

# Generation PB

PowerBuilder Newsletter

## Herzlich willkommen zur dritten Ausgabe von Generation PB! Sie ist vollgepackt mit Nachrichten und Berichten vom Sybase-PowerBuilder-Team.

Was hat sich seit der letzten Ausgabe ereignet? Ein langer, heißer Sommer für die meisten von uns; im August fand die alljährliche Sybase User Conference/Techwave in Orlando statt. Anlässlich der Techwave wurde die lange erwartete Version Pocket PowerBuilder 1.0 freigegeben, und die neueste PB-Roadmap der zukünftigen Ausstattungsmerkmale und Funktionen von PowerBuilder wurde vorgestellt.

Da PB 9.0 das Fundament der UDE (Unified Development Environment) bildet, werden zukünftige Versionen die Strategie von Sybase fortschreiben, neue Technologien wie Web- und komponentenbasierte Entwicklung, Modellierung und Lifecycle-Management (mit PowerDesigner), Web Services, Wi-Fi und

Anwendungsintegration in das PowerBuilder-Instrumentarium einzubeziehen.

In dieser Ausgabe können Sie nachlesen, wie Sie auf verschiedene Komponenten aus Ihrer Pocket PowerBuilder- (PPB)-Anwendung zugreifen können; außerdem stellen wir ein weiteres neues PowerBuilder-Produkt vor: DataWindow .NET, dessen Betazyklus kurz bevorsteht. Darüber hinaus enthält Ausgabe 3 neueste Produktnachrichten und Hinweise auf Veranstaltungen. Achten Sie auch auf Pressemitteilungen und -artikel zu PowerBuilder. Ich werde in der nächsten Zeit in ganz Europa Interviews für die IT-Fachpresse geben. Die Aussichten für PowerBuilder werden immer besser!

**Steven Dunn** - PowerBuilder Sales Manager EMEA

## Ausverkauft! PB9-Launch-Tour durch Europa



Cordiali saluti da Milano - PB9-Launch-Seminar von Sybase Italien



Saludos de Barcelona - PB9-Launch-Seminar von AST



Grüße aus Hamburg - PBUG-Seminar von PowerPeople

Es war eine aufregende Reise: In zwei Monaten 12 Veranstaltungen mit insgesamt über 900 Teilnehmern! Wir möchten an dieser Stelle den Organisatoren, den Kursleitern und natürlich Ihnen, den PowerBuilder-Benutzern, für Ihren Beitrag zum Erfolg dieser Tour danken.

Die große Frage lautet jetzt: Praktizieren Sie auch, was unsere ‚technischen Missionare‘ gepredigt haben? Wir hoffen, die Produktvorführungen (Web-Services, XML DataWindow, PBNI, Pocket PowerBuilder) sind für Sie und Ihre Projekte von Nutzen.

Wir freuen uns auf ein Wiedersehen demnächst! Die Termine der nächsten Veranstaltungen finden Sie auf Seite 6.

### *In der nächsten Ausgabe:*

Weitere Artikel zu Funktionen von PowerBuilder, Pocket PowerBuilder und DataWindow .Net. Um Generation PB regelmäßig zu beziehen, schicken Sie uns bitte eine E-Mail an: [softwarecentre@sybase.com](mailto:softwarecentre@sybase.com)

# Who's who? Cao Lin

## Heute möchten wir Ihnen unseren Senior Engineering Manager Cao Lin vorstellen.

Dr. Cao Lin ist promovierter Diplomingenieur (Master of Engineering und Ph. D.). Als Softwareingenieur hat er ein Datenbanksystem für Finanzinstitute, ein Decision-Support-System, einen C++-Code-Generator, ein Informationsabrufsystem und viele andere Applikationen entwickelt. Er ist in Singapur tätig und arbeitet seit über 5 Jahren in leitender Position an der Entwicklung von PowerBuilder. Sein erstes Projekt bei Sybase war die Entwicklung chinesischer und koreanischer Versionen von PowerBuilder 6.5. Im Jahr 2000 baute er das ausgezeichnete PowerBuilder-Asia-Entwicklungsteam sowie QA-Teams auf. Gemeinsam mit dem PowerBuilder-Team Nordamerika trug er maßgeblich zum Erfolg der Version PowerBuilder 8.0 bei. Als leitender Ingenieur ist er heute für die Entwicklung von PowerBuilder weltweit verantwortlich. Dank Dr. Cao Lin und seinem Team ist PowerBuilder 9.0 die bahnbrechendste Version seit PB 5.0. Er freut sich über das wiederhergestellte Vertrauen der Kunden in PowerBuilder 9.0.



**Cao Lin**  
Senior Engineering Manager

Für ihn ist PowerBuilder das Anwendungsentwicklungstool der nächsten Generation, denn es abstrahiert und vereinfacht die komplexen Technologien und hilft jungen wie erfahrenen Entwicklern, produktiver und mit größerer Wettbewerbseffizienz Anwendungen auf offenen Plattformen zu entwickeln. Insbesondere sind für PowerBuilder 10 und die Folgeversionen folgende Kernfunktionen geplant: Das iterative Designentwicklungsverfahren wird es Entwicklern ermöglichen, durch Modellieren und Reverse-Engineering ein komplettes Bild des Klassendiagramms zu erzeugen. Das DataWindow.NET wird die Kombination von DataWindow mit anderen .NET-Sprachen erlauben. Der .NET-Compiler wird PB-Entwicklern die Benutzung anderer .NET-Sprachen in der PowerBuilder-IDE gestatten. Mit dem DataWindow für Java werden Java-Entwickler DataWindow auch auf der Java-Plattform benutzen können. Künftig werden PB-Entwicklern Optionen zur Generierung von PB-, .NET- oder Java-Anwendungen in der PowerBuilder-IDE zur Verfügung stehen. Außerdem werden sie wählen können, ob sie ihre Anwendungen an Desktop- oder kabellose Geräte bereitstellen wollen.

## Tipps und Tricks

### 10 Tipps für die Migration von Anwendungen zu PowerBuilder 9.0 Lee Cheetham, Sybase Professional Services Consultant

**1) Mit dem Migration Assistant** können Sie obsoletere Funktionen und Events identifizieren. Der Migration Assistant durchsucht PowerBuilder-Bibliotheken nach potenziellen Syntaxproblemen, bevor die Anwendungsmigration durchgeführt wird. Die Funktion ‚Obsolete Syntax‘ existiert seit Version 6. Den Migration Assistant finden Sie auf der Registerkarte "Tool" des Dialogfelds "New" (Object).

**2) PSR-Dateien:** Versucht eine Anwendung, eine mit einer früheren Version von PowerBuilder (vor 8.0 Build 7063 oder vor 7.0 Build 10102) erstellte PSR-Datei zu öffnen, schlägt dies fehl. Der Grund ist eine Veränderung der Funktion SaveAsASCII. Zur Behebung des Problems müssen Sie die PSR-Datei zunächst in einer neueren Build-Version von PowerBuilder aktualisieren.

**3) Neue reservierte Wörter** (Try, Catch, Finally, Throw, Throws) wurden in PowerBuilder 8.0 für das Exception-Handling eingeführt. Der Migration Assistant identifiziert jedes Vorkommen dieser reservierten Wörter. Um das Problem zu beheben, ändern Sie in Ihrer Anwendung die Wörter Try, Catch, Finally, Throw, Throws überall, wo sie vorkommen, mit einer früheren Version von PowerBuilder in andere geeignete Wörter um.

**4) Verteiltes Programmieren:** wird seit PowerBuilder 8.0 nicht mehr unterstützt.

Das Objekt "Transport" ist überflüssig geworden. Die Anwendungsereignisse "connectionbegin" und "connectionend" werden nicht mehr ausgeführt. Die Lösung: Stellen Sie die serverseitigen NVOs in EA Server bereit.

**5) Die Verarbeitung von SystemError-Events** wurde zwischen PowerBuilder 7 und neueren Versionen von PowerBuilder verändert. In PowerBuilder 7 wird, wenn beim Auftreten eines Fehlers der Event "SystemError" ausgelöst wurde, nach dessen Abarbeitung die Steuerung wieder an das Skript zurückgegeben, in dem der Fehler aufgetreten war. In PowerBuilder 8 und 9 dagegen wird beim Auftreten eines Fehlers der Aufruf-Stack aufgelöst; anschließend wird der Event "SystemError" ausgelöst. Nach dessen Abarbeitung wird das Programm NICHT an der Stelle wiederaufgenommen, an der der Fehler aufgetreten war. Erfassen und behandeln Sie Fehler künftig lokal mit Hilfe der neuen Exception-Handling-Funktionalität von PowerBuilder 8 und 9.

**6) Die Funktion IsValid** wurde in PowerBuilder 8 geändert. Sie gibt den Wert "FALSE" zurück, wenn von dem Objekt keine Instanz gebildet oder ein ungültiges Objekt übergeben wurde. Vor Version 8 wurde bei Übergabe eines ungültigen Objekts der Event "SystemError" ausgelöst.

**7) Fenster** erben nicht mehr das Symbol "Application"; stattdessen müssen Sie jetzt

den Programmcode entsprechend schreiben oder die Eigenschaft (Property) auf "AppIcon!" einstellen.

**8) TreeView-Event:** Der Event "rbuttonup!" wird nicht mehr ausgeführt; dies gilt seit PowerBuilder 7.

**9) ListView:** Der Event "pbm\_rbuttonup!" wird bei Anklicken eines ListView-Eintrags mit der rechten Maustaste nicht mehr ausgeführt, sondern nur noch dann, wenn man auf dem weißen Bereich der ListView-Ansicht klickt. Anstelle dieses Events sollten Sie künftig den neuen Event "pbm\_contextmenu!" benutzen, der immer ausgeführt wird, wenn die rechte Maustaste losgelassen wird.

**10) Web Targets,** die von dem Steuerelement WebDataWindow Gebrauch machen, müssen geändert werden und die neue Komponente HTMLGenerator90 benutzen. Klicken Sie mit der rechten Maustaste auf das Element WebDataWindow Design Time Control auf Ihrer Webseite und wählen Sie "Properties". Ändern Sie auf der Registerkarte "HTML Generator" die Angabe von "HTMLGenerator80" in "HTMLGenerator90" um. Nach dieser Änderung gehen manchmal die Datenbank-Verbindungseigenschaften verloren; deshalb sollten Sie die Registerkarte "Connection" dahingehend überprüfen, ob die Datenbank-Verbindungseinstellung nach der Änderung noch stimmt.

# DataWindow .NET – So bringen Sie die Leistungsfähigkeit des DataWindows in Visual Studio .NET ein

David Avera, Staff Engineer, Sybase, Inc

Erfahrene PowerBuilder-Programmierer sind mit der Leistungsfähigkeit des DataWindow und des DataStore wohl vertraut. Diese patentgeschützte Software (US-Patente 5566330, 5752018, 5832481, 5937415) wird in der Patentschrift beschrieben als "Erfindung, die ein System zur Definition einer Datenbankschnittstelle vorsieht, mit welcher Anwendungsprogrammierer das Datenfenster grafisch definieren, anzeigen und [als sichtbares Objekt] zur indirekten Handhabung von Daten in einer Anwendungsdatenbank verwenden können". Außerdem heißt es: "Die Erfindung sieht ein Programmobjekt [ein DataWindow-Objekt] vor, das eine Schnittstelle zwischen einem Computerdatenbank-Manager, der eine Datenbanktabelle verwaltet, und einem Client-Anwendungsprogramm außerhalb des Schnittstellenobjekts bereitstellt."

Sie können die Patentanmeldungen im Internet nachlesen – die Wirkung bei Schlaflosigkeit ist überwältigend! Kurz gefasst, das DataWindow ist der Kronjuwel von PowerBuilder und eins der herausragenden Merkmale, in denen PowerBuilder alle Konkurrenzprodukten überflügelt. Bis heute hat kein anderer Anbieter ein Produkt vorgelegt, das mit der Leistungsfähigkeit und Benutzerfreundlichkeit des DataWindow mithalten könnte.

## Warum DataWindow .NET?

Seit den Anfängen in den frühen 90er Jahren war das DataWindow ein exklusives Merkmal der PowerBuilder-Entwicklung, wenn man von DataWindow ActiveX und DataWindow for Java™ absieht. Mit der .NET-Technologie von Microsoft bot sich Sybase eine Chance, das DataWindow auch an Entwickler zu lizenzieren, die Microsoft-Sprachen und -Tools verwenden. Hier eröffnete sich ein sehr großer Markt, der primär Visual-Basic-Entwickler, aber auch in zunehmendem Maße C#-Entwickler umfasst.

Darüber hinaus gibt es immer mehr PowerBuilder-Schmieden, die einen Teil ihrer Anwendungen auftragsgemäß mit .NET-Tools anfertigen. Mit DataWindow .NET können diese Entwickler ihre Investitionen in DataWindow-Technologie weiternutzen; all das, ohne sich in der Arbeit mit den Microsoft-Produkten DataGrid, DataView, DataSet und den diversen Adapter- und Konnektortypen schulen bzw. weiterbilden zu müssen. Bislang waren die Benutzer von Microsoft-Entwicklungstools auf ein buntes Gemisch von Komponenten wie DataGrid, DataView, DataAdapters für jede unterstützte Datenbank, DataSets, Sql Connection und Sql Command Objects angewiesen. Es gibt einen lebhaften Markt für Produkte von Drittanbietern zur Erweiterung dieser Komponenten und zur Erleichterung ihrer Benutzung (wobei sie an das DataWindow angeglichen werden).

Sybase wird eine bessere Lösung für Datenzugriff, Datenberichterstattung und Datenhandhabung anbieten: DataWindow .NET.

Das Produkt DataWindow .NET wird sämtliche Dienste der vertrauten PowerBuilder-Komponenten DataWindow und DataStore in der Visual-Studio-IDE bereitstellen. Die Benutzer werden in der Lage sein, DataWindow .NET mit jeder beliebigen .NET-Sprache zu programmieren.

## Woraus besteht DataWindow .NET?

DataWindow .NET wird als Satz von DLLs ausgeliefert werden, die ein .NET-Frontend sowie einen DataWindow-Server als Backend bilden.

Für die Erstellung und Wartung von DataWindow-Definitionen im Rahmen des Designs wird das Gesamtpaket auch den DataWindow Builder einbeziehen.

Das DataWindow .NET-Frontend bildet die Schnittstelle zwischen dem Visual-Basic- bzw. C#-Anwendungsprogramm und dem DataWindow-Server. Über das Frontend werden die vertrauten Methoden, Events und Eigenschaften (Properties) des DataWindow auf die Anwendung abgebildet. Darüber hinaus sorgt das Frontend dafür, dass .NET-Anwendungen nur mit reinem .NET-Code direkt interagieren. Der DataWindow-Backend-Server ist im Prinzip für die Anwendung unsichtbar; er hat lediglich dafür zu sorgen, dass sich die DLLs auf dem Ladepfad befinden.

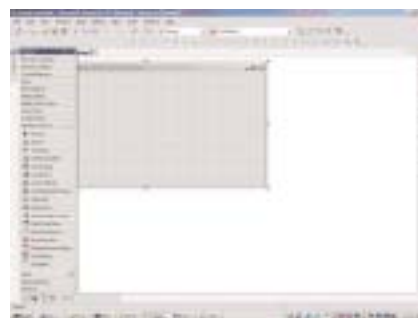
Der DataWindow-Server verwaltet das Laden und die Präsentation von DataWindow-Objekten und die Kommunikation mit Datenbanken. Alle traditionellen Funktionalitäten des DataWindow, z. B. die Wartung der Datenpuffer, die Sortierung, Filterung und der Export von Daten in diverse Dateiformate und DataWindow-Ausdrücke, werden hier abgewickelt.

## Wie arbeitet man mit DataWindow .NET?

Und jetzt fängt die Sache an, richtig gut zu werden!

Angenommen, Sie haben bestimmte DataWindow-Definitionen in einer PBL- oder PBD-Datei gespeichert. Erstellen wir als Beispiel eine einfache C#-Anwendung mit dem Visual-Studio-Assistenten für neue Projekte. Wählen Sie "Visual C# Projects" und "Windows Application". In der Anwendung werden wir ein DataWindow verwenden, mit der Datenbank Verbindung aufnehmen und die Daten abrufen und anzeigen. Außerdem werden wir einen Event-Handler für den DataWindow-Event RetrieveEnd einbauen.

*Hinweis des Autors: Die nachstehenden Abbildungen und Programmbeispiele entstammen der Vor-Alpha-Version und stimmen nicht unbedingt exakt mit dem fertigen Produkt überein.*



**Abbildung 1: Vorlage mit Toolbox**

In der Toolbox sehen Sie die Komponente 'DataWindowControl'. Klicken Sie darauf, und legen Sie die Komponente auf der leeren Vorlage (Form) ab. Sie sehen nun ein leeres Rechteck, das die zukünftige Position der Komponente DataWindowControl wiedergibt.



**Abbildung 2: Vorlage mit leerem Element DataWindowControl**



### Eingabe von Eigenschaften für die DataWindowControl

**Abbildung 3.** Die Eigenschaften-Registerkarte für DataWindowControl

Machen Sie das DataWindow durch Anklicken zum aktuellen Steuerelement, und rufen Sie die Registerkarte "Properties" auf.

Hier sehen Sie nun einige vertraute Eigenschaften des DataWindow-Objekts. Wählen Sie "LibraryList", um festzulegen, woher die DataWindow-Definition geladen werden soll. Wählen Sie dann die Eigenschaft 'DataObject', und wählen Sie ein spezifisches DataWindow-Objekt aus.

### Auswählen einer PBL (oder PBD) mit DataWindow-Definitionen

**Abbildung 4.** Dialogfeld "LibraryList"



### Auswählen eines DataWindow aus der PBL bzw. PBD

**Abbildung 5.** Auswahl-Dialogfeld "DataWindowObject"

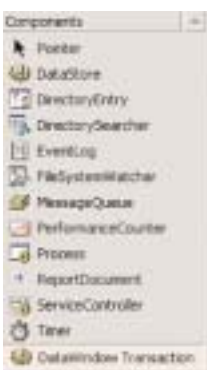


### Auf der Vorlage sieht das Steuerelement DataWindowControl jetzt so aus:

**Abbildung 6.** DataWindowControl mit Titeln

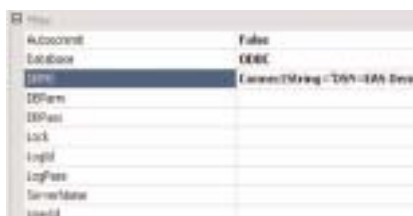


### Hinzufügen einer DataWindow-Transaktion zu der Anwendung



**Abbildung 7.** Toolbox von Visual Studio

Wählen Sie auf der Registerkarte "Components" aus der Toolbox die Komponente "DataWindow Transaction" aus, und legen Sie diese auf der Vorlage ab. Füllen Sie die Eigenschaften aus.



**Abbildung 8.** Transaktions-eigenschaften

(Weiter zu Abb. 7 und 8) Durch das Ziehen und Ablegen der Transaktion und das Festlegen der Eigenschaften wird in der Anwendung automatisch eine Transaktionsobjekt-Instanz gebildet (wir nennen sie SQLCA). Dies kann natürlich auch im C#-Code erfolgen. Sehen wir uns nun den von der IDE generierten Code an:

```
this.DWC1 = new
Sybase.DataWindow.DataWindowControl();
this.SQLCA = new
Sybase.DataWindow.Transaction(this.components);
```

und

```
this.DWC1.DataWindowObject = "d_dept";
this.DWC1.LibraryList = "E:\PB Diamond
Workspaces\externa.pbl";
```

### Jetzt etwas ,richtige' Programmierarbeit

Im 'Form Constructor' können Sie folgende Verbindung zur Datenbank herstellen:

```
public Form1()
{
    //
    // Required for Windows Form Designer
    support
    //
    InitializeComponent();
    //
    // TODO: Add any constructor code after
    InitializeComponent call
    //
    SQLCA.Connect( );
}
```

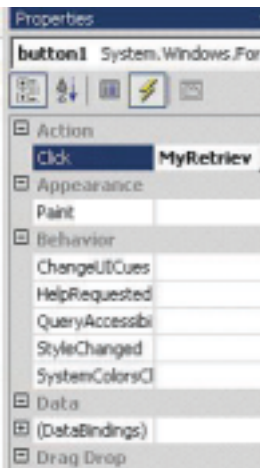
Jetzt folgt der Code für den DataWindow-Abruf. Wählen Sie über die Toolbox auf der Registerkarte "Windows Form" ein Button-Steuerelement, und legen Sie es auf der Vorlage ab. Geben Sie für die Button-Beschriftung Eigenschaften ein, etwa "Retrieve DataWindow"; wählen Sie auf der Registerkarte "Properties" Events aus, und geben Sie den Namen einer Methode ein, die Sie anwenden möchten.

**Abbildung 9.** Die Vorlage mit Button



**Abbildung 10.** Registerkarte ,Properties' für den Button mit dem markierten Event

Auf der Code-Ansicht der IDE können Sie den folgenden eingefügten Standardcode sehen:



(weiter zu Ab. 10)

```
private void
MyRetrieve(object sender,
System.EventArgs e)
{
}

```

Fügen Sie folgenden Code in die MyRetrieve-Methode ein:

```
DWC1. SetTransObject
(SQLCA);
DWC1. Retrieve ( );

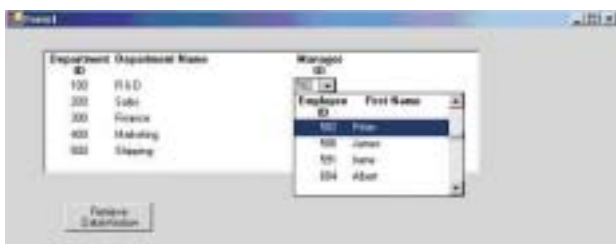
```

```
int DataWindowControl.Retrieve (params
object[] Argstlist)
Retrieves rows from the database for the
DataWindow control. If arguments are included,
the argument values are used for the retrieval
arguments in the SQL SELECT statement for the
DataWindow object.
```

Wenn Sie die Maus über das Verb bringen, zeigt die ‚Intellisense‘-Funktion von Visual Studio eine Erklärung zu der Methode an.

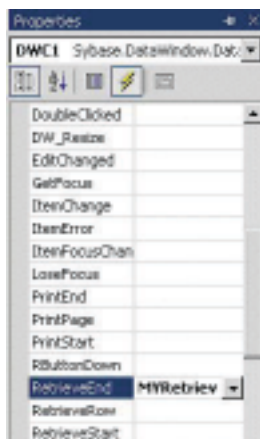
### Nun führen Sie die Anwendung aus und betätigen den Button "Retrieve DataWindow":

**Abbildung 11.** Anzeige der Vorlage bei der Ausführung Die bekannte ‚Department‘-Tabelle wurde abgerufen und wird angezeigt. Beachten Sie, dass Dropdown-DataWindows funktionieren.



### Angenommen, wir wollen wissen, wann der Abruf endet, und wie viele Zeilen abgerufen wurden.

Kehren Sie zur IDE zurück, machen Sie DataWindowControl zum aktuellen Steuerelement, und rufen Sie die Registerkarte "Properties" auf.



**Abbildung 12.** Registerkarte ‚Properties‘ der DataWindowControl mit Events

Auf der Code-Ansicht erscheint dieser Standardcode:

```
private int MYRetrieveEnd(object sender,
Sybase.DataWindow.DataWindowEvents.RetrieveEndArg
s args)
{
return 0;
}

```

Fügen Sie folgenden Code in die Methode ein:

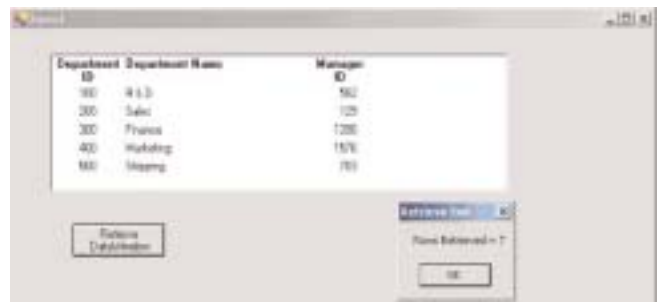
```
String Rows = args.RowsRetrieved.ToString( );
MessageBox.Show ("Rows Retrieved = "+ Rows,
"Retrieve End");

```

### Abruf mit ‚RetrieveEnd‘-Eventhandler-Methode

**Abbildung 13.** Ausführung der Vorlage mit Meldungsbbox "Retrieve End"

Dieses Beispiel zeigt, wie einfach es ist, in einer .NET-C#-Anwendung ein vorhandenes DataWindow zu verwenden. In Visual Basic .NET ist es nicht schwerer.



Über das am obigen Beispiel Gezeigte hinaus werden wir die ganze Leistungsfähigkeit des DataWindows für .NET-Programmierer erschließen. Einer unserer internen PowerBuilder-Experten erzielt mit der Portierung der PowerBuilder-DataWindow-Beispiele auf Visual Basic bereits beachtliche Erfolge für ein Produkt in der Vor-Alpha-Phase. Diese Beispiele werden beweisen, dass komplexere Verwendungsweisen des DataWindow genau so praktikabel sind wie unser einfaches Beispiel

### Was wird vom Endprodukt unterstützt werden?

Fast alle vertrauten DataWindow-Methoden, Eigenschaften und Events werden aussehen wie in PowerBuilder, mit wenigen Ausnahmen:

1. DataWindow-Events werden anhand des .NET-Eventmodells programmiert; die gleichen Event-Argumente werden zur Verfügung stehen, jedoch in Form von Event-spezifischen Argumentobjekten.
2. Die Punkt- (Dot-) Notation wird in DataWindow .NET nicht unterstützt. SetProperty- und GetProperty-Methoden werden verfügbar sein.
3. Das Richtext-DataWindow-Format wird nicht unterstützt.

## Fazit

In diesem Artikel habe ich versucht darzulegen, wie einfach es sein wird, mit DataWindow .NET an einer Visual-Studio-Anwendung in C# zu arbeiten. Wenn Sie mit Visual Basic vertraut sind, ist das Verfahren ebenso einfach. Anhand unseres Beispiels haben wir gesehen, wie man eine Anwendung erstellen kann, die Datenbankdaten abrufen und anzeigt, indem man ein paar Elemente aus der Toolbox anklickt und ablegt, mit Eigenschaften versieht und mit etwas zusätzlichem Code ergänzt. Was sich hinter diesem simplen Beispiel verbirgt, ist die enorme Leistungsfähigkeit des DataWindow und seinen Merkmalen...

1Datenformatierung 1maskierte Dateneingabe 1,SaveAs' in jedem unterstützten Format, auch XML 1leistungsfähige Berichtsoptionen wie Gruppen und zusammenfassende Gruppen – verschiedenste Berichtformate, Raster, Freiform, Kreuztabellierung, Graphen usw. 1Aktualisieren der Datenbank einschließlich gespeicherter Prozeduren 1DataWindow-Ausdrücke 1Sortieren 1Filterung 1und vieles, vieles mehr!

Mit DataWindow .NET bietet Sybase Visual-Studio-Entwicklern eine Alternative zu DataGrid, DataView und anderen Microsoft-Objekten an. PowerBuilder-Programmierern, die .NET-Technologie in Anwendungen verwenden müssen, stellt Sybase ein Instrumentarium zur Verfügung, das die bereits gewonnene DataWindow-Erfahrung in der neuen Umgebung nutzbar macht.

## Frisch aus der Presse...

### Veranstaltungen: Jetzt vormerken!

**18. November 2003 – Norwegen:** Sybase-Seminar. Informationen und Anmeldung unter [www.sybase.no](http://www.sybase.no)

**3. Dezember 2003 – GB: STUN-Konferenz.** Informationen und Anmeldung unter [www.stun.org.uk](http://www.stun.org.uk)

**4. Dezember 2003 – Dänemark: PBUG-Meeting.** Informationen und Anmeldung unter [www.ravenholm.dk](http://www.ravenholm.dk)

**9. Dezember 2003 – Belgien: Sybase Techserie, Kinopolis Brussels.** Informationen und Anmeldung unter [www.sybase.be](http://www.sybase.be)

**10. Dezember 2003 – Luxemburg: PB-9-Seminar, Sofitel Luxembourg.** Informationen und Anmeldung unter [www.sybase.be](http://www.sybase.be)

### Neue Bücher zu PB 9:

Es gibt zwei neue Bücher zu PowerBuilder 9.0, geschrieben und herausgegeben von Experten auf diesem Gebiet:

**PowerBuilder 9 Advanced Client/Server Development**, Bruce Armstrong, Millard F. Brown III

**PowerBuilder 9 Internet and Distributed Application Development**, William Green, John D. Olson

Ebenfalls lieferbar: **Mobile and Wireless Design Essentials** von Martyn Mallick, Product Manager, iAnywhere Solutions Das erste und bisher einzige Buch, das eine detaillierte Analyse sämtlicher aktuellen Mobil- und Drahtlostechnologien aus der Perspektive des Softwareentwicklers liefert.

### NEU: POWERBUILDER 9.0 SCHULUNGEN

**DEV119 - Fast Track: Developing Robust Applications using PowerBuilder 9.0**

**DEV749 - PowerBuilder Masters Boot Camp**

**DEV542 - Mastering the DataWindow Object in PowerBuilder 9.0**

**DEV537 - Integrating XML, Web Services, and .NET with PowerBuilder**

**DEV135 - Fast Track to Pocket PowerBuilder Development**

**DEV333 - Building Wireless Enterprise Applications Using Pocket PowerBuilder**

**SYBASE EXPERT CENTER DÜSSELDORF**

[www.education.sybase.de](http://www.education.sybase.de)

Tel. 0211 5976 510 [esc\\_germany@sybase.com](mailto:esc_germany@sybase.com)

## Kostenlose Evaluations-CD

Rufen Sie uns noch heute an, um Ihre PB- und PPB-Lizenzen zu erwerben oder eine GRATIS-Evaluations-CD anzufordern. Beides können Sie auch auf unserer Website tun:

[www.softwarecentre.sybase.com](http://www.softwarecentre.sybase.com)

Deutschland Tel 069 9508 6182 email [esc\\_germany@sybase.com](mailto:esc_germany@sybase.com)  
Schweiz Tel 01 800 9220 email [esc\\_switzerland@sybase.com](mailto:esc_switzerland@sybase.com)

## Aktuelle Versionen/EBFs

BEZEICHNUNG	VERSION	DATUM	TYP
PB Enterprise - 9.0.1 (6533) Maintenance Release	9.0.1	05 Sep 2003	Maint/Update
PB Professional - 9.0.1 (6533) Maintenance Release	9.0.1	05 Sep 2003	Maint/Update
PB Desktop - 9.0.1 (6533) Maintenance Release	9.0.1	05 Sep 2003	Maint/Update
PB Enterprise - 9.0.1 (6533) Localized Runtime Files	9.0.1	08 Oct 2003	Maint/Update
PB Enterprise - 9.0.1 (6533) Localized PFC Files	9.0.1	08 Oct 2003	Maint/Update
PB Enterprise - 8.0.4 (10501) Maintenance Release	8.0.4	06 Oct 2003	Maint/Update
PB Professional - 8.0.4 (10501) Maintenance Release	8.0.4	08 Oct 2003	Maint/Update
PB Desktop - 8.0.4 (10501) Maintenance Release	8.0.4	08 Oct 2003	Maint/Update
PB Enterprise - 7.0.3 (10312) EBF Release	7.0.3	20 Aug 2003	EBF/Patch

Hinweis: PB v7.x EBF Build ist die endgültige Lieferung dieser Version, die am 15.7.2003 auslief.

Hinweis: Weitere Informationen unter [www.downloads.sybase.com](http://www.downloads.sybase.com)

## Pocket PowerBuilder, SOAP und PocketSOAP *Ian Thain, Technical Evangelist, IPG, Sybase Inc.*

Sie haben verschiedenartige Systemkomponenten in Ihrem Unternehmen und müssen mit Ihrer Pocket-PowerBuilder-Anwendung darauf zugreifen? Ich will mich an dieser Stelle nicht lange mit SOAP und Web-Services aufhalten, da zu diesen Themen kein Mangel an Lesestoff herrscht – genug für schlaflose Nächte. Worum es mir hier geht, ist das Umfeld der Technologien SOAP und Web-Services und was wir daraus machen können.

### SOAP

Das Simple Object Access Protocol (SOAP) gilt allgemein als das Rückgrat einer neuen Generation plattform- und sprachenübergreifender, verteilter Anwendungen, nämlich der Web-Services. Wie sein Name erwarten lässt, ist das SOAP ein ‚schlankes‘, XML-basiertes Protokoll für den Informationsaustausch in verteilten Umgebungen. Das SOAP-Protokoll besteht aus drei Teilen: Einer "Hülle" (**Envelope**), die Rahmenbedingungen für die Definition des Inhalts und der Verarbeitung der Nachricht festlegt, einem Satz von **Codierungsregeln** für Datentypen und schließlich einer **Konvention** für Prozeduraufrufe und -antworten.

### Web-Services

Die öffentlichen Schnittstellen und Bindungen eines Web-Service werden mit XML beschrieben und definiert und haben zur Identifikation gemeinsam eine URL. Zur Definition und Beschreibung dient die WSDL (Web Services Definition Language); die Erkennung durch andere Softwaresysteme erfolgt über UDDI. Nach der Erkennung des Web-Service findet die Interaktion anhand von XML-basierten Nachrichten statt, die über Internetprotokolle übermittelt werden, wobei die Web-Service-Definition die Modalitäten regelt.

### PocketSOAP

PocketSOAP ist eine Komponente des Component Object Model (COM) für Open-Source-SOAP-Clients und wurde ursprünglich für den PocketPC konzipiert; inzwischen wurde PocketSOAP außer für Pocket PC auch für Windows 95, 98, Me, NT4.0 und 2000 entwickelt. PocketSOAP kann entfernte Prozeduraufrufe an SOAP-Server richten, die mit 4s4c, ROPE, Apache SOAP, SOAP::Lite, DM SOAP/Perl implementiert sind, sowie an den XMethods-SOAP-Server. Zu dem Paket gehört ein HTTP-Client für HTTP-basierte SOAP-Anforderungen, andere Transportprotokolle können hinzugefügt werden. PocketSOAP wird unter der Mozilla Public License (MPL) vertrieben und unterstützt u. a. Verschlüsselung nach SOAP Section 5, Dokument- und Literal-SOAP-Dienste (z. B. ASP.NET), Anhänge sowohl über DIME als auch über ‚SOAP with Attachments‘, SOAP, HTTP 1.1 einschließlich persistente Verbindungen, SSL, Proxies, Authentifizierung, Proxy-Authentifizierung, Umleitungen, Cookies und Komprimierung. Weitere unterstützte Funktionen und Verfahren finden Sie unter <http://www.pocketsoap.com>

### Pocket PowerBuilder und PocketSOAP

Die Interaktion zwischen Pocket PowerBuilder und PocketSOAP erfolgt über die PPB-eigene Schnittstelle für PocketSOAP. Hierbei handelt es sich um einen externen DLL-Wrapper (da PPB über kein COM-Objekt oder "OLEObject" verfügt), der einen einfachen Zugang zu dem PocketSOAP-Paket über dessen COM-Schnittstelle ermöglicht. Das Modell für die API ist so konzipiert, dass zuerst das Service-Objekt erstellt wird und dieses dann den Zugriffs-Handle zurückgibt. Darauf folgend können Sie die diversen Attribute in beliebiger Reihenfolge definieren. Die Attribute werden einfach intern zur späteren Verwendung gespeichert. Dann lassen Sie den SOAP-Aufruf folgen. Dabei kommen die bereits definierten Attribute zur Anwendung. Abschließend zerstören Sie das Service-Objekt, wodurch alle zuvor erstellten COM-Objekte verfallen. All dies setzt auf das PocketSOAP-System auf. Um mit PocketSOAP zu arbeiten, rufen Sie die Installationsdateien von <http://www.pocketsoap.com> ab und installieren sie auf dem Desktop und dem Pocket-PC-Gerät.



#### EAServer

Betrachten wir nun ein Anwendungsbeispiel von PocketSOAP innerhalb von Pocket PowerBuilder; hier wird ein Web-Service mit EAServer 4.2 aufgerufen. Bei diesem



Web-Service handelt es sich um eine EJB, die alte Produktpreise über einen Verbindungs-Cache aus einer ASA-Datenbank abrufen. Die EJB wird mit dem Web Services Toolkit von EAServer als Web-Service definiert. Nach diesem Verfahren können wir beliebige Komponenten in EAServer als Web-Services bereitstellen. Im Rahmen der WST wird der gesamte benötigte WSDL-Code generiert.

Über diesen Code wird durch PocketSOAP auf den Web-Service zugegriffen.

```
string sMethod = "getPFShareList"
string sEndPoint =
"http://localhost:10000/WEBSERVICES/SOAP"
string sNamespace = "PFShareListAll/PFShareList"
int cRetBufLen = 128
string sArgs
string sResult
long lHandle
int iRet

// create & set attributes on for the soap call
iRet = PocketSoap_Create( true, REF lHandle )
wf_message( iRet, "PocketSoap_Create" )

iRet = PocketSoap_SetEndPoint( lHandle, sEndPoint )
wf_message( iRet, "PocketSoap_SetEndPoint" )

// preallocate the result string
sResult = Space( cRetBufLen )

// make a "simple call"
iRet = PocketSoap_SimpleCall( lHandle, sNamespace,
sMethod, "", "" )
wf_message( iRet, "PocketSoap_SimpleCall" )

// get the Result
iRet = PocketSoap_GetResult( lHandle, cRetBufLen, REF
sResult )
wf_message( iRet, "PocketSoap_GetResult" )

// clean up
iRet = PocketSoap_Destroy( lHandle )
wf_message( iRet, "PocketSoap_Destroy" )

wf_message( integer ai_ret, string as_method )
If ai_ret <> 0 Then
MessageBox("Return code", String(ai_ret) + " " +
as_method )
End if
```

Die drei wichtigen Elemente, die wir einstellen müssen, sind EndPoint, Namespace und Method. Der **EndPoint** spezifiziert einen Service-Zugriffsort unter Angabe eines Protokolls und Datumsformats. **Namespace** spezifiziert den jeweiligen Web-Service und natürlich die **Methode (Method)**, die innerhalb des Web-Service aufzurufen ist. In unserem Beispiel benutzen wir die Methode PocketSoap\_SimpleCall, die mit einem einzigen Argument einen synchronen Aufruf an die EndPoint-URL ausführt. Das Argument ist ein Namen-Wert-Paar und ist wahlfrei. Für komplexere Aufrufe gibt es die Methode PocketSoap\_Call, die Argumente ebenfalls als Name-Wert-Paare annimmt, jedoch in einem einzigen String darstellt.

```
Name1 ~t value1 ~t data-type1 ~r~n
Name2 ~t value2 ~t data-type2 ~r~n
Name3 ~t value1 ~t data-type1 ~r~n
Etc
```

PocketSOAP übernimmt die Generierung der SOAP-Nachricht und parst die Antwort. Der SOAP-Aufruf und die Antwort des Servers lassen sich mit dem Dienstprogramm SOAP Util von Apache (<http://www.apache.org>) anzeigen. Um dieses zu starten, verwenden Sie die Befehlszeile:

```
java org.apache.soap.util.net.TcpTunnelGui
10000 ithain-home 8080
```

Dieser Befehl veranlasst das Dienstprogramm, alle Anforderungen abzufangen, die an den localhost:10000 gerichtet werden, sie anzuzeigen und sie dann an ithain-home:8080 weiterzuleiten. Entsprechend fängt das Dienstprogramm die Antwort von ithain-home:8080 ab, zeigt sie an und



leitet sie an den Aufrufer weiter. Für EAServer 4.x brauchen wir das Attribut SoapAction. Dies ist damit begründet, dass wir in EAServer 4.x in unserer eigenen, proprietären

Implementierung von Web-Services dieses Attribut SoapAction verwenden; es ist jedoch kompatibel zu anderen Implementierungen, z. B. von Apache. Auch für andere Implementierungen ist SoapAction erforderlich. Kein Grund zur Beunruhigung – ab EAServer 5.0 erfolgt die Implementierung von Web-Services als "Axis", d. h. SoapAction wird nicht mehr gebraucht.



## XMethods

Hier ein Beispiel für den Aufruf eines Web-Service, gefunden auf XMethods (<http://www.xmethods.net>). Der Web-Service fordert eine Börsen-Preisauskunft an (20 Minute delayed Quote) und ist über GLUE implementiert. Auf XMethods sind viele weitere Web-Services aufgelistet, die mit MS .NET, Delphi, SOAPLite, WASP Server for Java, ColdFusion, BEA WebLogic, ApacheSOAP, AXIS und vielen anderen implementiert sind.

```
string sMethod = "getPFShareList"
string sEndPoint =
"http://localhost:10000/WEBSERVICES/SOAP"
string sNameSpace = "PFShareListAll/PFShareList"
int cRetBufLen = 128
string sArgs
string sResult
long lHandle
int iRet

// create & set attributes on for the soap call
iRet = PocketSoap_Create( true, REF lHandle )
wf_message( iRet, "PocketSoap_Create" )

iRet = PocketSoap_SetEndPoint( lHandle,
sEndPoint )
wf_message( iRet, "PocketSoap_SetEndPoint" )

// preallocate the result string
sResult = Space( cRetBufLen )

// make a "simple call"
iRet = PocketSoap_SimpleCall( lHandle,
sNameSpace, sMethod, "", "" )
wf_message( iRet, "PocketSoap_SimpleCall" )

// get the Result
iRet = PocketSoap_GetResult( lHandle,
cRetBufLen, REF sResult )
wf_message( iRet, "PocketSoap_GetResult" )

// clean up
iRet = PocketSoap_Destroy( lHandle )
wf_message( iRet, "PocketSoap_Destroy" )

wf_message (integer ai_ret, string as_method)
If ai_ret <> 0 Then
MessageBox("Return code", String(ai_ret) + " "
+ as_method)
End if
```

In diesem Fall wurde das Dienstprogramm SOAP Util mit folgender Befehlszeile gestartet:

```
java org.apache.soap.util.net.TcpTunnelGui
10000 services.xmethods.net 9090
```

Als EndPoint wurde der String sEndPoint = "http://localhost:10000/" anstelle von http://services.xmethods.net/soap im obigen Codebeispiel benutzt. Wie Sie an der Anforderung sehen, kommt diese Implementierung ohne das Attribut 'SoapAction' aus.



## Fazit

In den letzten Jahren haben wir Ihnen immer wieder empfohlen, Ihre Anwendungen in Abschnitte zu unterteilen, Code wiederzuverwenden und die Vorteile Ihres bewährten Anwendungsservers (EAServer) zu nutzen. Mit PocketSOAP können Sie auf Komponenten zugreifen, die als Web-Services bereitgestellt sind. PocketSOAP ist von Pocket-PowerBuilder-Anwendungen aus über eine PPB-Schnittstelle zugänglich.

Dieser Artikel wurde ursprünglich im PowerBuilder Developer's Journal (Band 10, Ausgabe 8) veröffentlicht. Um das PBDJ zu abonnieren, lassen Sie sich bei [subscribe@sys-con.com](mailto:subscribe@sys-con.com) registrieren. Weiter Informationen finden Sie unter [www.sys-con.com](http://www.sys-con.com).