal purchase order schema, with promoted properties representing the columns in the library. Details will be revealed in the SharePoint configuration section below.

The configuration of the Receive Location that actually will poll for any reviewed purchase orders in SharePoint looks like this:



Advanced	
Adapter Web Service Port	**80**
Timeout	**100000**
General	
Archive Filename	
Archive Location URL	
Archive Overwrite	**No**
Batch Size	**20**
Error Threshold	**10**
Namespaces Aliases	
Polling Interval	**60**
SharePoint Site URL	**http://adam/**
Source Document Library URL	**Purchase Orders**
View Name	**Reviewed**
Microsoft Office Integration	
Microsoft Office Integration	**Optional**

Adapter Web Service Port
The HTTP port of the IIS Web site where the Windows SharePoint Services adapter Web service is installed. By default, this is the Default Web Site co...

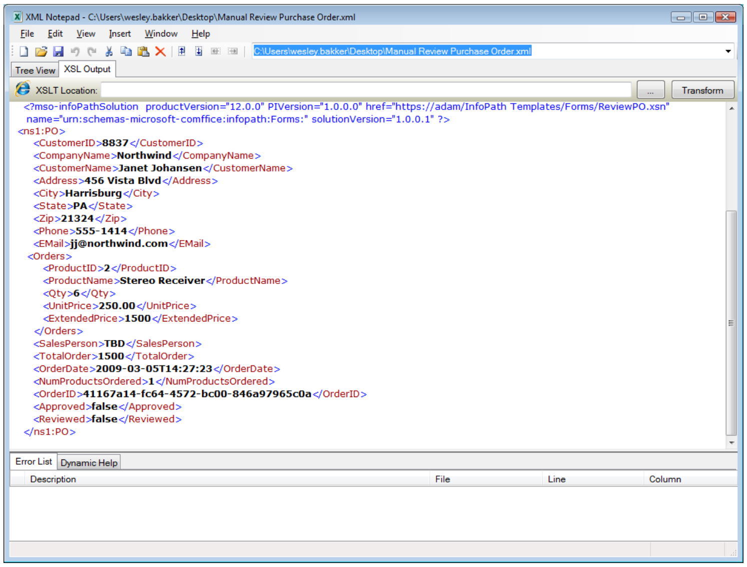
Figure 11

It makes use of a view in SharePoint to only retrieve only the purchase orders that have been reviewed. How this can be configured in SharePoint will be discussed in the next chapters.

SharePoint Configuration

Basic setup

An XML document representing our internal purchase order looks like this:

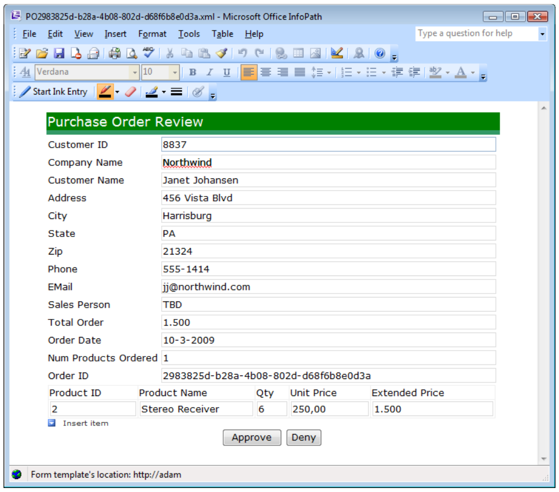

XML Notepad - C:\Users\wesley.bakker\Desktop\Manual Review Purchase Order.xml
File
Edit
View
Insert
Window
Help
C:\Users\wesley.bakker\Desktop\Manual Review Purchase Order.xml
Tree View
XSL Output
XSLT Location:
...
Transform
Error List
Dynamic Help
Description
File
Line
Column

```xml
<?mso-infoPathSolution  productVersion="12.0.0" PIVersion="1.0.0.0" href="https://adam/InfoPath Templates/Forms/ReviewPO.xsn"
 name="urn:schemas-microsoft-comffice:infopath:Forms:" solutionVersion="1.0.0.1" ?>
<ns1:PO>
   <CustomerID>8837</CustomerID>
   <CompanyName>Northwind</CompanyName>
   <CustomerName>Janet Johansen</CustomerName>
   <Address>456 Vista Blvd</Address>
   <City>Harrisburg</City>
   <State>PA</State>
   <Zip>21324</Zip>
   <Phone>555-1414</Phone>
   <EMail>jj@northwind.com</EMail>
  <Orders>
     <ProductID>2</ProductID>
     <ProductName>Stereo Receiver</ProductName>
     <Qty>6</Qty>
     <UnitPrice>250.00</UnitPrice>
     <ExtendedPrice>1500</ExtendedPrice>
   </Orders>
   <SalesPerson>TBD</SalesPerson>
   <TotalOrder>1500</TotalOrder>
   <OrderDate>2009-03-05T14:27:23</OrderDate>
   <NumProductsOrdered>1</NumProductsOrdered>
   <OrderID>41167a14-fc64-4572-bc00-846a97965c0a</OrderID>
   <Approved>false</Approved>
   <Reviewed>false</Reviewed>
 </ns1:PO>
```

Figure 12

This XML is generated for us by our order processing application. Just have a quick look at the XML and notice the "Approved" and "Reviewed" elements at the bottom. Our form will be used to change the values of those two elements.

What we will try to accomplish is that our form will eventually look somewhat like this:



Purchase Order Review

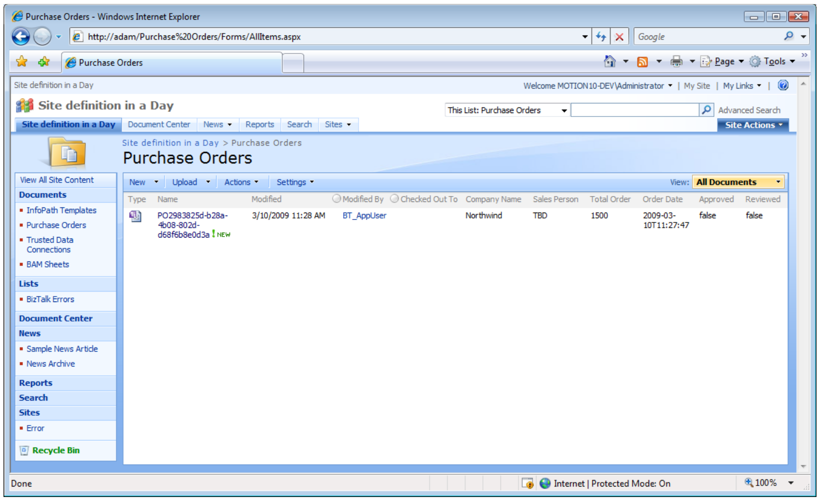
Customer ID	8837
Company Name	Northwind
Customer Name	Janet Johansen
Address	456 Vista Blvd
City	Harrisburg
State	PA
Zip	21324
Phone	555-1414
EMail	jj@northwind.com
Sales Person	TBD
Total Order	1.500
Order Date	10-3-2009
Num Products Ordered	1
Order ID	2983825d-b28a-4b08-802d-d68f6b8e0d3a

Product ID	Product Name	Qty	Unit Price	Extended Price
2	Stereo Receiver	6	250,00	1.500

Figure 13

The values of all the elements in the purchase order message are ready to be reviewed and there are two buttons to approve or deny the purchase order.

We do not want to force our clients to open every purchase order when they would like to know its status. So we need some promoted properties (we will refine the concept of promoted properties further on) in our form which show up in list views. Our list view will eventually look like this:



Type	Name	Modified	Modified By	Checked Out To	Company Name	Sales Person	Total Order	Order Date	Approved	Reviewed
	PO2983825d-b28a-4b08-802d-d68f6b8e0d3a ! NEW	3/10/2009 11:28 AM	BT_AppUser		Northwind	TBD	1500	2009-03-10T11:27:47	false	false

Figure 14

Notice the columns which are extracted from the XML schema and displayed to our client. Our client can use these promoted properties in views and search.

First we need to create a Forms Library on our SharePoint server. Navigate to your site and perform the following steps:

1. Navigate to *Site Actions -> View All Site Content*
2. Click "*Create"*
3. Click "*Form Library"*
4. Type an appropriate name (f.e. "Purchase Orders") and click *"Create"*

This results in a new Form Library in our SharePoint site.

Next, we need to create a new form. Open InfoPath from your start menu.

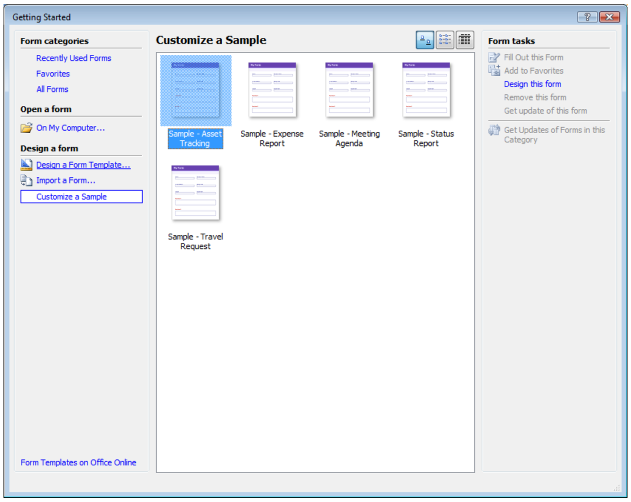
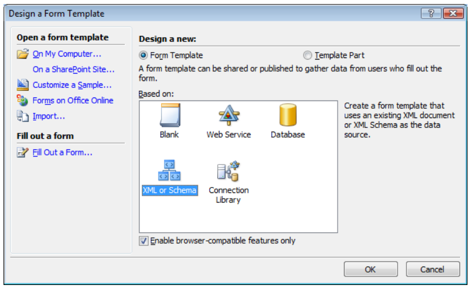


Figure 15

Select *"Design a Form Template…"*



● Form Template ○ Template Part

A form template can be shared or published to gather data from users who fill out the form.

Create a form template that uses an existing XML document or XML Schema as the data source.

☑ Enable browser-compatible features only

Figure 16

Select *"XML or Schema"* and check *"Enable browser-compatible features only"*. Next is to follow the steps of the wizard to end up with a blank form.

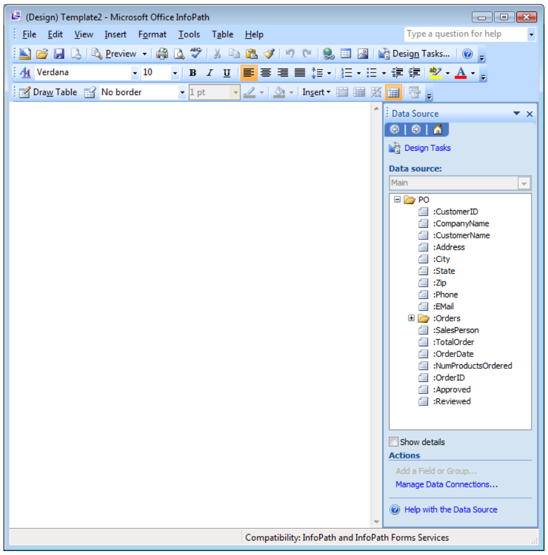
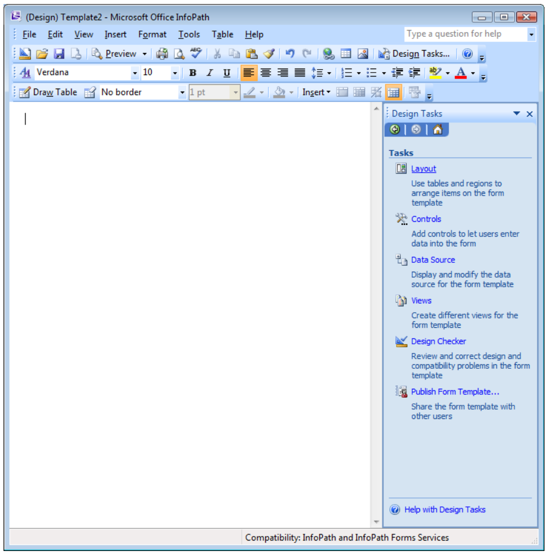


Figure 17

The new form can now be designed. Click *"Design Tasks"* or navigate to *"View -> Design Tasks"*


(Design) Template2 - Microsoft Office InfoPath

Design Tasks

Tasks

Layout
Use tables and regions to arrange items on the form template

Controls
Add controls to let users enter data into the form

Data Source
Display and modify the data source for the form template

Views
Create different views for the form template

Design Checker
Review and correct design and compatibility problems in the form template

Publish Form Template…
Share the form template with other users

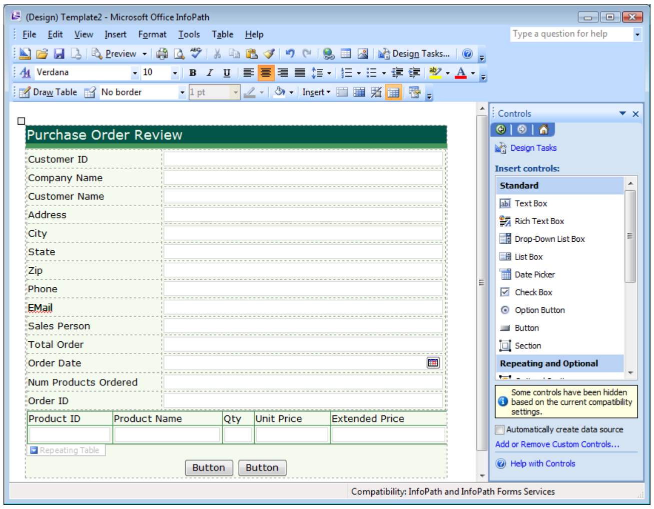
Help with Design Tasks

Compatibility: InfoPath and InfoPath Forms Services

Figure 18

On the *"Design Task"* menu you will find all the steps you need to take to design your form to your likings.

Once you are finished with your layout, you can drag and drop two buttons from the *"Standard"* section in *"Insert controls"* to your design surface.


(Design) Template2 - Microsoft Office InfoPath

Purchase Order Review

Customer ID
Company Name
Customer Name
Address
City
State
Zip
Phone
EMail
Sales Person
Total Order
Order Date
Num Products Ordered
Order ID

Product ID	Product Name	Qty	Unit Price	Extended Price

Repeating Table

Button Button

Controls

Design Tasks

Insert controls:

Standard
Text Box
Rich Text Box
Drop-Down List Box
List Box
Date Picker
Check Box
Option Button
Button
Section
Repeating and Optional

Some controls have been hidden based on the current compatibility settings.

□ Automatically create data source

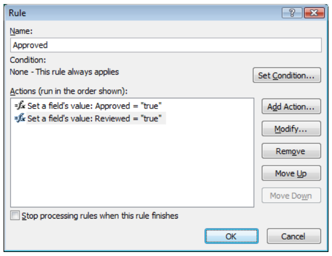
Add or Remove Custom Controls…

Help with Controls

Compatibility: InfoPath and InfoPath Forms Services

Figure 19

You can modify the rules that are performed when a user clicks the button.
Add two actions which change the field values of *"Approved"* and *"Reviewed"* use the appropriate values for the button selected.



Rule

Name: Approved

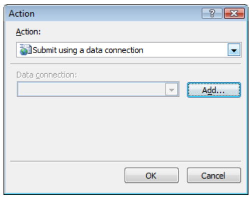
Condition:
None - This rule always applies

Actions (run in the order shown):
- Set a field's value: Approved = "true"
- Set a field's value: Reviewed = "true"

□ Stop processing rules when this rule finishes

Figure 20

Add an action which submits our XML to our SharePoint Forms Library. Use the *"Submit using a data connection"* action and follow the wizard to create your connection.

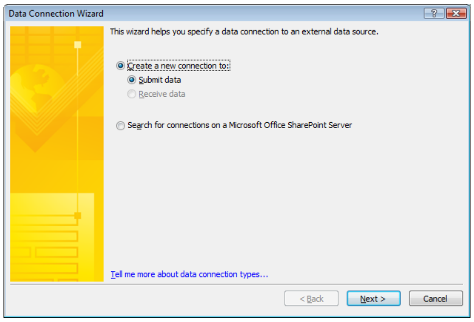


Action

Action: Submit using a data connection

Data connection:

Figure 21



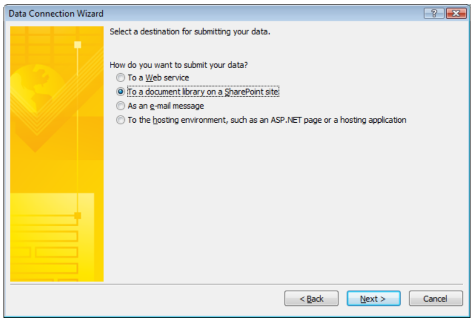
Data Connection Wizard

This wizard helps you specify a data connection to an external data source.

● Create a new connection to:
  ● Submit data
  ○ Receive data

○ Search for connections on a Microsoft Office SharePoint Server

Tell me more about data connection types...



Data Connection Wizard

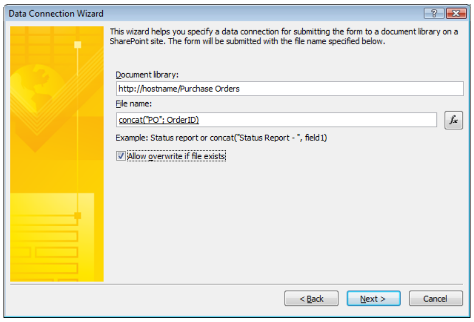
Select a destination for submitting your data.

How do you want to submit your data?
○ To a Web service
● To a document library on a SharePoint site
○ As an e-mail message
○ To the hosting environment, such as an ASP.NET page or a hosting application

Figure 22

Figure 23

Type the URL to our Forms Library. For the file name we are going to create a function that uses the *OrderId* element. We are also going to check *"Allow overwrite if file exists"* because we want the existing purchase order to be overridden.



Data Connection Wizard

This wizard helps you specify a data connection for submitting the form to a document library on a SharePoint site. The form will be submitted with the file name specified below.

Document library: http://hostname/Purchase Orders

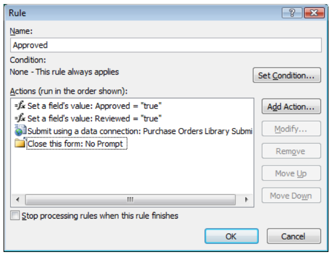
File name: concat("PO"; OrderID)

Example: Status report or concat("Status Report - ", field1)

☑ Allow overwrite if file exists

Figure 24

Once the *"Submit using a data connection"* action is finished we add a last action to close our form.



Name:
Approved

Condition:
None - This rule always applies

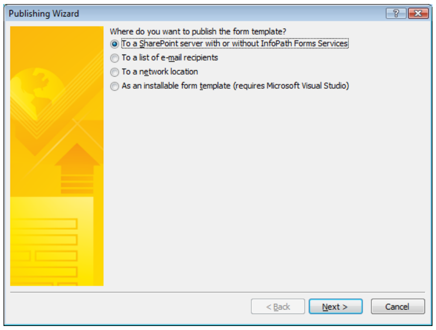
Actions (run in the order shown):
- Set a field's value: Approved = "true"
- Set a field's value: Reviewed = "true"
- Submit using a data connection: Purchase Orders Library Submi
- Close this form: No Prompt

☐ Stop processing rules when this rule finishes

Figure 25

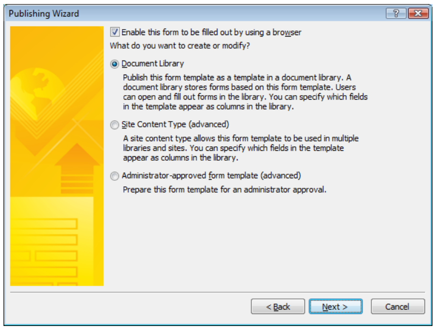
Now that our form is finished we need to publish the form to our Forms library.

Get back to the "Design Tasks" and check the design first with "Design Checker". If the design holds no errors, we follow the "Publish Form Template…" wizard to publish our form template to SharePoint to the point where you can select columns that will be available in SharePoint sites and Outlook folders.



Where do you want to publish the form template?
● To a SharePoint server with or without InfoPath Forms Services
○ To a list of e-mail recipients
○ To a network location
○ As an installable form template (requires Microsoft Visual Studio)

Figure 26



☑ Enable this form to be filled out by using a browser

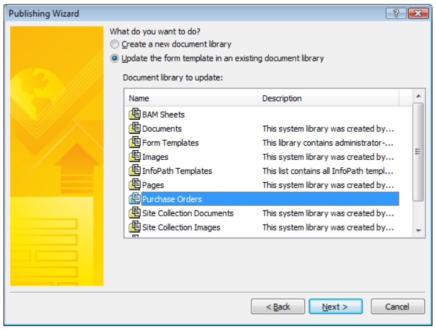
What do you want to create or modify?

● Document Library
Publish this form template as a template in a document library. A document library stores forms based on this form template. Users can open and fill out forms in the library. You can specify which fields in the template appear as columns in the library.

○ Site Content Type (advanced)
A site content type allows this form template to be used in multiple libraries and sites. You can specify which fields in the template appear as columns in the library.

○ Administrator-approved form template (advanced)
Prepare this form template for an administrator approval.

Figure 27



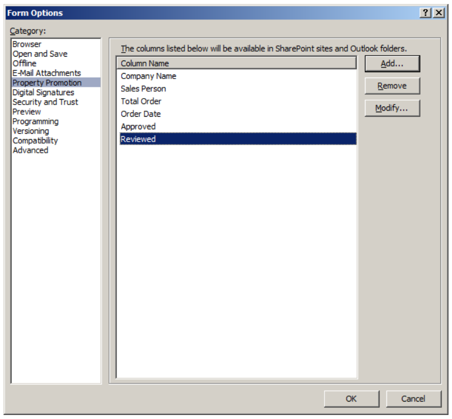
What do you want to do?
○ Create a new document library
● Update the form template in an existing document library

Document library to update:

Name	Description
BAM Sheets	
Documents	This system library was created by...
Form Templates	This library contains administrator-...
Images	This system library was created by...
InfoPath Templates	This list contains all InfoPath templ...
Pages	This system library was created by...
Purchase Orders	
Site Collection Documents	This system library was created by...
Site Collection Images	This system library was created by...

Figure 28

It is very important to note that the columns you "promote" are available in SharePoint sites and Outlook folder. It is even more important that the columns you do not "promote" will not be available. Choose your columns with care. Include columns you would like to add to views or use for search and filtering. Because we need to create a SharePoint view which filters the items based on their 'Review' status we need to add at least that column.



Form Options

Category:

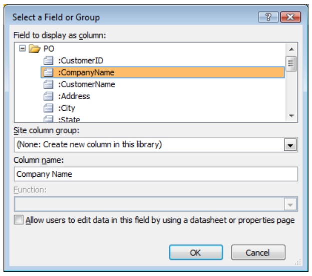
- Browser
- Open and Save
- Offline
- E-Mail Attachments
- Property Promotion
- Digital Signatures
- Security and Trust
- Preview
- Programming
- Versioning
- Compatibility
- Advanced

The columns listed below will be available in SharePoint sites and Outlook folders.

Column Name
Company Name
Sales Person
Total Order
Order Date
Approved
Reviewed

Figure 29

You can change the display name of the column by changing the column name. If all the selected columns are added finish the wizard and you're done.



Select a Field or Group

Field to display as column:

- PO
  - :CustomerID
  - :CompanyName
  - :CustomerName
  - :Address
  - :City
  - :State

Site column group: (None: Create new column in this library)

Column name: Company Name

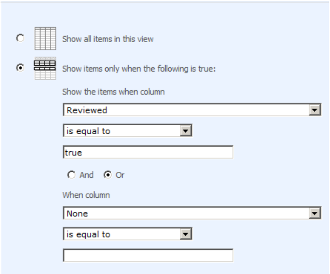
Function:

□ Allow users to edit data in this field by using a datasheet or properties page

Figure 30

Create the view for BizTalk

In order for BizTalk server to retrieve only the reviewed items we have to create a view which returns those items. With the *'Reviewed'* element as promoted property we can create a view on that list as usual and apply a filter on the 'Reviewed' column.



○ Show all items in this view

● Show items only when the following is true:

Show the items when column

Reviewed

is equal to

true

○ And ● Or

When column

None

is equal to

Figure 31

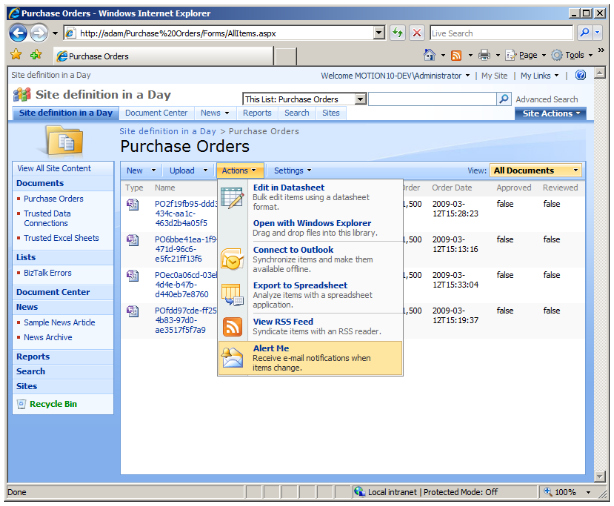
Notifications and Workflows

If you followed all steps above, you have only created the possibility to indeed approve or reject a purchase order. In the end you would really like to be sure that the approvers actually know that there's a purchase order for them to have a look at.
There are two ways to accomplish this:

1. Notifications
2. Tasks

Notifications

Notifications are by far the easiest way for end users to subscribe themselves to change notifications on any SharePoint list or list item. The good news is that users are free to subscribe to any list and list item they want, the bad news is that it does require an action from the end user.



Edit in Datasheet
Bulk edit items using a datasheet format.

Open with Windows Explorer
Drag and drop files into this library.

Connect to Outlook
Synchronize items and make them available offline.

Export to Spreadsheet
Analyze items with a spreadsheet application.

View RSS Feed
Syndicate items with an RSS reader.

Alert Me
Receive e-mail notifications when items change.

Type	Name	rder	Order Date	Approved	Reviewed
	PO2f19fb95-ddd3 434c-aa1c- 463d2b4a05f5	,500	2009-03- 12T15:28:23	false	false
	PO6bbe41ea-1f9 471d-96c6- e5fc21ff13f6	,500	2009-03- 12T15:13:16	false	false
	POec0a06cd-03el 4d4e-b47b- d440eb7e8760	,500	2009-03- 12T15:33:04	false	false
	POfdd97cde-ff25 4b83-97d0- ae3517f5f7a9	,500	2009-03- 12T15:19:37	false	false

Figure 32

Workflows

A lot of people are not aware that they can create custom workflows with SharePoint Designer. This is really unfortunate because SharePoint Designer workflows can fill in the great gap between no workflows at all and the more complex workflows created with Visual Studio. If you combine workflows, notifications and client integration you can really deliver a start to end solution.

A standard collaboration portal site already contains a Task List which alerts the assignees when a new task is created for them. We can use a SharePoint workflow to add task to that task list in very few steps.

Open SharePoint designer and go to *'File > Open Site…'* use the dialog to navigate to your site collection and open your site. We can now add a new workflow with *'File > New… > Workflow…'*

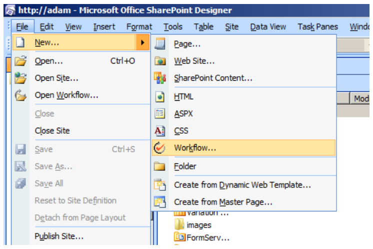
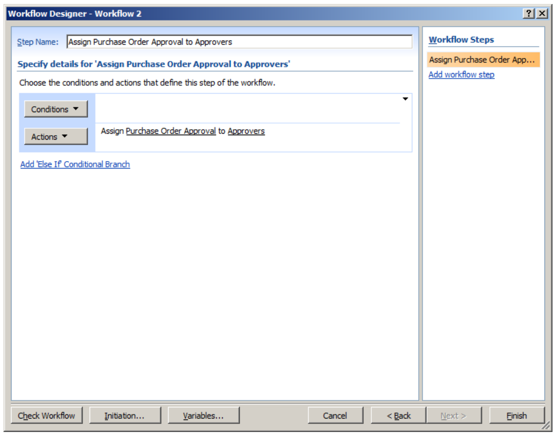


Figure 33

Simply follow the wizard. Attach the workflow to our Purchase Orders list, let it start automatically. From the *'Actions'* dropdown button you can select *'Assign a To-do Item'* and fill in the blanks.


Workflow Designer - Workflow 2

Step Name: Assign Purchase Order Approval to Approvers

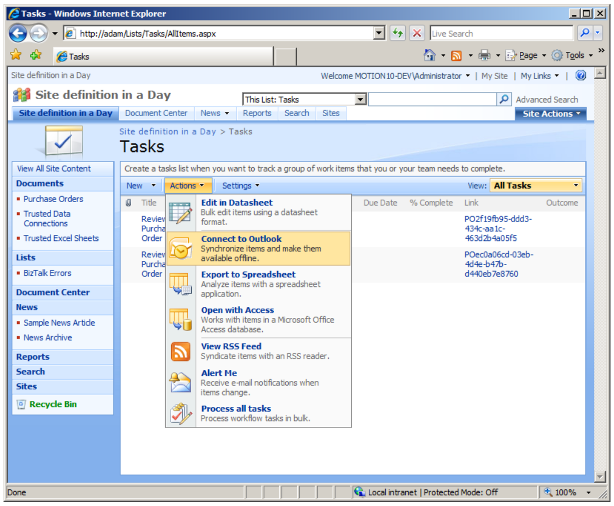
Specify details for 'Assign Purchase Order Approval to Approvers'

Choose the conditions and actions that define this step of the workflow.

Assign Purchase Order Approval to Approvers

Figure 34

We will have a Task generated for all users in the *'Approvers'* role when a new item is created from now on. The approvers will also receive an email alert because they are subscribed to the task list by default. And still we can create an even tighter integration by synchronizing items from the SharePoint task list with Outlook. This is one of those valuable features right there in SharePoint. It is nothing more than a simple click of a button and we have the tasks automatically added to our Outlook task list.



Create a tasks list when you want to track a group of work items that you or your team needs to complete.

Title	Due Date	% Complete	Link	Outcome
Review Purcha Order			PO2f19fb95-ddd3-434c-aa1c-463d2b4a05f5	
Review Purcha Order			POec0a06cd-03eb-4d4e-b47b-d440eb7e8760	

Figure 35

From now on our task will appear in our task list. A very nice detail is that as soon as we now open the Purchase Order with the InfoPath client from our task list within Outlook, the InfoPath client will recognize that we are editing this form as part of a workflow task. We can edit this workflow task right from within the InfoPath client again.

Results

The result is a SharePoint forms library with a published InfoPath form with which we can modify the XML files that are posted to the library. Thanks to promoted properties we can search, filter and view the content of XML Elements right inside SharePoint and Outlook.

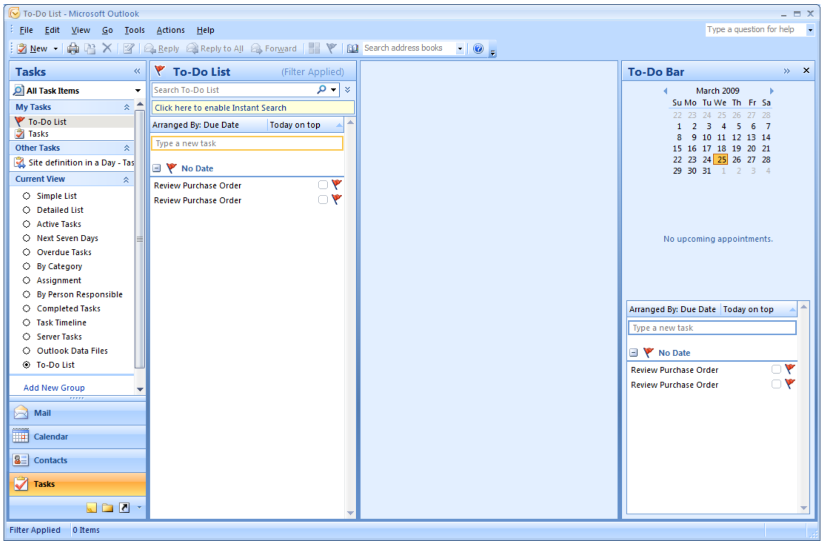
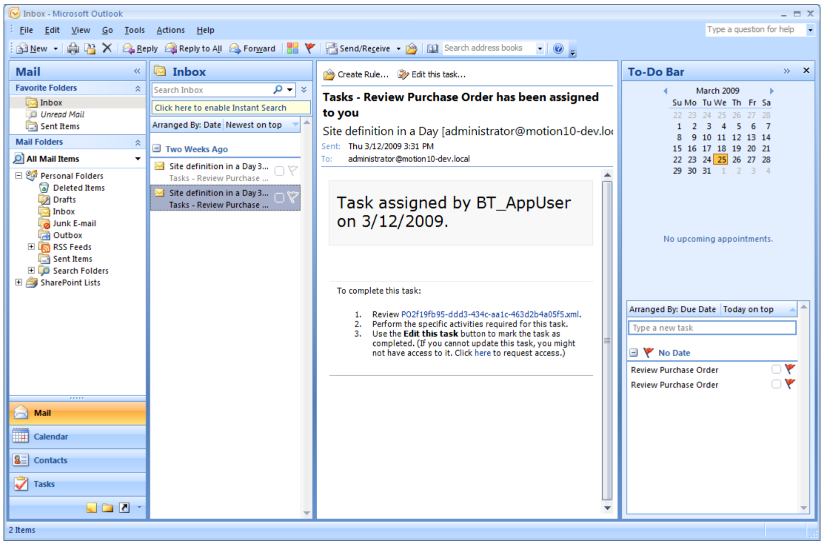


Figure 36

We can add alerts or workflows to the list to notify the end users if new items are added or when existing items are updated.


Inbox - Microsoft Outlook
File Edit View Go Tools Actions Help
Type a question for help
New Reply Reply to All Forward Send/Receive Search address books
Mail
Favorite Folders
Inbox
Unread Mail
Sent Items
Mail Folders
All Mail Items
Personal Folders
Deleted Items
Drafts
Inbox
Junk E-mail
Outbox
RSS Feeds
Sent Items
Search Folders
SharePoint Lists
Mail
Calendar
Contacts
Tasks
Inbox
Search Inbox
Click here to enable Instant Search
Arranged By: Date Newest on top
Two Weeks Ago
Site definition in a Day 3...
Tasks - Review Purchase ...
Site definition in a Day 3...
Tasks - Review Purchase ...
Create Rule... Edit this task...
To-Do Bar
March 2009
No upcoming appointments.
Arranged By: Due Date Today on top
Type a new task
No Date
Review Purchase Order
Review Purchase Order
2 Items

Tasks - Review Purchase Order has been assigned to you

Site definition in a Day [administrator@motion10-dev.loc

Sent: Thu 3/12/2009 3:31 PM

To: administrator@motion10-dev.local

Task assigned by BT_AppUser on 3/12/2009.

To complete this task:

1. Review P02f19fb95-ddd3-434c-aa1c-463d2b4a05f5.xml.
2. Perform the specific activities required for this task.
3. Use the **Edit this task** button to mark the task as completed. (If you cannot update this task, you might not have access to it. Click here to request access.)

Figure 37

BizTalk Dashboard

Introduction

During processing of transactions, BizTalk Server and – optionally – System Center Operations Manager (SCOM) produces a wealth of technical and business level information. Technical information consists of event log records, process states, suspended messages and retry information. Much of the data generated here usually falls in the category “exception handling”. This kind of information is normally handled by BizTalk Operators. Business level information is produced by gathering payload data and event data during the processing by means of the BAM (Business Activity Monitoring) interface. This chapter describes how to present and use the information generated by BizTalk and SCOM in an end-user friendly SharePoint environment.

BizTalk Configuration

Integration with SCOM

In order to capture technical “health” information from any Microsoft Server product, System Center Operations Manager is used. This product can be configured by means of dedicated product management packs to record events occurring in these products. SCOM can also be used to set thresholds and apply rules, and whenever these thresholds are exceeded or the rules are not met, alerts will fire.

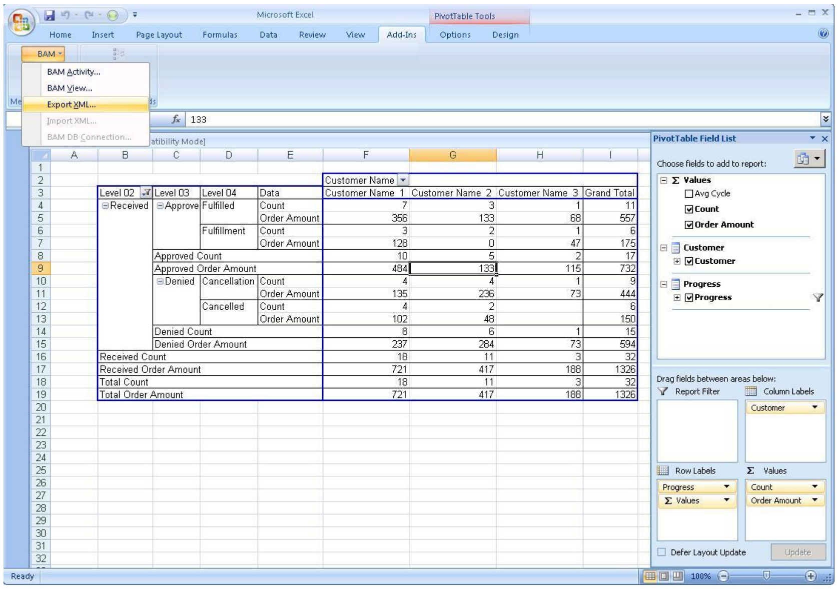
Unfortunately, at the time of writing this white paper, the SCOM 2007 R2 Management Pack for BizTalk Server 2009 had not been released yet (and there was no way we were able to get BizTalk Server 2009 monitored by the 2006 R2 Management Pack and certainly not in combination with SCOM 2007 R2 Beta), so a detailed description of how to configure the management pack for BizTalk Server is not available yet. We will publish an updated version of this white paper as soon as we have a complete working solution with these latest versions.

The information recorded by SCOM for the integration environment (which may include BizTalk Server, but also SQL Server Integration Services and SharePoint itself) will be exposed to the SharePoint portal through SQL Reporting Services.

Configuring BAM

With regard to business level information, you should use Business Activity Monitoring (BAM). There are several ways in BizTalk to capture information with BAM. In this paper, we will describe the easy way using the Tracking Profile Editor. There are other papers describing how you can build your own BAM Interceptors and besides, this paper is about integration with SharePoint and not about how to setup BAM!

Setting up BAM is really not that difficult. The process starts with a business analyst who needs to decide what information to capture during the execution of the business processes. The tool most appealing to a business analyst to configure this is Excel.



				Customer Name			
Level 02	Level 03	Level 04	Data	Customer Name 1	Customer Name 2	Customer Name 3	Grand Total
Received	Approve	Fulfilled	Count	7	3	1	11
			Order Amount	356	133	68	557
		Fulfillment	Count	3	2	1	6
			Order Amount	128	0	47	175
	Approved Count			10	5	2	17
	Approved Order Amount			484	133	115	732
	Denied	Cancellation	Count	4	4	1	9
			Order Amount	135	236	73	444
		Cancelled	Count	4	2		6
			Order Amount	102	48		150
	Denied Count			8	6	1	15
	Denied Order Amount			237	284	73	594
Received Count				18	11	3	32
Received Order Amount				721	417	188	1326
Total Count				18	11	3	32
Total Order Amount				721	417	188	1326

Figure 38

Once the business analyst has created the pivot tables and cubes, this "BAM Workbook" can be saved (a good practice is to also add it to your BizTalk Solution to keep everything nicely together) has to be exported to an XML file. You need to use the BAM add-in for that, so make sure that this add-in has been activated in Excel. It is installed by default when you have indicated to use BAM in the BizTalk Configuration, but activation in Excel has to be done manually. The next step is to use the "bm.exe" tool in the Tracking folder of the BizTalk Program Files installation tree to deploy this workbook as a BAM activity. This will create the necessary activities and views in SQL Server. The command looks like this:

```
bm.exe deploy-all –DefinitionFile:"BAM Workbook.xml"
```

The now deployed BAM Activity Definition can be imported in the BizTalk Tracking Profile Editor. Here, the developer (or analyst) can actually make the link between the BizTalk business process and the BAM workbook created by the analyst using a simple drag-and-drop interface:

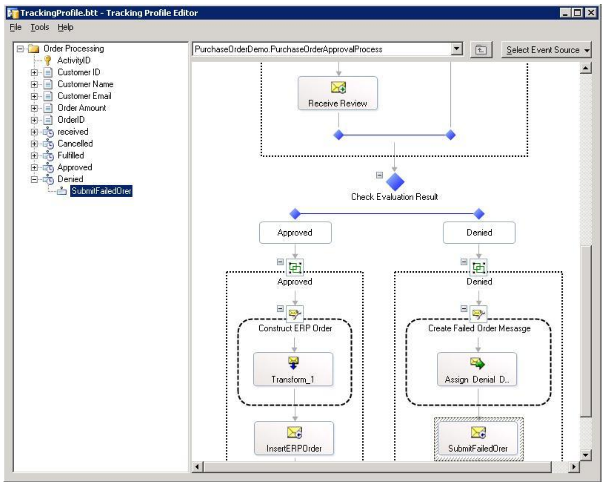


Figure 39

In the above screenshot, you can see that the *Denied* event has been linked to the *Submit Failed Order* shape in the orchestration.

Information to be retrieved from the actually payloads handled by the BizTalk orchestration can be linked to the BAM workbook as well using the same drag-and-drop interface:

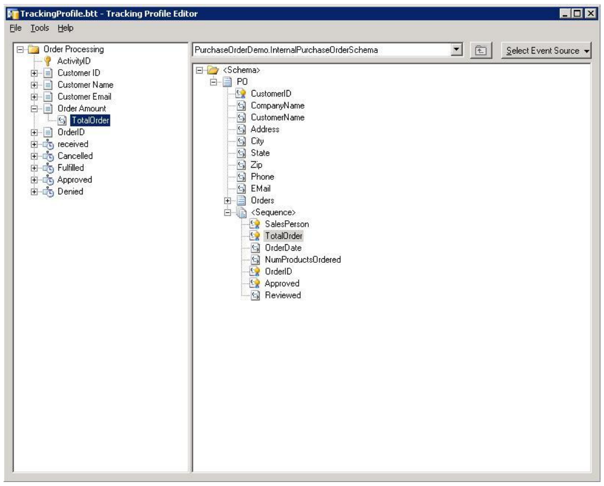


Figure 40

In the above screenshot, you can see that *TotalOrder* has been linked to the *TotalOrder* element in the XML schema for the Internal Purchase Order message.

Once the tracking profile has been created and saved (a good practice is to make the .btt file part of your BizTalk Solution to keep everything together), it can be applied by selecting "Apply Tracking Profile" from the Tools menu. BAM is now ready to be used and will be "filled" automatically when executing the business process by feeding it with purchase orders.

The actual disclosure of the business activity monitoring information in your SharePoint portal will be explained in the next chapters.

SharePoint Configuration

The Microsoft Platform offers some great integration capabilities between components. People tend to forget however that, although we would like to share our data over multiply layers, we still need to take security very serious. We would like the data from our analysis services cubes to be available to our users, but only those slices they are authorized for. And that is where things go bad very often. Administrators have two options basically:

1. The proper way…
   Take all the steps to configure every layer of the system.
2. The lazy / sloppy / bad way…
   Impersonate the Administrator account starting right from the first request. Forcing our way through.

Fortunately SharePoint is secure by default so you do have to make some configurations before you will be able to access your Analysis Services cubes. You still have the options to configure them badly, but that is up to you.
In this walkthrough we are going to show you the proper way for configuring Excel Services on MOSS 2007. In the end we will be querying our SQL Server Analysis Services 2008 cubes with the windows credentials of the logged on user.

Where to store you connection?

There are multiple ways to store and use data connections. Let us have a look at them first to help us understand why we are creating a Data Connection Library anyways.

Embed a data connection inside your document

Embedding data connections inside your document is okay if you are testing your data. Otherwise simply do not use it. It introduces manageability problems right away. If a server name changes for example you will only to find out that it your connection does not work as soon as you open your document and start to update your data. The nice thing though is that it is easy to share the documents and your connection data is embedded. You do have to question though if you would really like to share your connection data with anybody else at all.

Save the connection file on the computer

Saving the connection on the computer is somewhat better already. First of all you can reuse the connection in multiple documents. If something changes, you change that one connection file and all documents will continue to work as expected. Once you start to share your document to other computers you will notice that the data cannot be refreshed. Especially nowadays most people work from more than one computer so this is not a really good option either.

Save the connection file on a network share

Saving the connection on a network share is better again because multiple users who have access rights to the connection can reach and thus use the connection. One of the problems that can arise is that you are not connected to the network at all times. Also consider that it is hard to maintain connection files when people start to copy them all over your network.

Save the connection file to a SharePoint Data Connection library

Saving the connections to a SharePoint Data Connection library has some major advantages over all the above mentioned. It is a centralized place to store and maintain your connection files. Access rights can be easily maintained from within SharePoint. SharePoint can be reached both from inside the network as well as outside the network.
Another advantage of using a SharePoint Data Connection library is that Excel Services can use those connections to update the data on the server. The client does not need Excel at all.

Create a Data Connection library

We now know why, so let us create a Data Connection Library. Open a browser and navigate to your SharePoint portal site. Login if necessary and head for "*Site Actions -> View All Site Content*". Click *"Create"* and choose the *"Data Connection Library*". Finish the succeeding form as needed and we will have our Data Connection Library.

Where to store your Excel Sheets?

Most financial departments have one or more employees that master Excel to an unbelievable degree. Most of the time they store those Excel sheets locally or on a file share and send management a copy through email. But there are good reasons why you would want your Excel sheets stored in a SharePoint Report Library.

Sending them through email creates a whole lot of versions of a file scattered all over everybody's inboxes. People tend to poke around in those sheets a little, perhaps change some functions or add some records to see another outcome and then save their version of the truth to their computer again. Just a month later they cannot remember they have changed some functions here and there or that they have added some records and they will email their sheet to upper management again. Within the corporation you will now have multiple versions of the truth and no way to know which version is the actual one.

Saving your Excel sheets to a document library makes your Excel sheets discoverable, manageable and it ensures that everybody is looking at the same Excel sheet.

Create a Report Library

We now know why, so Let us create a Report Library. Open a browser and navigate to your SharePoint portal site. Login if necessary and head for "*Site Actions -> View All Site Content*". Click *"Create"* and choose the *"Report Library*". Finish the succeeding form as needed and we will have our Report Library.

The next step is a very important step which is, unfortunately, not so well documented. Excel Services can operate with two different access models:

1. **Trusted Subsystem**
   when this access model is used no full user token is handed to Excel Services so it cannot delegate the user token to other machines. Excel Services will retrieve files from SharePoint libraries with the Excel Services account and verifies if the requesting account has the correct permissions.
2. **Delegation**
   when this access model is used Excel Services will get the full user token and uses this token for delegation in all requests. So it will retrieve the SharePoint files with the given user token and make connections with the given user token

Depending on the type of installation an access model is chosen which cannot be changed from within the UI. For single box installations delegation is set, trusted subsystem is chosen for multi box / farm installations.

It is best to use delegation because your connections will be made under the user account as well. This enables you to use windows authentication and - more important – authorization in your data sources.

Be aware though that **delegation between two computers depends on Kerberos**.

Change the Access Model to use Delegation

Type in the following commands at a command prompt replacing “SharedServices1” with the name of you Shared Service provider to change the access model to use delegation:

```
stsadm -o set-ecssecurity -ssp SharedServices1 -accessmodel delegation
stsadm -o execadmsvcjobs
iisreset
```

Excel Services makes the Excel sheets to run on the server and you do not want any malicious code or connections to run on your server. That is why Excel Services is secure by default. You have to explicitly define trusted:

- Data Connections Libraries
- File locations
- Data Providers
- User-defined function assemblies

If your sheet is not in a trusted file location, uses a non trusted data connection, a data connection that uses a non trusted provider or an assembly that is not trusted it will simply not work.

It is a farm administrator’s task to allow trust, so all these settings are configured from the Shared Services Administration pages of the share service that runs Excel Services. All the next steps you need to take start from the landing page of the Shared Services Administration configuration site of the shared services server that runs Excel Services.

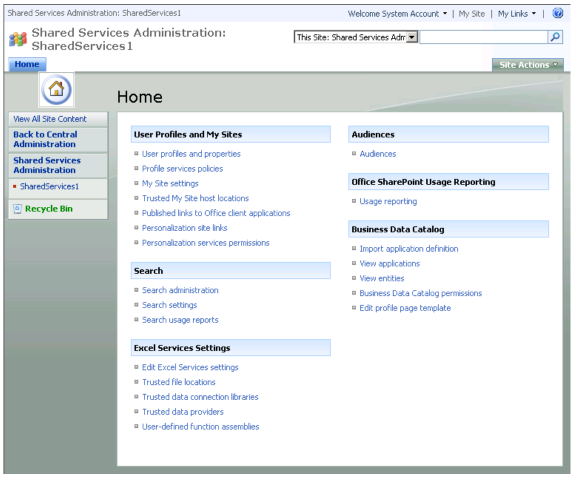

Shared Services Administration: SharedServices1
Welcome System Account ▾ | My Site | My Links ▾ |
Shared Services Administration: SharedServices1
This Site: Shared Services Adr
Home
Site Actions ▾
Home
View All Site Content
Back to Central Administration
Shared Services Administration
SharedServices1
Recycle Bin
User Profiles and My Sites
User profiles and properties
Profile services policies
My Site settings
Trusted My Site host locations
Published links to Office client applications
Personalization site links
Personalization services permissions
Search
Search administration
Search settings
Search usage reports
Excel Services Settings
Edit Excel Services settings
Trusted file locations
Trusted data connection libraries
Trusted data providers
User-defined function assemblies
Audiences
Audiences
Office SharePoint Usage Reporting
Usage reporting
Business Data Catalog
Import application definition
View applications
View entities
Business Data Catalog permissions
Edit profile page template

Figure 41

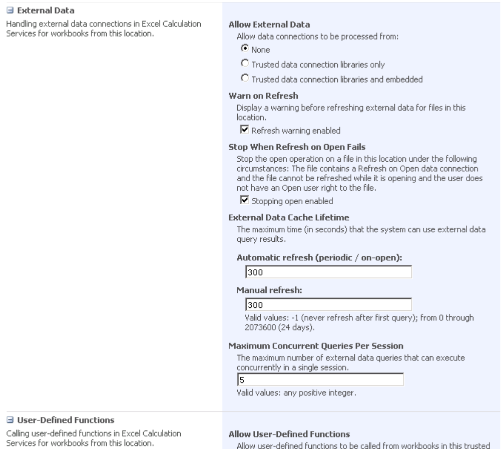
Trusted Locations

You will need to configure your trusted data connection libraries and trusted file locations first. Trusting your data connection libraries holds nothing more that actually fill in the url to the previously created library. There are no additional settings there. Either you trust or do not trust a data connection library.

With trusted file locations there are some options you would definitely like to have a look at. You start by filling in the URL to your previously created library just like you did before while trusting the data connection library.

Secure by default Excel Services does not allow any data connections, which limits the usability of your Excel sheets. If your sheets do not need any external data access that is fine, but for reporting purposes this is hardly ever the case. Embedded libraries on the other hand are difficult to manage and to control. If you want both flexibility and manageability select "Trusted data connection libraries only" in the "External Data" settings group.

In that same settings group you will find the *"Refresh warning enabled"* setting. If checked the end user will get a warning every time the external data connections are refreshed. My experience is that users are annoyed and confused by this kind of messages and select *"continue"* anyways. I prefer to uncheck this checkbox.



⊟ External Data

Handling external data connections in Excel Calculation Services for workbooks from this location.

Allow External Data

Allow data connections to be processed from:

● None

○ Trusted data connection libraries only

○ Trusted data connection libraries and embedded

Warn on Refresh

Display a warning before refreshing external data for files in this location.

☑ Refresh warning enabled

Stop When Refresh on Open Fails

Stop the open operation on a file in this location under the following circumstances: The file contains a Refresh on Open data connection and the file cannot be refreshed while it is opening and the user does not have an Open user right to the file.

☑ Stopping open enabled

External Data Cache Lifetime

The maximum time (in seconds) that the system can use external data query results.

Automatic refresh (periodic / on-open): 300

Manual refresh: 300

Valid values: -1 (never refresh after first query); from 0 through 2073600 (24 days).

Maximum Concurrent Queries Per Session

The maximum number of external data queries that can execute concurrently in a single session.

5

Valid values: any positive integer.

⊟ User-Defined Functions

Calling user-defined functions in Excel Calculation Services for workbooks from this location.

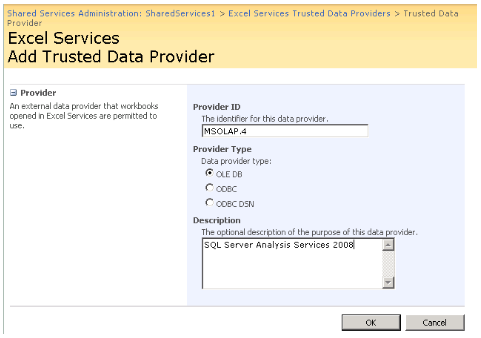
Allow User-Defined Functions

Allow user-defined functions to be called from workbooks in this trusted

Figure 42

Trusted data providers

Excel Services does contain a list of well-known Data Providers which are trusted. Check that the provider you use is on that list. To connect to Microsoft SQL Server Analysis Services 2008 you will need to add the MSOLAP.4 which is not in the list by default.



Shared Services Administration: SharedServices1 > Excel Services Trusted Data Providers > Trusted Data Provider

Excel Services
Add Trusted Data Provider

⊟ Provider
An external data provider that workbooks opened in Excel Services are permitted to use.

Provider ID
The identifier for this data provider.
MSOLAP.4

Provider Type
Data provider type:
● OLE DB
○ ODBC
○ ODBC DSN

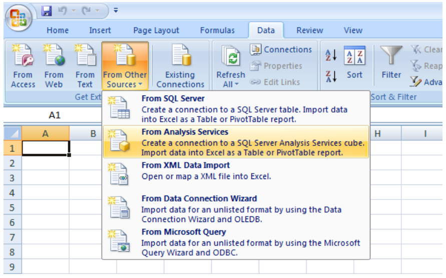
Description
The optional description of the purpose of this data provider.
SQL Server Analysis Services 2008

Figure 43

We have now configured all necessary properties to start using Excel Services. All we need to do now is to create an Office Data Connection file and an Excel sheet using that data connection. We will start by creating an Office Data Connection File.

Create an Office Data Connection File

From the Ribbon in Excel select *"Data > From Other Sources > From Analysis Services"* this will start the Data Connection Wizard for you. Follow this wizard and be sure to check the *"Always attempt to use this file to refresh data"* checkbox.



From SQL Server
Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.

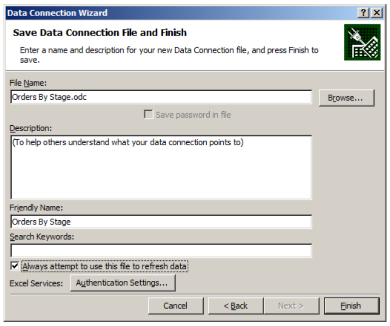
From Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.

From XML Data Import
Open or map a XML file into Excel.

From Data Connection Wizard
Import data for an unlisted format by using the Data Connection Wizard and OLEDB.

From Microsoft Query
Import data for an unlisted format by using the Microsoft Query Wizard and ODBC.

Figure 44



Data Connection Wizard

Save Data Connection File and Finish
Enter a name and description for your new Data Connection file, and press Finish to save.

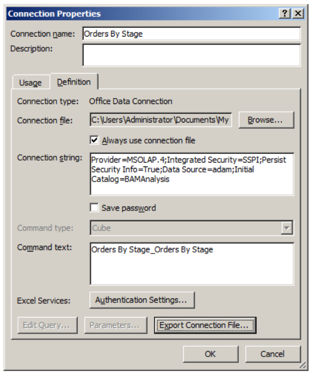
File Name: Orders By Stage.odc
□ Save password in file
Description: (To help others understand what your data connection points to)
Friendly Name: Orders By Stage
Search Keywords:
☑ Always attempt to use this file to refresh data
Excel Services: Authentication Settings...

Figure 45

Once you have clicked *"Finish"* another wizard will start to aid you in creating a pivot table. Do not use this wizard because we only want to create the connection here first. Our Excel sheet does now contain an embedded data connection which is something you would not want to use as explained before. Excel has an option however to export your embedded data connections. Go to *"Data > Connections"* and select your connection.

Then click *"Properties"* and select the *"Definition"* tab. You will find the option to *"Export Connection File…"* right there. Click this button and save your connection directly to your Data Connection Library or to a file share for your administrator to verify, upload and approve.

Do not forget to approve your data connection file once it is uploaded to your Data Connection Library.



Connection Properties

Connection name: Orders By Stage

Description:

Connection type: Office Data Connection

Connection file: C:\Users\Administrator\Documents\My

☑ Always use connection file

Connection string: Provider=MSOLAP.4;Integrated Security=SSPI;Persist Security Info=True;Data Source=adam;Initial Catalog=BAMAnalysis

☐ Save password

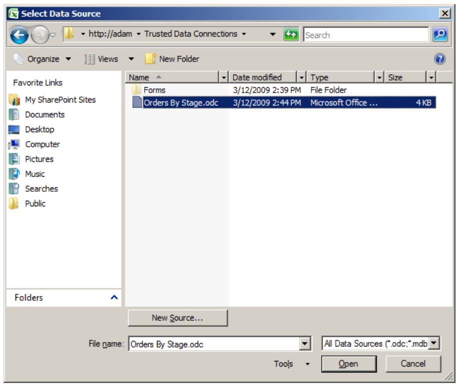
Command type: Cube

Command text: Orders By Stage_Orders By Stage

Figure 46

From that moment on you can use the data connection file in your Excel sheets.

Next, create a new Excel sheet or delete the embedded data connection from the Excel sheet we used for creating the data connection file. Go to *"Data >* ***Existing*** *Connections"* and navigate to your SharePoint Data Connection Library.



Select Data Source

http://adam ▸ Trusted Data Connections

Name	Date modified	Type	Size
Forms	3/12/2009 2:39 PM	File Folder	
Orders By Stage.odc	3/12/2009 2:44 PM	Microsoft Office ...	4 KB

File name: Orders By Stage.odc

All Data Sources (*.odc;*.mdb

Figure 47

Select your data connection file, click *"Open"* and follow the wizard to complete the Excel sheet you would like to publish to SharePoint. Note that you can actually name objects in your Excel sheet. It is good practice to give all of your objects a decent name because they will be used in SharePoint later on. Simply select an object and navigate to *"Layout"* or *"Options"*, depending on the type of object, to change the object's name.

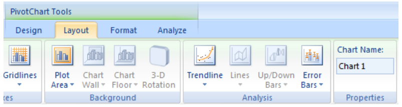


Chart Name: Chart 1

Figure 48

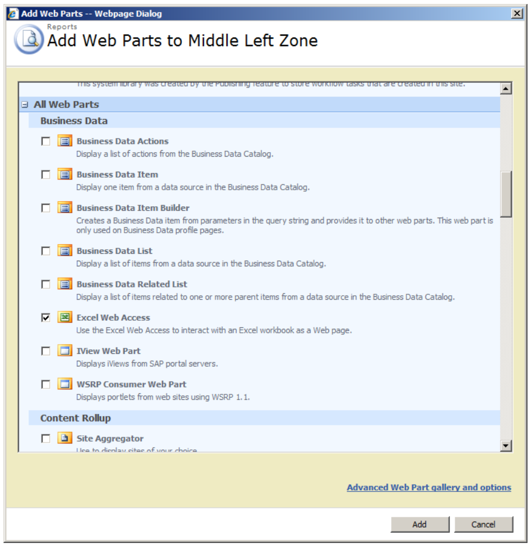
Once you are finished you can publish the sheet to the trusted report library we have created before. You can simply follow the wizard that starts when you select *“Publish > Excel Services”* from the Excel menu. Do pay attention to the “Excel Services Options…” button in the “Save as” dialog. If you click that button you can select which parts of the Excel sheet you would like to publish.

Once you have finished the dialog you can navigate to the SharePoint library and view your Excel sheets directly from there, rendered by Excel Services.

Designing a Dashboard

With the Excel sheet created we are not finished yet. You cannot expect a BizTalk administrator to open multiple Excel sheets and combine the data from those sheets into a clear view. You can solve this by adding a dashboard page to your site. On that dashboard page you can add multiple Excel Web Access web parts.

Edit the page as you would normally do and click *‘Add a Web Part’* in the web part zone you would like to add an object from you excel sheet. From the list of web parts in the *‘Add Web Part – Webpage Dialog’* you select the *‘Excel Web Access’* web part.



Add Web Parts -- Webpage Dialog

Reports
Add Web Parts to Middle Left Zone

This system library was created by the Publishing feature to store workflow tasks that are created in this site.

⊟ All Web Parts

Business Data

☐ Business Data Actions
Display a list of actions from the Business Data Catalog.

☐ Business Data Item
Display one item from a data source in the Business Data Catalog.

☐ Business Data Item Builder
Creates a Business Data item from parameters in the query string and provides it to other web parts. This web part is only used on Business Data profile pages.

☐ Business Data List
Display a list of items from a data source in the Business Data Catalog.

☐ Business Data Related List
Display a list of items related to one or more parent items from a data source in the Business Data Catalog.

☑ Excel Web Access
Use the Excel Web Access to interact with an Excel workbook as a Web page.

☐ IView Web Part
Displays iViews from SAP portal servers.

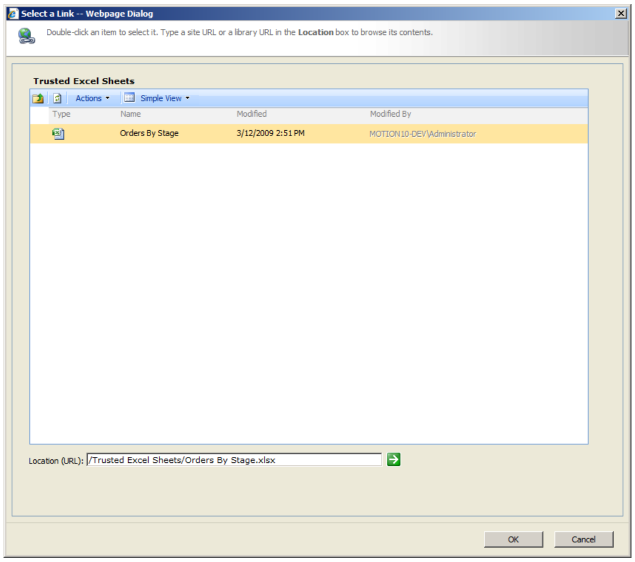
☐ WSRP Consumer Web Part
Displays portlets from web sites using WSRP 1.1.

Content Rollup

☐ Site Aggregator
Use to display sites of your choice.

Figure 49

All you have to do now is edit the properties of the web part. For the *‘Workbook’* property navigate to the published workbook.



Select a Link -- Webpage Dialog

Double-click an item to select it. Type a site URL or a library URL in the **Location** box to browse its contents.

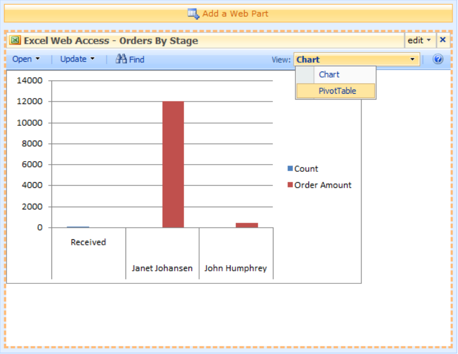
Trusted Excel Sheets

Type	Name	Modified	Modified By
	Orders By Stage	3/12/2009 2:51 PM	MOTION10-DEV\Administrator

Location (URL): /Trusted Excel Sheets/Orders By Stage.xlsx

Figure 50

Once you click apply you'll find your workbook rendered in your browser. Now remember the part where I told you to name your objects decently? You will find that objects in the web part toolbar in the *'View'* dropdown.



Excel Web Access - Orders By Stage

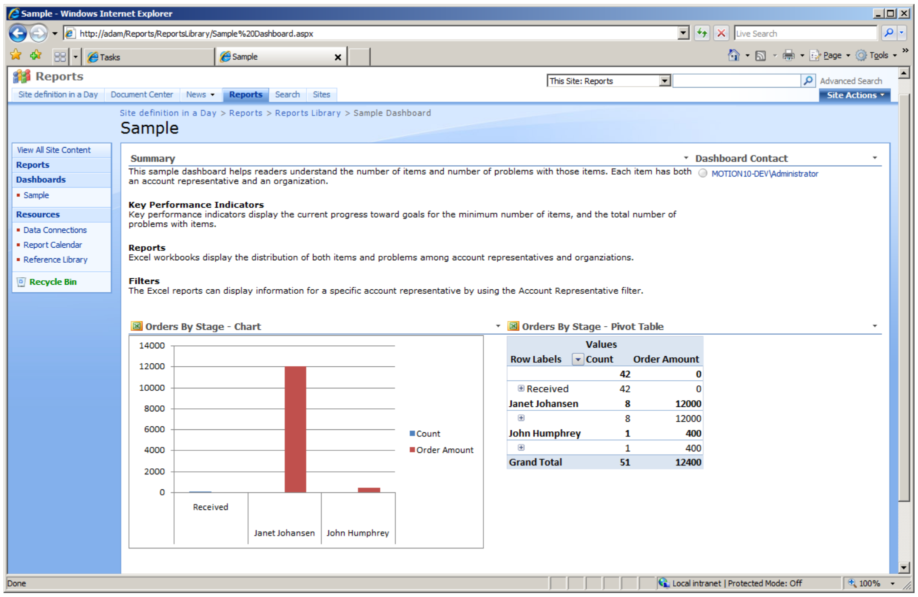
View: Chart

Figure 51

If you do not want the end user to switch between views, you can enter the name of an object into the *'Named Item'* property of your web part and set the *'Type of Toolbar'* property to "None". You can add the same Excel sheet multiple times and only show one of the objects in that sheet to design your dashboard.

Results

The result is a dashboard page were a business user can get an instant overview of how things are going inside the BizTalk server processes from a business perspective. In an updated version of this white paper we will also describe how to integrate the SCOM 2007 R2 reporting to enable a technical view for the BizTalk Operators.



Sample

Summary

This sample dashboard helps readers understand the number of items and number of problems with those items. Each item has both an account representative and an organization.

Key Performance Indicators

Key performance indicators display the current progress toward goals for the minimum number of items, and the total number of problems with items.

Reports

Excel workbooks display the distribution of both items and problems among account representatives and organziations.

Filters

The Excel reports can display information for a specific account representative by using the Account Representative filter.

Orders By Stage - Chart

Orders By Stage - Pivot Table

	Values	
Row Labels	Count	Order Amount
	42	0
⊞ Received	42	0
Janet Johansen	8	12000
⊞	8	12000
John Humphrey	1	400
⊞	1	400
Grand Total	51	12400

Figure 52

Conclusion

With this white paper we have hopefully been successful in conveying our message that the integration of BizTalk Server and SharePoint results in actually more than the sum of these two fine products. The three examples given can be used in practice right away, giving BizTalk Operators and the business people a user-friendly and efficient inter"face" to the rather technically oriented, but great integration platform that BizTalk Server is.

BizTalk + SharePoint: 1+1=3

About motion10

motion10 is a global provider of consultancy services geared to complex integrated environments. Its service offering also includes implementation, support and training. Unlike other systems integrators motion10 focuses exclusively on the Microsoft Application Platform which supports products such as BizTalk Server, SharePoint Server and SQL Server. The company has its own experts who also advise Microsoft in the further development of these products. Headquartered in Rotterdam (the Netherlands), motion10 has offices in Atlanta and Seattle, and an extensive partner network across Europe, North America and Asia. The motion10 consultants have completed a total of more than 500 BizTalk Server implementations. For more information: www.motion10.com.