

# XForms and DB2 pureXML

## End-to-end XML data exchange with XForms and DB2 pureXML for IRS e-File Form 1120

Skill Level: Intermediate

[Keith Wells \(wellsk@us.ibm.com\)](mailto:wellsk@us.ibm.com)

Advisory Software Engineer  
IBM

[Susan Malaika \(malaika@us.ibm.com\)](mailto:malaika@us.ibm.com)

Senior Technical Staff Member  
IBM

[Christian Pichler \(cpichle@us.ibm.com\)](mailto:cpichle@us.ibm.com)

Data Server Solutions (Co-op)  
IBM

08 May 2008

Understand the end-to-end exchange of XML data from an XForms-based browser to an IBM® DB2® database with full XML support. Learn how easy it is to create XForms and have them communicate with a DB2 database, where XML data can be stored, retrieved, or deleted. Learn, also, how to create the XForms that access the DB2 pureXML through Universal Services.

## Introduction

XML is being widely used as the medium to exchange messages between many software systems due to its simplicity, readability, extensibility, and acceptance across the IT industry. XML technology has been proven to be so successful that other industries such as banking, insurance, healthcare, and retail are also developing XML standards to make communication between systems easier, interoperable, and less cumbersome.

The software industry is also improving the standards and tools to build applications to create and move XML messages and documents across multi-disciplinary applications.

The article "[Universal Services for pureXML using Data Web Services](#)" (developerWorks, May 2008) complements the XForms-based approach described here. Check it out.

This article presents you with technologies for quick creation of pureXML databases for XML messages, Universal Web Services to interact with these pureXML databases, and XForms which can be used to interrogate and visualize data from the stored XML messages to a user in a client-based browser.

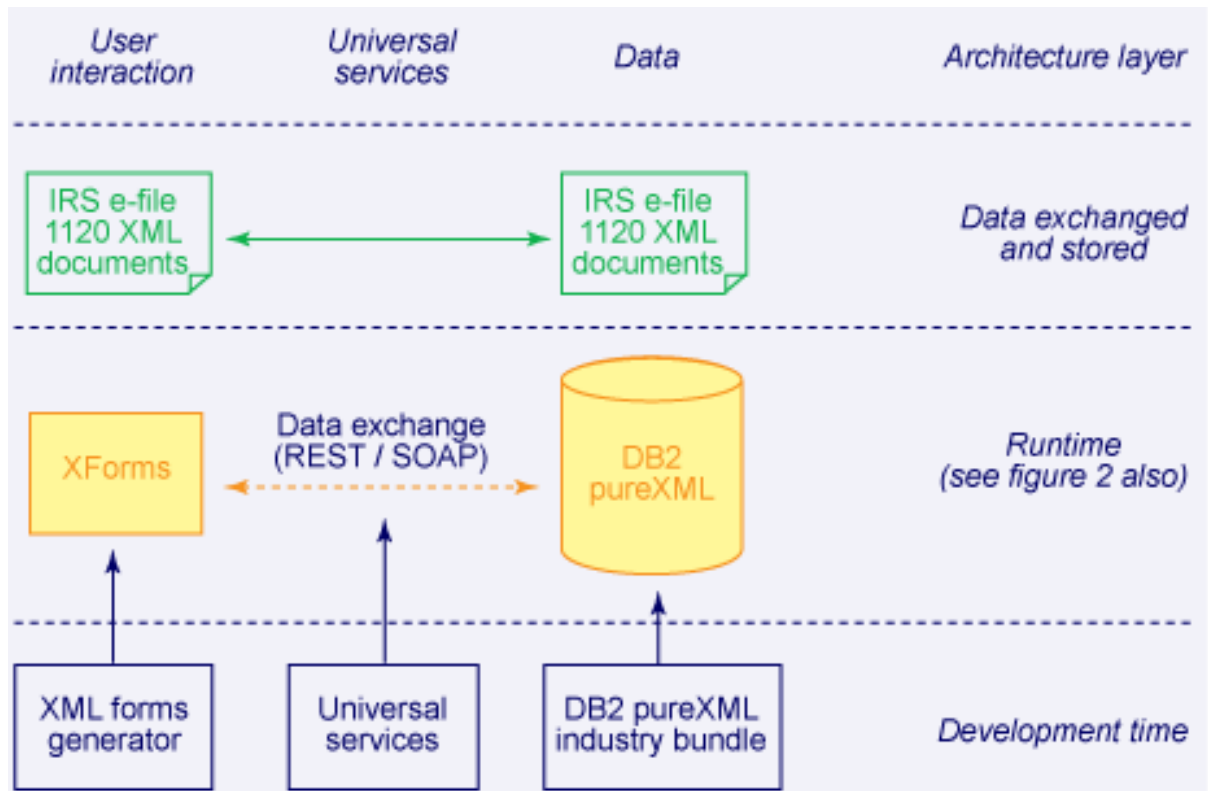
## Goal

The goal of this article is to help you create an XML-based application quickly — making XML standards work for you by exchanging XML documents from an XForms-implementing browser to a DB2 pureXML database that stores XML data natively.

## Setting the scene

This section provides the necessary background you need in order to follow the approach presented throughout this article.

### Figure 1. Setting the scene



### IRS e-File 1120 message

The Internal Revenue Service (IRS) is part of the United States Department of the Treasury that supports individuals and companies to report their income, credits, and other information. Therefore, the IRS defined e-file messages based on the Extensible Markup Language (XML). E-file messages are an electronic alternative to filing paper reports. In particular, the IRS e-File 1120 message is designed for corporations to determine the tax liability for the corporation.

### XForms

XForms is a W3C XML standard for presenting and collecting form-based data. It is tightly integrated with XML with a strong separation between presentation controls and data. See the [Resources](#) section of this article for more developerWorks articles and other resources on XForms.

### XML Forms Generator

**Download XML Forms Generator for FREE!**  
[Download](#) XML Forms Generator (XFG) for free, and use it to create functional, standards-compliant forms.

The XML Forms Generator tool is an IBM alphaWorks package (Eclipse plug-in)

intended to jump-start XForms development. It produces valid and functional forms containing XForms markup embedded within an XHTML document. The input to form generation may be an XML message (optionally) backed by an XML schema or a WSDL document. Response processing templates and combination request/response forms also may be generated from a WSDL document.

Any document the XML Forms Generator produces may serve as a starting point for further form, layout, and styling customizations. An extension point provides opportunities for post-processing during generation of a form.

## DB2 pureXML

### Download DB2 Express-C for FREE!

[Download](#) and use a full-functional relational and XML data server for free.

IBM DB2 supports storing, managing, and querying entire XML documents or document fragments. Application developers can now store XML data directly inside a DB2 database and reap the benefits of transactions, advanced data resiliency, secure access, and the ability to search large amounts of XML data using XQuery.

DB2 pureXML provides simple, efficient access to XML with the same levels of security, integrity, and resiliency taken for granted with relational data.

## DB2 pureXML industry bundles

The industry bundle sample packages and demos contain vertical industry samples of XML documents, links to relevant XML schemas, and sample code to help database administrators, software developers, and architects in vertical industry businesses get started with DB2 9.x pureXML features. If you need more information or are interested in downloading these samples, please refer to the [Resources](#) section. In this article, you will be working with the IRS e-File 1120 pureXML industry bundle.

## Universal Services

The Universal Services enable someone to set up a fixed set of database operations on top of a pureXML database within a short period of time. These database operations are furthermore exposed as Web services operations. The Web services operations exposed allow the user to insert, update, delete, and query XML data stored in a pureXML column of a DB2 database.

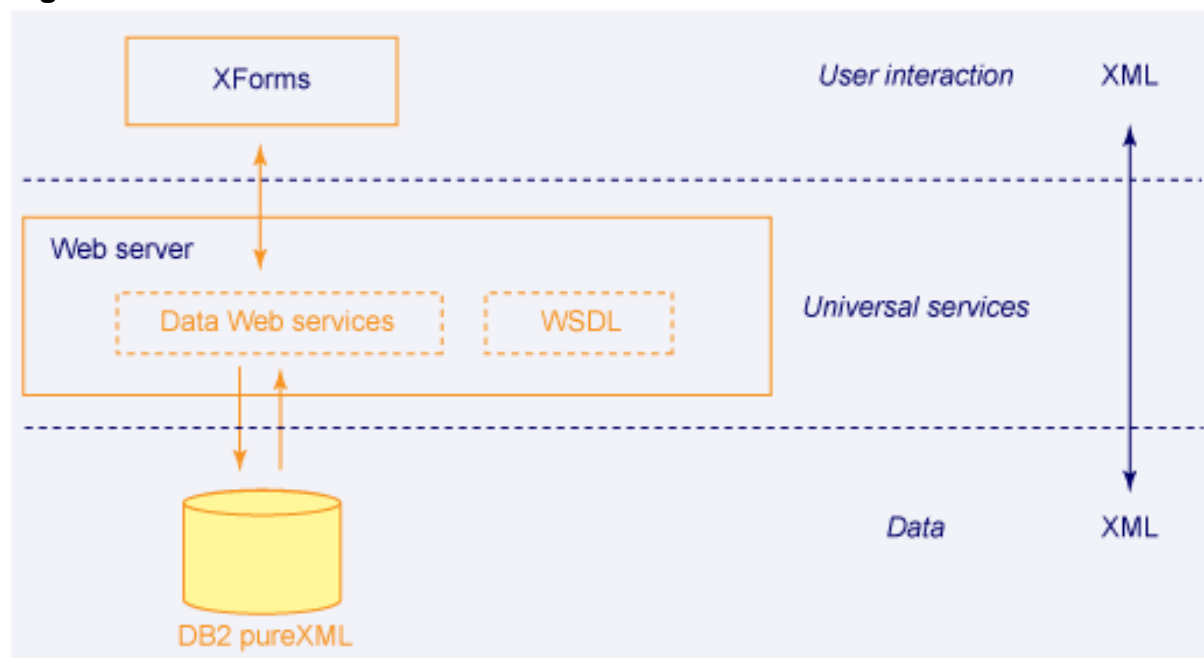
Universal Services are based on the Data Web Services mechanism, which allows you to expose database operations, stored procedures, and user-defined functions as Web services.

## Scenario

This section of the article provides an overview of the scenario setup and describes the parts of the scenario that need to be set up in advance. As shown in [Figure 2](#), the approach followed in this article consists of three layers — data, Universal Services, and user interaction. [Figure 2](#) furthermore illustrates one of the advantages of the approach shown here: **XML data end-to-end**, which means that data is stored as XML, exchanged as XML, and presented to the user as XML.

As illustrated in [Figure 1](#), the same IRS e-File 1120 XML message is stored, exchanged, and used as the basis for display through an XForm.

**Figure 2. Overview architecture**



The purpose of each of these layers is as follows:

- **Data:** The data layer represents a DB2 pureXML database containing a table that has a column of type XML. The pureXML column enables the database to store XML messages in a native format. The data stored in this database is the data you're going to design the XForms for, as part of this article — the IRS 1120 e-File message. A convenient and easy way to set up sample databases for a great variety of different XML-based industry standards is to use the DB2 pureXML Industry Bundles. The industry bundle used in this scenario is the one created for the IRS e-File Form 1120.
- **Universal Services:** The Universal Services layer enables the column of

type XML in the DB2 pureXML database in the Data layer to be accessed and modified through Web service operations.

- **User interaction:** The user interaction layer is the main purpose of this article. It represents user interaction through XForms. These XForms allow the retrieval, modification, and update of the data in the DB2 pureXML database. The XForms access the data in the pureXML database through the Universal Services. The setup of a DB2 pureXML database to store IRS e-File 1120 XML messages and the setup of the Universal Services are described in the article "[Universal Services for pureXML using Data Web Services](#)."

## XForms

This section describes the basic steps to create XForms capable of end-to-end XML exchange with a DB2 pureXML database. These steps are:

- [Step 1: Obtain WSDL file](#)
- [Step 2: Build XForms that interact with Universal Services](#)
- [Step 3: Build XForms that display IRS e-File Form 1120 XML messages](#)
- [Step 4: Merge the XForms \(from Steps 2 and 3\) to create an interactive end-to-end XML application](#)

### Step 1: Obtain WSDL file

The first step in designing the XForm is to generate the WSDL file for the Universal Services by accessing the URL <http://localhost:8080/UniversalServices/wSDL>, as described in the article "[Universal Services for pureXML using Data Web Services](#)." Depending on your local system setup, hostname, port, or both may need to be adapted. Save this file locally, and import it or copy and paste it as a .wsdl file in an Eclipse project. In this article, the WSDL file is saved with the file name `universalservices.wsdl`.

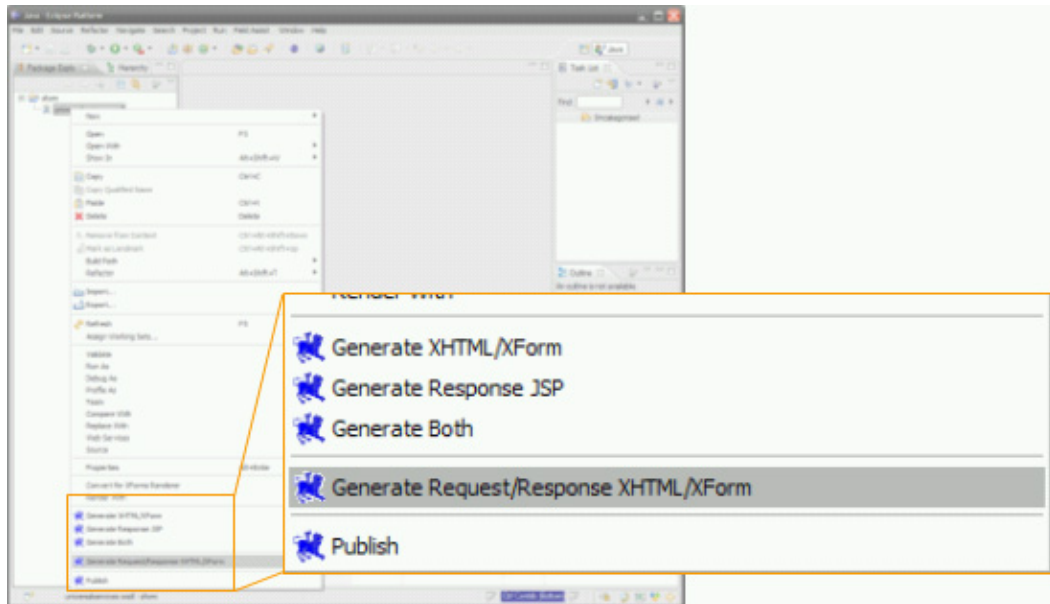
### Step 2: Build XForms that interact with Universal Services

You can generate the interaction (request and response) XForms and the corresponding XML files to communicate with the Universal Web Services for a pureXML solution using the XML Forms Generator (XFG) tool. The XML Forms Generator Eclipse plug-in processes the Universal Services Web service descriptions that were imported into the Eclipse project in the previous step.

1. Select the `universalservices.wsdl` file, right-click on the WSDL file, and choose the **Generate Request/Response XHTML/XForm** wizard from

the XML Forms Generator menu item (see Figure 3 below).

**Figure 3. Generate Request/Response XHTML/XForm from WSDL file**



2. From the wizard, select the Web service operations for the XForms you are producing.

The XML Forms Generator wizard produces two files for each Web service operation selected. One operation is `getXMLDocumentByKey`. The two files produced by the wizard are `getXMLDocumentByKey.xhtml` and `getXMLDocumentByKey.xml`. As you can see in Listing 1, below, `getXMLDocumentByKey.xml` is a SOAP message that will be submitted in your IRS e-File 1120 application:

**Listing 1. Generated SOAP message**

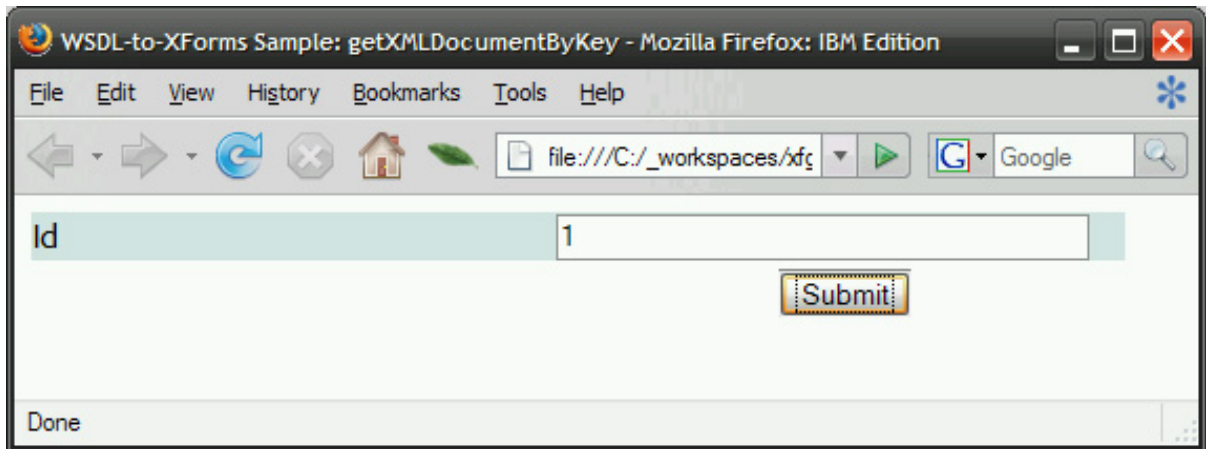
```
<?xml version="1.0" encoding="ASCII"?>
<soapenv:Envelope
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soapenv:Header
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
  <soapenv:Body
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UniversalServices:getXMLDocumentByKey
      xmlns:UniversalServices="UniversalServices">
      <id xsi:type="xsd:string" />
    </UniversalServices:getXMLDocumentByKey>
  </soapenv:Body>
</soapenv:Envelope>
```

The other file generated, namely `getXMLDocumentByKey.xhtml` is the interaction

XForm. If you were to render the generated interaction XForms in a browser, it would look like Figure 4:

**Figure 4. Rendered view of the getXMLDocumentByKey Web service**



You'll use parts of this interaction XForm, as illustrated in Listing 2, later.

**Listing 2. XForms submission for Universal Services**

```
<xforms:submission id="submit_model_Envelope"
  action="http://localhost:8080/UniversalServices/services/UniversalServices"
  method="post"
  mediatype="text/xml"
  replace="instance">
  <xforms:toggle ev:event="xforms-submit-done" case="response"/>
</xforms:submission>
```

### Step 3: Build XForms that display IRS e-File Form 1120 XML messages

Import an IRS 1120 e-File XML message from the industry bundle (as described in the article "[Universal Services for pureXML using Data Web Services](#)") into your Eclipse project. Generate the display XForms with the XML Forms Generator "Generate XHTML/XForms" wizard. The wizard produces an XHTML/XForms file (the display XForms) using the data from the IRS 1120 e-File XML message. Figure 5 shows what the display XForms looks like after it was generated, customized, and rendered in Firefox:

**Figure 5. Rendered view of the IRS e-File 1120 XML sample message**

#### Step 4: Merge the XForms (from Steps 2 and 3) to create an interactive end-to-end XML application

By merging the interaction and display XForms (see [Listing 3](#)), you can put together an application that queries the pureXML DB2 database for IRS e-File 1120 XML messages, selects different messages, and displays the messages using XForms presentation controls. To merge the interaction and display XForms, complete the following steps:

1. Ensure that all namespaces are set properly.
2. Establish the XForms model created during Step 3 to represent the data being returned.
3. Modify the relative XPath references to account for accessing the data from a SOAP envelope.

#### Listing 3. Basic structure of our combined XForms

```
<xforms:switch>
  <xforms:case id="request" selected="true">
    <!-- Start Page for XForms demo -->
    <!-- Create Trigger to get all IRS e-File 1120 messages -->
```

```
</xforms:case>
<xforms:case id="primaryKeys">
  <!-- Select from list of documents -->
  <!-- Copy/Paste from the XForms generated for the
  getXMLDocumentByKey web service -->
</xforms:case>
<xforms:case id="request_xforms">
  <!-- Form and data of an IRS e-File 1120 message -->
  <!-- Copy/Paste from the XForms generated from the
  IRS e-File 1120 XML Document -->
</xforms:case>
</xforms:switch>
```

The [Download](#) section of this article provides a complete example that illustrates how to merge the interaction and display XForms.

## Conclusion

Following the instructions in the article "[Universal Services for pureXML using Data Web Services](#)," you created a database and installed the Web services to interact with the database through the Universal Services. By following the instructions in this article, you built XForms that utilize the Web service interfaces to retrieve, store, and delete XML documents from a DB2 database.

This solution, from XForms to DB2 pureXML, provides end-to-end XML data exchange between XForms-capable browsers and a database (see [Figure 1](#) and [Figure 2](#)). Integrating and submitting XML data from XForms through an SOA interface to a database means there are fewer steps between the user and the database, less server processing, and quicker development time to retrieve and process XML data from a XForms-capable browser.

## Downloads

Description	Name	Size	Download method
XForms example	xforms_example.zip	49KB	<a href="#">HTTP</a>

[Information about download methods](#)

# Resources

## Learn

- ["Universal Services for pureXML using Data Web Services"](#) (developerWorks, May 2008): Get started with configuring, testing, and modifying the Universal Services, a simple but fixed set of database operations that allow the querying and modification of XML data.
- [W3C XForms](#): Find links to the official XForms specification and a variety of XForms rendering options.
- [W3C site](#): Find out more about XHTML, Cascading Style Sheets (CSS), XML, XML Events, XPath, and other related standards.
- [Internal Revenue Service - United States Department of the Treasury](#): Find information on the Internal Revenue Service (IRS), forms to file different reports and electronic versions of these forms, called e-File forms.
- ["IBM Data Studio: Get Started with Data Web Services"](#) (developerWorks, November 2007): Get a very detailed and simple introduction to developing your first Data Web Service.
- ["Data Web Services: Build Web Services the new way to access IBM database servers"](#) (developerWorks, December 2007): Create and customize a Data Web Service. Useful theoretical background on Data Web Services is provided, which includes an architectural overview on Data Web Services. The article addresses different aspects of Data Web Services, such as security.
- ["Expose DB2 9 pureXML using WebSphere Integration Developer"](#) (developerWorks, September 2007): Build a client to access the Universal Services. Follow step-by-step instructions on how to build, test, and deploy a mediation module that can store well-formed XML documents in a DB2 XML column using a WebSphere Integration Developer module.
- [developerWorks Information Management zone](#): Learn more about Information Management. Find technical documentation, how-to articles, education, downloads, product information, and more.
- [developerWorks XForms space](#): Find information on XForms, and share you experiences with other XForms users.
- Stay current with [developerWorks technical events and webcasts](#).
- [Technology bookstore](#): Browse for books on these and other technical topics.

## Get products and technologies

- [Industry Formats and Services with pureXML](#): Download a great variety of examples, for free! Each example illustrates how to work with XML-based

Industry Formats and pureXML. The examples show how to register an XML Schema, how to perform validation of XML instance documents, how to query XML data using XQuery or SQL/XML and much more.

- [DB2 Express-C](#): Download the free version of DB2, which includes the core functionality as the other Data Servers, such as the pureXML technology. DB2 Express-C is free to develop, deploy, and distribute.
- [Mozilla XForms](#): Render your standards-compliant forms in Mozilla Firefox using this plug-in.
- [FormsPlayer](#): Render XForms on Internet Explorer using this plug-in.
- [XML Forms Generator](#): Create functional, standards-compliant forms with a click of the mouse using this Eclipse-based tool from alphaWorks.
- [Visual XForms Designer](#): Check out links to installation instructions, prerequisites, and the forum for this standards-based, easy-to-use Eclipse plug-in enabling the rapid development of documents with XForms mark-up using a visual user interface.
- [Compound XML Document Toolkit](#): Explore other open-standard XML markups, including Scalable Vector Graphics (SVG), MathML, VoiceXML, and Synchronized Multimedia Integration Language (SMIL).
- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.

## Discuss

- [Participate in the discussion forum for this content.](#)
- [XForms technology forum](#): Post or inquire on XForms tips, gotchas, or technical help. Help bring XForm novices and veterans together to advance XForms education and development.
- Participate in [developerWorks blogs](#) and get involved in the developerWorks community.

## About the authors

### Keith Wells

Keith Wells is the technical and team lead for XForms development in IBM Emerging Standards in RTP, N.C. Currently, Mr. Wells is a member of the W3C Forms Working Group and is exploring opportunities with XForms, emerging software standards, industry standards and other XML-based technologies.

---

### Susan Malaika

Susan Malaika is a senior technical staff member in IBM's Information Management Group (part of IBM Software Group). Her specialties include XML, the Web, and databases. She has developed standards that support data for grid environments at the Global Grid Forum. She has also co-authored a book on the Web and published articles on transaction processing and XML. She is a member of the IBM Academy of Technology.

---

### Christian Pichler

Christian Pichler is a co-op from the Technical University of Vienna in Austria, where he is working on his thesis for a double Master's degree in Computer Engineering and Computer Science with a focus on healthcare. For IBM, Christian is working on technologies for storing XML in DB2, and accessing it through Web services, feeds, and XForms. He is specializing in XML standards for healthcare.