

# Pylon Application Server Developer Guide

Version 6.0



## Company information

### About iAnywhere Solutions®

iAnywhere Solutions, Inc. is the leading provider of mobile enterprise software. iAnywhere solutions automate business processes and improve the exchange of information between companies and their employees and customers, resulting in improved business efficiencies and more effective customer interactions. iAnywhere solutions include the best-of-breed mobile software platform for creating and deploying enterprise applications, proven customer relationship management (CRM), supply chain management (SCM) and personal information management (PIM) mobile business applications, and professional services to assist in the strategy, design and delivery of end-to-end mobile solutions. iAnywhere's customers, which include 28 of the Fortune 100 and over 2500 companies in 50 countries, include McKesson Corporation, Ford Motor Company, Cisco Systems, FedEx, Scottish Power, The US Senate, Tyco, Harvard Medical School, American Express, The World Economic Forum, BG Group, plc., Genencor and Alcatel. For more information, please visit <http://www.ianywhere.com>.

### Contacting iAnywhere Solutions

If you need assistance using iAnywhere software, please contact iAnywhere Technical Support at [corp\\_support@ianywhere.com](mailto:corp_support@ianywhere.com); services will be provided in accordance with your support agreement.

If you need help with customizing iAnywhere software to better serve your enterprise, please contact AvantGo Client Services at [contact\\_us@ianywhere.com](mailto:contact_us@ianywhere.com).

If you need information about other iAnywhere products for your enterprise, please contact iAnywhere Workforce Sales at [contact\\_us@ianywhere.com](mailto:contact_us@ianywhere.com).

If you have questions or suggestions about this document or other iAnywhere technical publications, please contact iAnywhere Technical Publications at [iasdoc@ianywhere.com](mailto:iasdoc@ianywhere.com).

### Pylon Application Server Developer Guide

© iAnywhere Solutions, Inc. All rights reserved. All unpublished rights reserved.

iAnywhere and M-Business Server are registered trademarks or trademarks of Sybase, Inc. or its subsidiaries. The ® symbol indicates registration in the United States of America. All other company and product names mentioned may be trademarks of the companies with which they are associated.

A portion of this software is based in part on the work of the Independent JPEG Group and on the work of the Netscape Communications' Netscape Communicator & Directory Server 5.0 (Mozilla) open source project.

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org>).

This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by iAnywhere Solutions, Inc. iAnywhere Solutions, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means — electronic, mechanical, recording, or otherwise — without the prior written permission of iAnywhere Solutions, Inc.

Written and designed at iAnywhere Solutions, Inc.

One Sybase Drive  
Dublin, CA 94568

<http://www.ianywhere.com>

# Table of Contents

## Before you begin

---

<b>CHAPTER 1. About this guide</b> . . . . .	13
Focus of this guide . . . . .	14
How this guide is organized . . . . .	15
Conventions . . . . .	16
Related publications . . . . .	17
<b>CHAPTER 2. Introduction</b> . . . . .	19
About Pylon Application Server . . . . .	20
When to use this product . . . . .	20
Instant cross platform Notes applications . . . . .	20
Server management . . . . .	21
Device usability . . . . .	21
Synchronization . . . . .	21
Architecture . . . . .	22
Server . . . . .	22
Client . . . . .	23
Data detail . . . . .	24
Data flow . . . . .	24
What's new in release 6.0? . . . . .	25

## Concepts and development guidelines

---

<b>CHAPTER 3. About M-Business Anywhere</b> . . . . .	29
M-Business Anywhere Overview . . . . .	30
Architecture . . . . .	31
About M-Business Server . . . . .	32
M-Business database conduit overview . . . . .	33
About M-Business Client . . . . .	34
Components of M-Business Client . . . . .	35

<b>CHAPTER 4. Creating device-friendly Notes databases</b> . . . . .	37
About Pylon Application Server. . . . .	38
General design guidelines . . . . .	40
Forms . . . . .	41
Design considerations . . . . .	42
Checkbox field columns. . . . .	42
Computing editable fields . . . . .	42
Date/time fields . . . . .	42
JavaScript errors . . . . .	42
Maximum number of fields. . . . .	42
Multiple checkboxes . . . . .	42
On-device database names. . . . .	43
True validation for rich text. . . . .	43
Views. . . . .	44
Icons . . . . .	45
Supported design objects. . . . .	46
Use Case: Agent sync . . . . .	47
Unsupported design objects . . . . .	48
<b>CHAPTER 5. Supported form field types</b> . . . . .	49
Overview . . . . .	50
Pylon Application Server Field Type Matrix: from Lotus Notes to/from UI Types. . . . .	51
Generating choices for lists . . . . .	57
Field labels support . . . . .	57
Keyword synonyms . . . . .	58
Null field values. . . . .	58
Readers fields and Pylon Application Server Lotus Notes ID file . . . . .	59
Authors Access. . . . .	60
<b>CHAPTER 6. JavaScript support</b> . . . . .	61
Overview . . . . .	62
Table of JavaScript events . . . . .	63
Events that work both in browsers and the Notes client . . . . .	65
JSHeader event . . . . .	65
onSubmit event . . . . .	65
Enabling JavaScript in the Lotus Notes client . . . . .	66
Event descriptions . . . . .	66
About sample code . . . . .	67
Pylon Application Server demo databases. . . . .	67
Form validation sample: validating a form for incomplete fields . . . . .	68
Computed field validation sample: validating a form for incomplete fields. . . . .	69
<b>CHAPTER 7. Implementing on-device database lookups</b> . . . . .	71
About on-device database lookups . . . . .	72

---

Creating a lookup field in a Lotus Notes database . . . . .	73
Creating a lookup field mapping . . . . .	74
Retrieving additional fields . . . . .	74
Synchronization issues . . . . .	75

## Sample implementations

---

<b>APPENDIX A. Sample databases . . . . .</b>	<b>79</b>
Overview . . . . .	80
Database configuration . . . . .	81
Database descriptions . . . . .	82
Accounts: accountsA.nsf . . . . .	82
Companies: companyA.nsf . . . . .	83
Inventory: inventoryA.nsf . . . . .	84
Orders: ordersA.nsf . . . . .	85
Setting up the databases . . . . .	86
Accounts . . . . .	86
Company . . . . .	86
Products . . . . .	86
Sales Orders2 . . . . .	86
Using the databases . . . . .	87
Accounts . . . . .	87
Companies . . . . .	87
Inventory . . . . .	87
Orders . . . . .	87
<b>Index . . . . .</b>	<b>89</b>



# List of Tables

Table 1-1	Formatting conventions . . . . .	16
Table 5-1	Supported/unsupported Form field type labels . . . . .	50
Table 5-2	Pylon Application Server Field Type Matrix . . . . .	51
Table 6-1	Supported JavaScript events . . . . .	62
Table 6-2	JavaScript event handlers in Lotus Domino Designer . . . . .	63
Table 6-3	JavaScript events . . . . .	66
Table 6-4	Demo database View and Form names . . . . .	67



# List of Figures

Figure 2-1	Server architecture . . . . .	22
Figure 2-2	Data detail . . . . .	24
Figure 2-3	Data flow . . . . .	24
Figure 3-1	M-Business Anywhere Architecture . . . . .	31
Figure 4-1	Database icon in Lotus Notes . . . . .	45
Figure 4-2	Database icon on device . . . . .	45
Figure 7-1	Creating a Lookup Field . . . . .	73
Figure 7-2	Creating a lookup field mapping . . . . .	74
Figure 7-3	Retrieving additional lookup fields . . . . .	75



# Before you begin

“About this guide” (page 13)

“Introduction” (page 19)



# CHAPTER 1. **About this guide**

---

---

- “Focus of this guide” (page 14)
- “How this guide is organized” (page 15)
- “Conventions” (page 16)
- “Related publications” (page 17)

## Focus of this guide

This Developer Guide provides you, the Lotus Notes developer, with information about Lotus Notes functionality that Pylon Application Server supports and how you can transfer that functionality onto the device.

## How this guide is organized

This guide is divided into two major sections: “Concepts and Development Guidelines” and “Sample Implementations” in the Appendix for reference use. Following is a list of each chapter and appendix included in this book, each followed by a short summary of its contents.

- CHAPTER 1., “About this guide” describes the focus of this book, its target audience, the book’s organization, typographical conventions used, and lists related publications.
- CHAPTER 2., “Introduction” provides you with an overview of Apollo, disusses supported events, and provides sample databases.
- CHAPTER 3., “About M-Business Anywhere” provides you with an overview of the M-Business Anywhere application.
- CHAPTER 4., “Creating device-friendly Notes databases” shows you how to create Notes databases, provides you with general design guidelines, Form, View, and Icon information, and lists unsupported design objects.
- CHAPTER 5., “Supported form field types” discusses form field types, generating choices for lists, and keyword synonyms. A Field Data Type matrix summarizes data type mappings between the Lotus Notes Form, the Lotus Notes Data Adaptor, and the Device.
- CHAPTER 6., “JavaScript support” lists supported JavaScript events and tells you how to download and access the JavaScript Engine sample files . Code samples are provided to demonstrate how to validate an incomplete form and how to validate a computed field.
- CHAPTER 7., “Implementing on-device database lookups” describes on-device database lookups and how to create a lookup field in a Lotus Notes database.
- APPENDIX A., “Sample databases” provides you with sample databases to study.

## Conventions

The following table lists the formatting conventions used throughout this guide.

**Table 1-1**  
Formatting  
conventions

Item	Treatment	Example
Name of publication	Italic	<i>Administrator Guide for M-Business Server</i>
User interface items: buttons, links, keywords	Bold	Click the <b>Reset</b> button.
Multi-level menu selections	Bold with vertical hash	Select <b>Start Settings  Control Panel</b> .
Text you type	Bold fixed width font	Type <b>Admin</b> in this field.
Text displayed in a file or on the screen	Fixed width font	The screen reads: Backup Complete
Variables	Angle brackets	http://<servername>:<port>
Keyboard key	Angle brackets	<Enter>
File names and paths	Italic	<i>pods.h</i> <i>C:/Program Files</i>

## Related publications

In addition to this document, there are several other iAnywhere Solutions publications available that you may find useful in setting up and using Pylon Application Server.

- *User Guide for M-Business Client*
- *Ensuring Mobile Security with AvantGo Technology: From the Device to the Datacenter*
- *Channel Developer Guide for M-Business Anywhere*
- *Dynamic Mobility Model Handbook*
- *Developer Guide — M-Business JavaScript Engine, M-Business Client Extension API, M-Business Database API*

All of these publications are available from the URL below:

[http://www.ianywhere.com/developer/product\\_manuals](http://www.ianywhere.com/developer/product_manuals)



## CHAPTER 2. Introduction

---

- “About Pylon Application Server” (page 20)
- “Architecture” (page 22)
- “What’s new in release 6.0?” (page 25)

## About Pylon Application Server

Pylon Application Server 6.0 is the new application server from iAnywhere Solutions that mobilizes Lotus Notes databases to handheld devices. Pylon Application Server provides a robust enterprise platform on which to synchronize data between Lotus Notes databases on the Domino server and the mobile device: enabling true mobile Lotus Notes applications. Reasons to mobilize your Lotus Notes applications include:

- Improved communication reduces cost and increases profitability
- Extended competitive advantage and revenue potential with more responsive customer service
- Increased ROI of existing Lotus Notes systems
- Mobilized Lotus Notes applications improve workforce efficiencies
- Organizations are moving away from local to server synchronization - for central management

Pylon Application Server supports JavaScript in the Lotus Notes databases. The JavaScript code lines are transferred to the mobile device and executed there as event triggers: the same as on the Lotus Notes desktop.

### When to use this product

Use Pylon Application Server when you want to mobilize Lotus Notes applications or create wireless or centralized applications

### Instant cross platform Notes applications

Use Pylon Application Server to deploy Lotus Notes applications and databases quickly and easily using a Web-based architecture. Benefits include:

- Instant cross-platform deployment
- Development using Lotus Domino Designer
- Mobilization of simple views/forms with no programming
- Full look and feel of the Lotus Notes workspace on the device
- Lotus Notes ACL and security model support
- Easy expansion of Lotus Domino mobilization to cover other databases, Web content and applications, and XML Web services

## Server management

Pylon Application Server management features include:

- A Web-based management console that provides centralized management of Applications, Users and Groups
- Group level administrator support and optional self-service application deployment portal
- Automatic remote application updates
- Mobilized databases and applications on multiple Domino Servers
- Tracking and reporting of key usage and performance indicators

Pylon Application Server security features include:

- Secure encrypted wireless transmission using 128 bit SSL and 163-bit ECC
- Integration with Domino ACL to support Domino security models

## Device usability

Pylon Application Server device usability enhancements enable you to:

- Build “smart client” applications that don’t depend on wireless networks
- Support JavaScript on device for device-side business logic
- Sort, Search and resize data columns on device allows users to find and display the most relevant information conveniently
- Use an intuitive user interface including ‘Notes Workspace’
- Support offline database lookups on-device

## Synchronization

With Pylon Application Server, you can:

- Run agents at sync time for server-side error checking
- Synchronize devices from desktop, wireless or any TCP/IP connection
- Optimize synchronization to minimize wireless data access fees

## Architecture

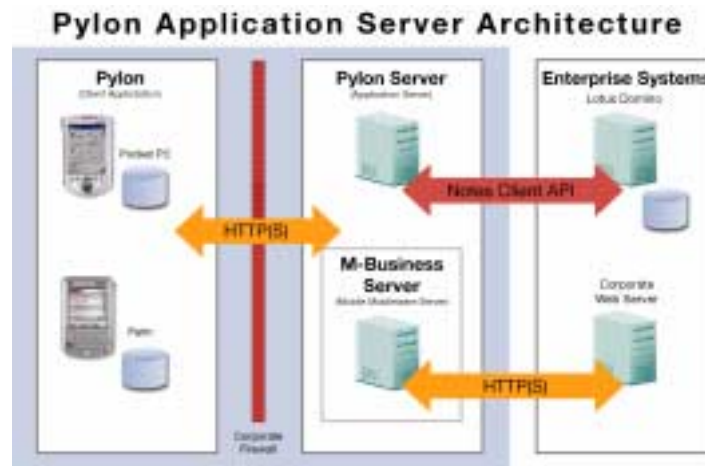
### Server

Pylon Application Server deploys Lotus Notes applications using your existing Web infrastructure and a Lotus Domino adapter. You get everything from Pylon Application Server and:

- A new look and navigation on-device
- On-device logic using JavaScript
- Field validation
- Form calculations
- Lookups like Lotus Notes
- Computed fields
- Non-editable fields
- Hidden fields

Refer to Figure 2-1 below, for an overview of the Pylon Application Server architecture.

Figure2-1  
Server architecture



### Server requirements

You will need the following to install Pylon Application Server:

- 1GB RAM and 2GB disk space (disk space and RAM requirements will vary, depending on user population)
- 500 MB free before install
- Pentium class machine
- Windows 2000 Server with Service Pack 4
- Windows 2003 Server

- Lotus Notes Client, version 5.0.10 or later installed on the server, single user install

### ***System requirements***

Pylon Application Server requires one of the following operating systems:

- Windows 2000 Server with Service Pack 4
- Windows 2003 Server
- Lotus Notes Client, version 5.0.10 or later

## **Client**

Dynamic, data-driven mobile applications can now be delivered on-device with:

- On-device dynamic presentation using HTML 4.01, XHTML 1.0, and CSS 1.0 style attribute for a dynamic user interface
- On-device XML data synchronization plus a cross-platform JavaScript API
- On-device logic using JavaScript and Document Object Model (DOM) to control presentation and access to data.

Client components include the following:

- M-Business and M-Business Connect
- Accessing custom Notes databases on the device
- On-Device logic using JavaScript
- Demonstration of sample databases

### ***System requirements***

Pylon Application Server requires one of the following operating systems on the client:

- Palm OS 5.0, or
- Pocket PC 2000 and later

### ***Supported platforms***

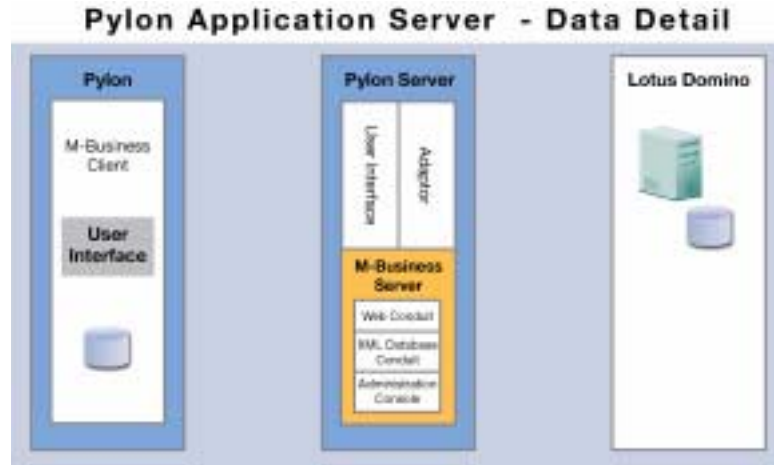
Pylon Application Server is supported on the following platforms:

- Notes/Domino v 5.x
- Notes/Domino v 6.x

## Data detail

Refer to Figure 2-2 below, for a detailed view of how Pylon Application Server manages data.

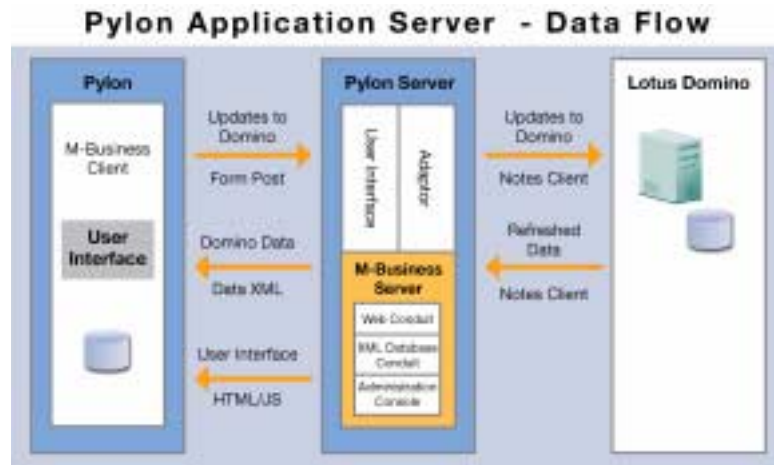
Figure 2-2  
Data detail



## Data flow

Refer to Figure 2-3 below, for an overview of the data flow between the client, the server, and the database.

Figure 2-3  
Data flow



## What's new in release 6.0?

The following features have been added to Pylon Application Server:

- Expanded developer options
- Built on M-Business Anywhere foundation
- Enhanced Lotus-like on-device user interface
- Data limits removed

**Expanded developer options**, making it easier to develop new mobile Lotus Notes applications or mobilize existing desktop Notes applications:

- On-device scripting via JavaScript. Script client-side business logic using JavaScript in Lotus Domino Designer.
- Lotus Domino Designer lookups function on-device exactly as in Lotus Notes, with the addition of “multi-function lookup.” When the user taps on a lookup field, go to the other database, allow user to search and pick, then go back to first database and populate multiple fields.
- Scanning capture can be built into forms.

**Built on M-Business Anywhere foundation.** Robust, mature platform for mobilizing applications and data using a web architecture.

- M-Business Anywhere is now used to perform user and group maintenance and application deployment for Pylon Application Server.
- Administrative functions integrated into M-Business Anywhere administrator console.
- Configuration of filters done in M-Business Anywhere administrator console forms of a Notes database.

**Enhanced, Lotus-like on-device user interface:**

- Standard HTML form widget support (drop-downs, text fields, radio buttons, checkboxes, etc.).
- Compound Search/Find within a view.
- Multiple Lotus Notes bitmaps per rich text field.
- Multiple rich text fields per form.
- Improved support for hidden fields.
- Improved support for multi-value fields.
- Improved on-device navigation via tabs and drop-down view menu.
- Improved conflict resolution options, with detection at field level and user control of resolution.
- Support for Lotus Domino hierarchical ACL groups.
- Support for read-only databases and forms.

**Data limits removed:**

- 20 databases (views)
- 128 fields per form

# Concepts and development guidelines

“About M-Business Anywhere” (page 29)

“Creating device-friendly Notes databases” (page 37)

“Supported form field types” (page 49)

“JavaScript support” (page 61)

“Implementing on-device database lookups”  
(page 71)



## CHAPTER 3. **About M-Business Anywhere**

---

---

- “M-Business Anywhere Overview” (page 30)
- “Architecture” (page 31)
- “About M-Business Server” (page 32)
- “M-Business database conduit overview” (page 33)
- “About M-Business Client” (page 34)
- “Components of M-Business Client” (page 35)

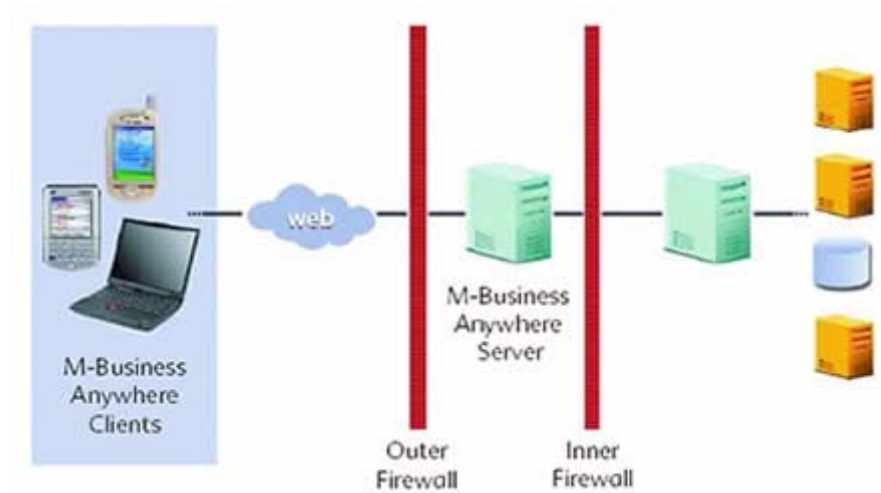
## M-Business Anywhere Overview

M-Business Anywhere is embedded in Pylon Application Server. It provides a platform for delivering Web-based content and applications to mobile devices rapidly and cost-effectively -- with minimal recoding. Web developers can leverage their existing skill sets and open standards to develop and deploy fully interactive Web applications with sync-and-go or wireless capabilities. So no matter where end users are, they'll be able to tap into the information, applications and Web services they need to make more effective business decisions, more quickly than ever before. Just as importantly, you can develop, deploy, and maintain mobile Web applications rapidly and with the lowest total cost of ownership (TCO) possible.

## Architecture

The following architecture diagram identifies the three major platform components: client, server, and datastore. A typical M-Business Server environment supports clients that are connected via desktop computers, via kiosks, or via wired or wireless modems.

**Figure 3-1**  
M-Business  
Anywhere  
Architecture



## About M-Business Server

With M-Business Server, system administrators can centrally manage information and administer data access for professionals who take critical business information and applications with them on their mobile devices. M-Business Server administrators extend user-interactive Web, intranet, or corporate database content and applications to mobile devices.

### *M-Business Admin Server*

The M-Business Admin Server (also referred to as the UI Server) provides authorized administrators and users access to M-Business Server through a standard desktop Web browser.

### *M-Business Sync Server*

The Sync Server handles requests from M-Business Client to perform synchronization functions.

### *Data store*

The data store is the location where both the Admin Server and the Sync Server store information about users, groups, channels, conduit settings, and server settings, as well as information about devices that have synchronized. M-Business Server ships with Adaptive Server Anywhere as its default data store.

## M-Business database conduit overview

M-Business Server Application Edition comes with M-Business database conduit. M-Business database conduit allows an M-Business Client application developer to send structured data to handheld devices, present that data in the M-Business Client browser, and to report changes to that data back to the originating database server through Pylon Application Server's standard form submission mechanism.

For more information on M-Business database conduit architecture and on creating applications for use with M-Business database conduit, see the *M-Business Developer Guide*, referenced in "Related publications" (page 18). For information on using the M-Business database conduit, see "Using database conduits" (page 151).

## About M-Business Client

M-Business Client is software that you install on your mobile device to provide you with mobile access to mission-critical applications and data.

M-Business Client works with M-Business Server to deliver these mobile applications and data to your device. When combined with powerful Mobile Engines for specific applications, M-Business Server provides the fastest, easiest, and most widely used way to deliver applications and data from backend systems to mobile devices based on the Palm, Microsoft Pocket PC, Windows XP, and RIM operating systems.

You can install and configure M-Business Client for yourself. Your system administrator installs and configures M-Business Server.

Either you register this account yourself, or your system administrator creates it for you. If you do not know which option you should use, check with your system administrator.

## Components of M-Business Client

M-Business Client consists of software for your desktop computer and for your mobile device.

### *M-Business Connect*

The desktop component of M-Business Client is M-Business Connect, which is the conduit that allows you to sync your device with M-Business Server. The settings in M-Business Connect provide the information that M-Business Client uses to communicate with M-Business Server.

If you use a Palm OS, Pocket PC OS, or Windows XP device, M-Business Connect also installs on your mobile device, allowing you to configure settings for M-Business Server directly on your device and to sync remotely (if your device is equipped with a modem, network, or wireless connection).

If you use a RIM Wireless Handheld, you can configure M-Business Connect from your desktop only. M-Business Connect does not install on your mobile device.

### *M-Business Client on your mobile device*

On your device, M-Business Client provides access to rich, interactive data and Web content. Using M-Business Client, you can view Web-based enterprise applications and content.

### *Communication between your device and M-Business Server*

Your device can communicate with M-Business Server in one of two ways: through your desktop computer, when you use a cradle to sync, or through a remote connection, such as a wireless or modem connection.

If you use a Palm OS, Pocket PC OS, or Windows XP device, M-Business Client always uses M-Business Connect to communicate with M-Business Server.



## CHAPTER 4. **Creating device-friendly Notes databases**

---

---

- “About Pylon Application Server” (page 38)
- “General design guidelines” (page 40)
- “Forms” (page 41)
- “Supported design objects” (page 46)
- “Use Case: Agent sync” (page 47)
- “Unsupported design objects” (page 48)

## About Pylon Application Server

Use Pylon Application Server to instantly extend your Lotus Domino-based applications. In version 6.0:

1. Simple operation and intuitive end-user interface allows you to:
  - Re-sort and re-size data in database views to display the most relevant data
  - Search data using full find and compound find in database views
  - Synchronize devices from multiple desktops and wireless services
  - Access via offline services, wireless connections and desktop cradles
  - View Notes 'workspace' as intuitive offline mobile home page
2. Robust security lets you:
  - Leverage 128-bit SSL and 163-bit ECC encryption during data transfer
  - Control data views using existing ACLs through integration with Lotus Domino and Domino security models
3. Easy administration and management gives you the ability to:
  - Mobilize databases and applications on multiple Lotus Domino servers, including Unix systems
  - Centrally manage users, groups and applications with Web-based management console
  - Extend administration tasks to group level administrators and end-users through optional self-service application deployment portal
  - Push remote application updates automatically through single update on the server
  - Track and report on key usage and performance indicators using administration console
  - Run an agent at sync time for server-side data checking or manipulation
  - Easily upgrade to M-Business Anywhere to deploy and manage web content and applications and Domino applications from a single infrastructure . Upgrade requires no reinstallation.
4. Quick robust mobile application development lets you:
  - Convert Lotus Domino applications into cross-platform handheld applications using existing Views and Forms or use Domino Designer to build/customize mobile applications
  - Build "smart client" applications that don't depend on wireless network connections
  - Add scanning input to forms applications

- Use JavaScript for device-side business logic including field validation and form calculations
- Implement offline database lookups on-device to simplify user data entry and build robust offline applications
- Personalize offline home page for individuals or groups using Domino formulas or views

## General design guidelines

The following are some more general guidelines for creating device-friendly custom databases for use with Pylon Application Server.

- Only one View and one Form per Pylon Application Server database. To sync multiple views and forms from a single Lotus Notes database, configure them as separate Lotus Notes channels in the M-Business Admin UI. After syncing to the device, the Lotus Notes channels will show up as separate databases on-device. Changes or additions made to the Pylon databases on the device will be synced back to the database in Lotus Notes, but changes made in one Pylon database will not be reflected in another on-device database until they are synced back to the Domino server, even when they are from the same Lotus Notes database.
- Only four columns at a time can be viewed on the listpage. By default, the first four columns will be displayed. You can configure the listpage to display any of the columns available in the view in any combination (up to four) by using the Configure option from the Utils tab.
- Only columns from the synced form show on the listpage.
- Only hard-coded keyword values are supported. Formulas will be ignored.

## Forms

Each Pylon Application Server database on the Pocket PC or the Palm device can contain exactly one Form. Apollo supports the following field types:

- authors
- checkbox
- combobox
- date/time
- dialog list
- list box
- names
- number
- readers
- rich text
- rich text lite
- text
- Computed fields:
  - Computed text (read only)  
A computed field formula that calculates each time a user creates, saves, or refreshes a document.
  - Computed when composed (read only)  
A computed-when-composed field formula that calculates only once, when the user first creates the document.
- Lookup—for more information, see “Implementing on-device database lookups” (page 71).

## Design considerations

Read the following information discussing design-related issues and limitations.

### Checkbox field columns

Use a dialog list to simulate both single-value and multiple-value checkbox field functionality. The data for a dialog list is stored the same as a checkbox in Notes, and will display as comma-separated list in the list view and multi-select on the form.

### Computing editable fields

Use `onLoad JS` to get compute Editable fields functionality when a formula is available.

### Date/time fields

JavaScript is not supported for date/time fields.

### JavaScript errors

Poorly written JavaScript won't work and may cause the rest of the application to not work on the device. iAnywhere Solutions, Inc. recommends first debugging your JavaScript in Lotus Domino Designer.

### Maximum number of fields

Note that the number of fields on a form only affects time it takes for last field on a form to get the focus when user taps it. The default setting (256 fields) may be too high for users to tolerate. Use 50 fields as an effective maximum.

As a Lotus Notes developer, test the form and decide if it needs to be broken up to make time-to-focus on last field acceptable to customer users on their devices.

### *Text fields*

For a "regular" `text` field (meaning `non-rich text`), Pylon Application Server will sync up to 1022 characters to device.

### Multiple checkboxes

Multiple checkboxes are implemented as separate fields on the device. Any JavaScript that is applied to the checkbox field in Lotus Notes will apply individually for each selection when it is synced to the device. For

example, a JavaScript alert as an `OnFocus` event for the checkbox field will be displayed when each checkbox is selected.

### **On-device database names**

An on-device database name cannot exceed 26 characters nor contain white space.

### **True validation for rich text**

When a rich text field is placed in edit mode, it is considered empty even if it had a previous value before editing. Therefore, it will trigger any field level validation you add to it.

## Views

Each Pylon Application Server database on a Palm or Pocket PC OS device contains exactly one View. The View can contain up to four columns. For views with more than four columns, only four columns can be displayed at one time. Each column must be linked to a field name that resides on the currently mapped form.

Views are customizable on the device and Fields can be reassigned and column widths can be resized as you work. However, changes made to the View on the device are not synced back to Lotus Notes.

In the current release of Pylon Application Server, there is no support for formulas within view columns. However, if a computed column contains exactly one valid Field within a View's formula, that Field name will be extracted and used in the Pylon Application Server database. For example, if the formula is `@UpperCase(Name)`; the `@UpperCase` formula will not be applied, but the column will be populated with a Field called **Name**.

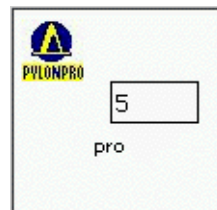
## Icons

Create a custom icon when designing your database. This icon will appear in the Lotus Notes Workspace as well as in Pylon Application Server on the device. To do this while in design mode, click **Other** | **Database Resources** | **Icon** in the Design pane.

**Figure 4-1**  
Database icon in  
Lotus Notes



**Figure 4-2**  
Database icon on  
device



## Supported design objects

The following design objects are supported on the device.

- ACL except for form section access control
- Agents when specified within the Lotus Notes channel configuration; any configured agents are executed after its corresponding database is synced.

For example, a typical sequence would be: db1 syncing, db1 closed, db1 agent executed; db2 syncing, db2 closed, db2 agent executed. Refer to "Use Case: Agent sync" below.

- Forms
- Form field labels
- Form fields -- most are supported
- JavaScript within the context of specific form and field level event handlers
- Views
- Formulas - in three contexts:
  - As an alternative to a Lotus Notes view name in the Lotus Notes channel configuration information
  - Within computed fields. However, computed fields are not computed until the document is synced back to the Lotus Domino server.
  - In View columns formulas will only work if only 1 field is specified. The formula will be ignored; it treats the column as a "Field" instead of "Formula." For example, `"@UpperCase ( Name ) ;"` - the `@Uppercase` will not be applied, but the column will be populated with the field called "Name".

## Use Case: Agent sync

You have a customer with a very large form called parent (lots of fields) that it uses for entering data in Lotus Notes, it wants to mobilize this form but only needs a subset of fields for data entry on the device. So it creates a child form which contains only the necessary fields. The child form is configured as a Lotus Notes channel to sync to the device.

**Problem:** When you open records that were created on the device in Lotus Notes, you don't have access to the additional fields that are in the parent form.

**Solution:** Run an agent to switch forms after the data is synced so that when the records are opened in notes, they open in the parent form.

```
forms: parent (notes form) and child (device form)
agent: formname - client formula action - FIELD Form
      := @If(Form =
           "child";"parent";Form);SELECT @All
```

---

---

**Note**

The fields on the child form must be shared fields that are also on the parent form.

## Unsupported design objects

The following design objects are not supported on the device:

- Access-controlled form sections
- Action bar and buttons (Pylon provides its own action buttons)
- Attachments
- Background images and background colors
- Categories
- Folders
- Form name synonyms
- Framesets
- Imagemaps
- LotusScript and Java
- Navigators
- OLE objects and custom controls
- Other resources, for example: Using Database documents, About Database documents, Database Script, and Shared Actions
- Outlines
- Pages
- Resources, for example: images, applets, subforms, shared fields, script libraries
- Synopsis

# CHAPTER 5. Supported form field types

---

---

- “Overview” (page 50)
- “Pylon Application Server Field Type Matrix: from Lotus Notes to/from UI Types” (page 51)
- “Readers fields and Pylon Application Server Lotus Notes ID file” (page 59)
- “Authors Access” (page 60)

## Overview

Pylon Application Server supports several form field types as options for defining a list field to generate. Refer to Table 5-1 below for a list of supported and unsupported form field type labels.

**Table 5-1**  
Supported/  
unsupported Form  
field type labels

Form field type label	Supported/Unsupported
Enter choices	Supported
Use formula for choices	Not supported
Use Address dialog for choices	Not supported
Use Access Control list for choices	Not supported
Use View dialog for choices	Supported for dialog list

---

---

**Note**

A hidden field attribute such as  
`UIType = hidden`  
would override any other setting for the UI type.

## Pylon Application Server Field Type Matrix: from Lotus Notes to/ from UI Types

Refer to the following table for a summary of Field type mappings between the Lotus Notes Form, the Lotus Notes Data Adaptor, and the Device. The Notes Form Field Type column lists the actual types of form fields available to the user within Lotus Domino Designer. The Supported Field Properties column lists the properties that Pylon Application Server processes for each Lotus Notes field type. The Handheld UI Type column lists the on-device UI Type to which each Lotus Notes form field is mapped. Any additional information on how Pylon Application Server supports the specified field type is listed in the Special Notes column. This table is followed by a discussion of how Pylon Application Server supports field labels, keyword synonyms, and null field values.

**Table 5-2** Pylon Application Server Field Type Matrix

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
authors	Notes v1.0 / v2.0	text field	
authors readers names	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Hide paragraph when document is (see 4)	text field	
authors readers names	allow multiple values	text field	downloads sends the text values all on one line separated by commas upload value is parsed for multiple comma-delimited values

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
checkbox	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices ■ Enter Choices Allow keyword synonyms Allow multiple values Hide paragraph when document is (see 4)	one or more checkbox fields: one for each specified choice	user can select 0 or more checkboxes
color	n/a	n/a	not supported
combobox	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices ■ Enter Choices Allow keyword synonyms Hide paragraph when document is (see 4)	select one	
date/time	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Display Date Display Time Hide paragraph when document is (see 4)	datetime field	
date/time	allow multiple values	datetime field	download only sends one date/time value to device upload will only assign one date/time value to the field

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
checkbox	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices ■ Enter Choices Allow keyword syn- onyms Allow multiple values Hide paragraph when document is (see 4)	one or more check- box fields: one for each specified choice	user can select 0 or more checkboxes
color	n/a	n/a	not supported
combobox	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices ■ Enter Choices Allow keyword syn- onyms Hide paragraph when document is (see 4)	select one	
date/time	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Display Date Display Time Hide paragraph when document is (see 4)	datetime field	
date/time	allow multiple values	datetime field	download only sends one date/time value to device upload will only assign one date/time value to the field

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
dialog list	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices: <ul style="list-style-type: none"> <li>■ Enter Choices</li> <li>■ Use View dialog for choices (see 6)</li> </ul> Allow keyword synonyms Hide paragraph when document is (see 4)	select one	
dialog list	allow multiple values	select multiple	user can select 0 or more values
formula	n/a	n/a	not supported
list box	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Choices <ul style="list-style-type: none"> <li>■ Enter Choices</li> </ul> Allow keyword synonyms Hide paragraph when document is (see 4)	select one	
list box	allow multiple values	select multiple	user can select zero or more values

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
number	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Number format Decimal places Hide paragraph when document is (see 4)	numeric field	Supports the double data type. Number Format Values: <ul style="list-style-type: none"> <li>■ Decimal: number with the specified decimal places</li> <li>■ Percent: number with the specified decimal places</li> <li>■ Scientific: number with all decimal places</li> <li>■ Currency: non-currency number with specified decimal places</li> </ul>
number	allow multiple values	text field	download sends field value as a single string that contains multiple numeric values separated by semicolons upload value is parsed for multiple semicolon-delimited numeric values EXAMPLE: 1;2;3;4;
password	n/a	n/a	not supported
rich text	Name Type Editable, Computed (see 2, 3) Hide paragraph when document is (see 4)	multi-line text field  Any successfully converted 1-bit or 16-bit Lotus Notes bitmaps will appear immediately below the text field on the read-only view document page. Images do not appear on the edit document page. (see 5)	only text is read/write to/from device images are read-only to device supports 1-bit and 16-bit Lotus Notes bitmaps to device on text-update from device to server, all existing original field images are maintained, but they are now appended below the field's updated text value on update from device to server, any existing rich text formatting for the item is disregarded

Notes Form Field Type	Supported Field Properties	Handheld UI Type	Special Notes
rich text lite	Name Type Editable Hide paragraph when document is (see 4)	multi-line text field  Any successfully converted 1-bit or 16-bit Lotus Notes bitmaps will appear immediately below the text field on the read-only view document page. Images do not appear on the edit document page (see 5).	only text is read/write to/from device images are read-only to device supports 1-bit and 16-bit Lotus Notes bitmaps to device on text-update from device to server, all existing original field images are maintained, but they are now appended below the field's updated text value on update from device to server, any existing rich text formatting for the item is disregarded
text	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Hide paragraph when document is (see 4)	text field	
text	allow multiple values	text field	download sends the text values all on one line separated by commas upload value is parsed for multiple comma-delimited text values EXAMPLE: lion,tiger,cougar
timezone	n/a	n/a	not supported
all other field types not mentioned above	Name Type Editable, Computed, Computed When Composed (see 1, 2, 3) Hide paragraph when document is (see 4)	text field	yes / no – attempts to treat all other types as type text

1. Any Lotus Notes field set with “Computed for display” is ignored by Pylon Application Server.
2. Computed fields are read-only on-device.

3. Computed fields are not computed until the document is synced back to the Lotus Domino server.
4. Check all of the checkboxes within the “Hide paragraph when document is” group within Lotus Domino Designer to hide a field on-device.

---

#### Note

“Hide paragraph from” properties are ignored.

5. For a specific application, if you want a Lotus Notes bitmap image to appear in a specific position on the device form relative to the text, such as above or below the text, store the bitmaps in separate rich text fields.
6. See the Pylon database lookup documentation for more information on how this is supported on-device.

---

#### Note

Supported date range: Pylon Application Server supports dates from 1902-2038. Dates or date/time values outside of this range will be blank on-device.

## Generating choices for lists

Note that only the Dialog list type contains the following option.

### *Use View dialog for choices*

Looking up values in a view lets you retrieve data from databases. This is convenient for displaying choices that change, such as customer names, sales territories, and job titles. This option brings up a dialog box containing entries from a column in a view. Select the database to look up, select a view, then select a column number.

View lookups provide the following benefits:

- Non-designers can maintain choices without having access to the rest of the database design.
- Designers can avoid hard-coding choices into fields. This makes verification and maintenance easier.
- Designers can hide the design of the database without affecting maintenance of choices.
- Designers or administrators can customize the application or conveniently translate lookup information to other languages.
- Users can review choices and codes more conveniently from outside the form or application.

## Field labels support

Lotus Notes field labels of up to 20 characters are supported on-device. If a Lotus Notes field label is greater than 20 characters then Pylon Application Server uses up to the first 32 characters of the field name as the on-device field label. If a Lotus Notes field has no label then the actual field name is used as the on-device label.

## Keyword synonyms

Pylon Application Server supports Lotus Notes form field keyword synonyms. On-device, only the keyword is available. When syncing, Pylon Application Server automatically maps keywords from/to keyword synonyms. When a keyword synonym is specified for a keyword, and that keyword is selected for a field value, the keyword synonym is stored within the Lotus Notes document and not the actual keyword. For example, if the following choice is defined for a dialog list field in Lotus Domino Designer,

```
Pylon Application Server | 2045
```

then when this choice is selected within the dialog list, "2045" is stored in the Lotus Notes document for that dialog list field. Keyword synonyms are supported for the following field types:

- listbox
- combobox
- checkbox
- dialog list
- radio button

## Null field values

On data download, all null field values for all data types except for checkbox fields are mapped to "". A field's value is considered null if either its item does not exist within the Lotus Notes document, or the existing value is the empty string "" and it is a non-string Domino data type. For checkbox fields, the null value is mapped to "false".

On data upload, for all data types except for checkboxes, if an empty string is uploaded then that field's internal value becomes "". For a checkbox field, the value returned from device will always be either "true" or "false". When Pylon Application Server sets a checkbox value within a Lotus Domino document, "false" is mapped to the empty string "", and "true" is mapped to the keyword / keyword-synonym for that value.

## Readers fields and Pylon Application Server Lotus Notes ID file

In order for the Pylon Application Server's Lotus Notes ID file (ID file) to be able to access documents with readers fields for other Lotus Notes users, that ID file must have access to every document that the Lotus Notes users want to access. Consequently, to prevent the ID file from ever losing read-access to documents, put a **Computed when Composed Authors** field on every form, with a formula such as "[Admin]". This will allow all members of [Admin] role to read the document, without preventing others from reading it as a Readers field would allow.

---

---

### Note

Lotus Domino Administrators: If there are documents that are hidden from everyone, including the database administrator, then you will have to access the Lotus Notes database from the server console to view and fix these documents. You may also have to temporarily disable the local security property ("Use consistent access control") in the database ACL.

## Authors Access

When a document is created on-device and synced to a Lotus Domino server, Pylon Application Server adds an Authors field called **\$PylonNotesUserName** to the document that contains the Lotus Notes username for the user that created the document. If this user is, or happens to become, an Author, this special field enables that user to access the documents they created using Pylon Application Server. On document updates from device-to-server, the value of **\$PylonNotesUserName** is not updated.

## CHAPTER 6. JavaScript support

---

- “Overview” (page 62)
- “Table of JavaScript events” (page 63)
- “Events that work both in browsers and the Notes client” (page 65)
- “Enabling JavaScript in the Lotus Notes client” (page 66)
- “About sample code” (page 67)

## Overview

Pylon Application Server supports JavaScript in the Lotus Notes databases. The JavaScript code lines are transferred to the mobile device and executed there as event triggers, the same as on the Lotus Notes desktop. Refer to Table 6-1 below, for a list of events supported in this release:

**Table 6-1**  
Supported  
JavaScript events

Event type	Event
General	JSHeader
Form events	OnLoad, OnSubmit
Field events	OnChange, OnFocus, OnBlur

For developers working with Lotus Notes 6.0 and later, Lotus Domino Designer provides three options for entering JavaScript for events:

- JavaScript for Client
- JavaScript for Web
- Common JavaScript

**Note**

---

---

JavaScript for Client is not supported by Pylon Application Server at this time.

## Table of JavaScript events

Following is a reference table from Lotus Corporation listing all Lotus Notes JavaScript events and their relevance to the client interface (Notes client or Web browser). "Browser only" means that the event handler only works in browsers that support it. "Browser and Notes" means that the event handler works both in browsers that support it and in the Lotus Notes client.

**Table 6-2**  
JavaScript event  
handlers in Lotus  
Domino Designer

Event(1)	Form, subform, page	Field	Action, button, hotspot
<b>JSHeader</b>	Browser, Notes, and Pylon	Field events can reference JSHeader functions.	
<b>onBlur</b>	N/A	Browser, Notes, and Pylon	Browser only
<b>onChange</b>	N/A	Browser, Notes, and Pylon	N/A
<b>onClick</b>	Browser only	Browser only	Browser and Notes
<b>onDblClick</b>	Browser only	Browser only	Browser only
<b>onFocus</b>	N/A	Browser, Notes, and Pylon	Browser only
<b>onHelp</b>	Browser and Notes	N/A	Browser only
<b>onKeyDown</b>	Browser only	Browser only	Browser only
<b>onKeyPress</b>	Browser only	Browser only	Browser only
<b>onKeyUp</b>	Browser only	Browser only	Browser only
<b>onLoad</b>	Browser, Notes, and Pylon	N/A	N/A
<b>onMouseDown</b>	Browser only	Browser only	Browser only
<b>onMouseMove</b>	Browser only	Browser only	Browser only
<b>onMouseOut</b>	Browser only	Browser only	Browser only
<b>onMouseOver</b>	Browser only	Browser only	Browser only
<b>onMouseUp</b>	Browser only	Browser only	Browser only

Event(1)	Form, subform, page	Field	Action, button, hotspot
onReset	Browser only, form only	N/A	N/A
onSelect	N/A	Browser only	N/A
<b>onSubmit</b>	Browser, Notes, and Pylon	N/A	N/A
onUnload	Browser and Notes	N/A	N/A

(1) Pylon Application Server supports Events listed in **bold** type.

## Events that work both in browsers and the Notes client

Restrict code that must work in both a browser and the Notes client to the following events:

- For document preprocessing and post-processing, use `onLoad`, `onUnload`, and `onSubmit`.
- For processing on entering and exiting fields, use `onFocus`, `onBlur`, and `onChange`.
- For an action, button, or action hotspot, use `onClick`.

### **JSHeader** event

`JSHeader` is a special event handler that loads code such as functions and global variable declarations that all the events in the object can access. This code goes under the `<HEAD>` tag in Lotus Domino-generated HTML.

### **onSubmit** event

The `onSubmit` event occurs in a browser and the Notes client when the `FileSave @command` executes. In a browser, you can return `false` from the `onSubmit` event to abort the save operation.

## Enabling JavaScript in the Lotus Notes client

To run JavaScript in a Lotus Notes client, the user must select "Enable JavaScript" under "Additional Options" after choosing File - Preferences - User Preferences. To expand or limit security, the user must select "Using JavaScript" under "What Others Do" after choosing File - Security - User Security.

A Form in Lotus Notes corresponds to a Detail page in M-Business Client on the device. A Field on a form in Lotus Notes corresponds to a field on a Detail page in M-Business Client on the device.

### Event descriptions

Table 6-3 below lists each event and its associated description:

**Table 6-3**  
JavaScript events

Event	Description
JSHeader	This is a placeholder for JavaScript functions. All functions placed here are available for Form and Field events to use. It provides a means to consolidate JavaScript codes.
OnLoad	JavaScript for this event is executed after the form (Detail Edit Page on device) and all fields data are loaded. It can be used to set initial values. This works the same as on the Notes desktop.
OnSubmit	JavaScript for this event is executed before the form (Detail Edit Page on device) is saved. It can be used to validate fields data. This works the same as on the Notes desktop.
OnChange	JavaScript for this event is executed when the field data is modified as the focus is leaving that field. One exception is for a Checkbox field, where the code is executed as the focus is coming to that field.
OnFocus	JavaScript for this event is executed after the field gets the input focus.
OnBlur	JavaScript for this event is executed after the field loses the input focus.

## About sample code

### Pylon Application Server demo databases

The following examples are from the demo databases that you can use with Pylon Application Server. Go to the following Web site to locate the zip file containing the Pylon Application Server software Download link:

<http://www.ianywhere.com/downloads/software/pylon.html>

Table 6-4 below lists the demo database files and their defined View and Form names.

**Table 6-4**  
Demo database View  
and Form names

Demo database	View name(s)	Form name
accountsA.nsf	Accounts Hardware Audit My Accounts	Account
companyA.nsf	Companies	Company
productsA.nsf	Inventory	Inventory
OrdersA.nsf	Orders	Orders

## Form validation sample: validating a form for incomplete fields

The following sample shows how to validate a form for incomplete fields at an `OnSubmit` event. It works on both Lotus Notes and the device-side M-Business Client.

```
var bCompleted = true;
var fm = document.forms[0];
var fname;
for(n = 0; n < fm.elements.length; n++)
{
    fname = fm.elements[n].name;
    if(fname != "" && fname != "Body")
    {
        if(fm.elements[n].type != "hidden" ||
           fname.substr(0, 7) == "lookup_")
        {
            if(fm.elements[n].value == "")
            {
                bCompleted = false;
                if(fname.substr(0, 7) == "lookup_")
                    fname = fname.substr(7);
                alert("You must complete the \"" +
                    fname + "\" before saving this form");
            }
        }
    }
}
return bCompleted;
```

**Note** Body is the default rich text field name which we allow to be empty; lookup\_ is the prefix for every lookup field name.

## Computed field validation sample: validating a form for incomplete fields

The following sample shows how to automate some of the field editing tasks. It calculates the cost based on the quantity ordered, unit pricing, and shipping method.

```
function calccost()
{
    var sm = 0, qt = 0, pc = 0;
    with(document.forms[0].Shipping)
    {
        for (n = 0; n < length; n++)
        {
            if (options[n].selected)
            {
                sm = options[n].value;
                if(sm == "Standard")
                    sm = 10;
                else if(sm == "3 day")
                    sm = 30;
                else if(sm == "Overnight")
                    sm = 90;
            }
        }
    }

    with(document.forms[0].Quantity)
    {
        for (n = 0; n < length; n++)
        {
            if (options[n].selected)
                qt = options[n].value;
        }
    }

    document.forms[0].Cost.value = qt * sm;
    pc = document.forms[0].Price.value;
    document.forms[0].Total.value = qt * pc + qt
        * sm;
}
```

This function is placed in JS Header section of the form, and the related field OnChange event calls it to update the total cost field.



# CHAPTER 7. **Implementing on-device database lookups**

---

---

- “About on-device database lookups” (page 72)
- “Creating a lookup field in a Lotus Notes database” (page 73)

## About on-device database lookups

Pylon custom database functionality is extended with the addition of the database lookup feature. The database lookup feature allows one custom database to access both the view and the data of one or more custom databases. Additionally, if there is more than one lookup field on the same Lotus Notes form that uses the same view and database, Pylon Application Server will automatically populate the other lookup fields when the first field on the form is selected.

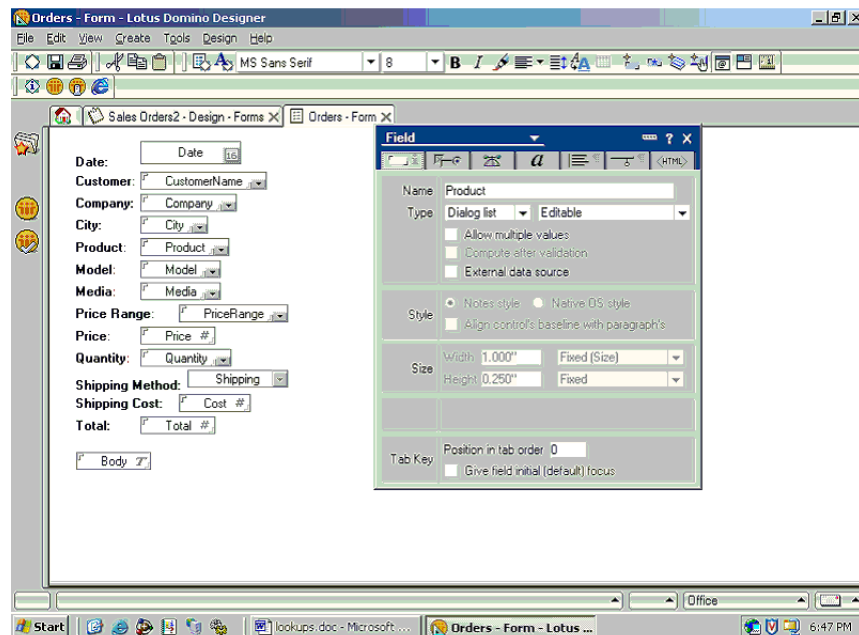
Lookups are an extension to the keyword functionality. The database lookup field allows one database to treat any view in a Pylon database as a user modifiable keyword list.

Another difference between normal keyword fields and lookup fields is the amount of data that can be brought back by a lookup. In a normal keyword field selection, only the keyword field is updated. However, in the case of a lookup field, the lookup operation can update more than just the lookup field; the mechanism supports the ability to update additional fields in the same form.

## Creating a lookup field in a Lotus Notes database

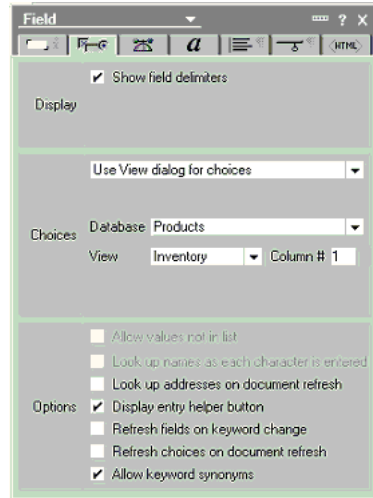
When creating a lookup mapping, there are always two databases involved: the current database and the lookup database. In the example below, “Orders” is the current database and “Products” is the lookup database.

**Figure 7-1**  
Creating a Lookup Field



You create a lookup field mapping by referencing within a local database form a column within a view of the lookup database. In Pylon Application Server, this Lotus Notes dialog list field is called a lookup field. Within Lotus Domino Designer, to create this lookup field you define a dialog list field with Choices configured for “Use View dialog for choices” as shown below. If you want additional columns in the view to automatically populate form fields when the first lookup field is selected, you would define each field after Product (here we do this for the Model, Media, and PriceRange fields) to be configured in the same fashion as the Product field except they map to Column # 2, Column # 3, and Column # 4 respectively.

**Figure 7-2**  
Creating a lookup  
field mapping



## Creating a lookup field mapping

To create a lookup field mapping:

1. Create a new field in a form and set the field Type to **Dialog list**.
2. In the Field designer, click the **Control** tab.
3. Set the Choices field in the Control panel to **Use View dialog for choices**.
4. Select a lookup **Database**, a specific lookup **View** within the lookup database, and a **Column #** within the view.

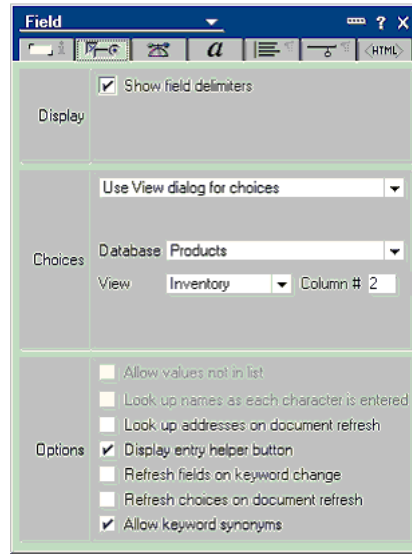
### Note

The lookup database does not have to be a separate Lotus Notes database; it may actually be a different view within the same Lotus Notes database. The lookup view can contain any fields that will be transferred into the current Pylon Application Server database, but the first column should contain the field that Pylon Application Server will use to sort the database.

## Retrieving additional fields

On-device, it is possible to retrieve more than just the lookup field in a single lookup operation. To automatically populate additional field values based on the selection of the first lookup field for the specific view and database, simply define additional fields below the first lookup field that are defined in a manner similar to the first field: except now set the Column # in the field's definition to refer to the column within the view to which you want to map that field. For example, in the above example the Control tab for the Model field is given below. Notice that it is identical to the Product field except now the Column # is 2.

**Figure 7-3**  
Retrieving  
additional lookup  
fields



## Synchronization issues

Make sure to synchronize both the current database and the lookup database and view specified by the lookup field in the current database. If you are using a lookup view within the same database, then you must synchronize the current database twice, creating one Pylon Application Server database using the current form and a regular view and the second Pylon Application Server database containing the lookup form and view.

### Note

The database object name must exactly match the view name (case-sensitive) for the database lookup to work.



# Sample implementations

“Sample databases” (page 79)



## APPENDIX A. **Sample databases**

---

---

- “Overview” (page 80)
- “Database configuration” (page 81)
- “Database descriptions” (page 82)
- “Setting up the databases” (page 86)
- “Using the databases” (page 87)

## Overview

This Appendix contains a summary of the four sample custom Lotus Notes databases that you can download from the Pylon Software Server Downloads Web site. The purpose of these databases is to demonstrate how to customize Lotus Notes databases and on-device functionality using Pylon Application Server 6.0. The sample Lotus Notes database files are as follows:

- Accounts: accountsA.nsf (page 82)
- Companies: companyA.nsf (page 83)
- Inventory: inventoryA.nsf (page 84)
- Orders: ordersA.nsf (page 85)

Go to the following Web site to locate the zip file containing the Pylon Application Server software Download link:

<http://www.ianywhere.com/downloads/software/pylon.html>

Use this chapter as a guide in setting up your actual databases. The database description accompanying each sample database listed in this chapter includes the View, Form, Field, and Type information.

## Database configuration

This section explains how to configure the sample databases included with Pylon Application Server for syncing. You have the choice of using the default group or creating your own Lotus Notes database group. If you choose to create a new group for the sample databases, remember to add the Pylon user to the new group to synchronize the databases. Use the following configuration information to set up the sample databases:

Object name	Database file name	View or formula	Form
Accounts	accountsA.nsf	accounts	Account
Companies	companyA.nsf	Companies	Company
Inventory	inventoryA.nsf	Inventory	Inventory
MyAccounts	accountsA.nsf	My Accounts	Account
Orders	ordersA.nsf	Orders	Orders

---

---

**Note**

The database, `accountsA.nsf`, is configured twice with two different views: `accounts` and `MyAccounts`. This creates two different on-device databases that refer to the same back-end Lotus Notes database: `accountsA.nsf`.

## Database descriptions

This section lists all of the sample databases along with a description of why and how they are used. It also lists database View, Form, Field, and Field Type information.

### Accounts: accountsA.nsf

This database represents individual contacts for your fictitious companies database. The fields "Company", "Address", "City", "State", and "Zip" use the Pylon Application Server "lookup" feature, which allows the client on the device to select values from another database and populate other field values on the form at the same time. Use the "Choose" link to "lookup" a company. When you choose a company, the other Company-specific fields are populated with information from that selection.

When submitting new entries, JavaScript stored in the form "onSubmit" action will validate that each field on the form has had information added to it. If one or more of the fields does not have a value, an alert message is indicates the field is missing a value, and the form is not saved.

- View: My Account
- Form: Account

Field	Type
Name	Editable Text
Title	Editable Text
Phone	Editable Text
Email	Editable Text
LastContact	Editable Date/Time
Company	Editable Dialog List (Lookup)
Address	Editable Dialog List (Lookup)
City	Editable Dialog List (Lookup)
State	Editable Dialog List (Lookup)
Zip	Editable Dialog List (Lookup)
Country	Editable Dialog List

## Companies: companyA.nsf

This database represents individual companies your sales rep may visit. If creating a new account, you will want to create a new company first.

- View: Companies
- Form: Company

Field	Type
Company	Editable Text
Address	Editable Text
City	Editable Text
State	Editable Text
Zip	Editable Text
Country	Editable Text

## Inventory: inventoryA.nsf

This database stores product information for sale by your fictitious representatives. This form takes advantage of "computed when composed" fields to make the fields read-only in edit mode.

- View: Inventory
- Form: Inventory

Field	Type
Product	Computed when Composed Text
Model	Computed when Composed Text
Price	Computed when Composed Text
Speed	Computed when Composed Text
Quality	Computed when Composed Text
Media	Computed when Composed Text
Standard	Computed when Composed Text
Options	Computed when Composed Text
MemoryStd	Computed when Composed Text
MemoryMax	Computed when Composed Text
PrintStd	Computed when Composed Text
PrintOpt	Computed when Composed Text

## Orders: ordersA.nsf

This is a fictitious ordering database your field representative would use to capture orders in the field. This form uses multiple, different "lookups" to pull information from other databases on the device.

This form also contains an example of on-device calculation. When entering a value in the "Price" field and choosing a "Quantity", the "Total" field is calculated by multiplying the two values. You can also choose a "Shipping Method", which automatically enters the cost into the "Shipping Cost" field and adds the cost to the "Total" field.

When submitting new entries, JavaScript stored in the form "onSubmit" action will validate that each field on the form has had information added to it. If one or more of the fields does not have a value, an alert message is thrown indicating the field missing a value and the form is not saved.

- View: Orders
- Form: Orders

Field	Type
Date	Editable Date/Time
CustomerName	Editable Dialog List (Lookup)
Company	Editable Dialog List (Lookup)
City	Editable Dialog List (Lookup)
Product	Editable Dialog List (Lookup)
Model	Editable Dialog List (Lookup)
Media	Editable Dialog List (Lookup)
PriceRange	Editable Dialog List (Lookup)
Price	Editable Number Field
Quantity	Editable Dialog List
Shipping	Editable Combo Box
Cost	Editable Number Field
Total	Editable Number Field

## Setting up the databases

You can use the default group or create your own Lotus Notes database group when you set up a sample database. If you choose to create a new group for a sample database, remember to add the Pylon Application Server user to the new group to synchronize the databases. Use the following configuration information to set up the demo databases:

### Accounts

Object Name: **Accounts**

Server: [Domino server where databases are stored]

Database: **accountsA.nsf**

View or Formula: **My Accounts**

Form: **Account**

### Company

Object Name: **Company**

Server: [Domino server where databases are stored]

Database: **companyA.nsf**

View or Formula: **Companies**

Form: **Company**

### Products

Object Name: **Products**

Server: [Domino server where databases are stored]

Database: **inventoryA.nsf**

View or Formula: **Inventory**

Form: **Inventory**

### Sales Orders2

Object Name: **Orders**

Server: [Domino server where databases are stored]

Database: **ordersA.nsf**

View or Formula: **Orders**

Form: **Orders**

## Using the databases

### Accounts

This database represents individual contacts for our fictitious companies database. The fields "Company", "Address", "City", "State", and "Zip" use the Pylon Application Server "lookup" feature, which allows the client on the device to select values from another database and populate other field values on the form at the same time. Use the "Choose" link to "lookup" a company. When you choose a company, the other Company-specific fields are populated with information from that selection.

When submitting new entries, JavaScript stored in the form "onSubmit" action will validate that each field on the form has had information added to it. If one or more of the fields does not have a value, an alert message is indicates the field is missing a value, and the form is not saved.

### Companies

This database represents individual companies our sales rep may visit. If creating a new account, you will want to create a new company first.

### Inventory

This database stores product information for sale by our fictitious representatives. This form takes advantage of "computed when composed" fields to make the fields read-only in edit mode.

### Orders

This is our fictitious ordering database a field representative would use to capture orders in the field. This form uses multiple, different "lookups" to pull information from other databases on the device.

This form also contains an example of on-device calculation. When entering a value in the "Price" field and choosing a "Quantity", the "Total" field is calculated by multiplying the two values. You can also choose a "Shipping Method", which automatically enters the cost into the "Shipping Cost" field and adds the cost to the "Total" field.

When submitting new entries, JavaScript stored in the form "onSubmit" action will validate that each field on the form has had information added to it. If one or more of the fields does not have a value, an alert message is thrown indicating the field missing a value and the form is not saved.



# Index

## A

- ACL 38
- Adaptive Server Anywhere database 32
- Admin Server 32
- Apollo
  - supported field types 41
- architecture
  - Pylon Application Server 22
- Authors field 60

## C

- Checkbox field
  - columns 42
- checkbox field
  - functionality 42
- child form configuration 47
- client
  - supported platforms 23
  - system requirements 23
- Client components 23
- code sample
  - computed field validation 69
  - form validation 67
- columns
  - viewing from synced form 40
- communication between server and device 35
- components
  - M-Business Client 35
- Computed when Composed Authors field 59
- conduit
  - database 33
- conventions 16
- creating a custom icon 45
- creating a Lookup Field 73
- custom icon creation 45

## D

- data flow
  - overview 24
- data store 32
  - defined 32
- database conduit 33
- database design guidelines 40
- database lookup
  - overview 72
- database object name and view name match 74
- database setup 80
- date/time fields
  - JavaScript support 42
- demo database file locations 67
- design objects
  - unsupported on-device 48
- device/server communication 35
- downloads site 67

## E

- ECC encryption 38
- editable fields
  - computing 42
- empty string
  - mapping 58
- events
  - descriptions 66
  - for both browser and Lotus Notes client 65
  - JavaScript 62
  - JSHeader 65
  - onBlur 65
  - onChange 65
  - onClick 65
  - onFocus 65
  - onLoad 65
  - onSubmit 65

- onUnload 65
- examples *see* sample code
- extend
  - existing Lotus Domino -based applications 38

## F

- field
  - Authors 60
  - Computed when Composed Authors 59
- field labels
  - maximum character width on-device 57
- field name
  - how Pylon Application Server uses 44
  - linking to a column 44
- field type labels 50
- field type mappings 51
- fields
  - effective maximum number 42
  - sync limit for text 42
- form
  - maximum fields 42
- form field keyword synonyms 58
- form field types
  - supported for defining a list field 50
- form mobilization 47
- formatting conventions 16
- formulas 40
  - support for 44

## G

- general design guidelines 40

## H

- hard-coded keyword values 40
- hidden field attribute 50
- httpd server (Unix) 32

## I

- icon creation 45
- ID file 59
- installation
  - server requirements 22
  - system requirements 23

## J

- JavaScript

- Lotus Notes database support 62
  - supported events 62
- JavaScript errors 42
- JavaScript events 63

## K

- keyword functionality and lookups 72
- Keyword synonyms
  - checkbox 58
  - combobox 58
  - dialog list 58
  - radio button 58
- keyword synonym supported field types 58
- Keyword synonyms 58
  - listbox 58
- keyword synonyms
  - form field 58

## L

- lookup field
  - retrieving more than one 74
- lookup field creation 73
- lookups and keyword functionality 72
- Lotus Notes applications
  - reasons to mobilize 20
- Lotus Notes ID file 59
- Lotus Notes JavaScript events 63

## M

- M-Business Admin Server 32
- M-Business Anywhere
  - overview 30
- M-Business Client
  - components 35
  - desktop component 35
  - overview 34
- M-Business Client with M-Business Server
  - overview 34
- M-Business Connect
  - about 35
  - Palm OS, Pocket PC OS, Windows XP device 35
- M-Business Database Conduit
  - overview 33
- M-Business Server
  - Admin Server 32
  - database 32
  - environment 31
  - Sync Server 32

multiple checkbox handling 42

## N

new features

    Pylon Application Server 38

new features in this release 25

null field value mapping 58

## O

object name/view name match for synchronization 75

on-device database name

    naming criteria 43

## P

Pylon Application Server

    benefits of use 20

    data management 24

    device usability 21

    new features 38

    security features 21

    server management features 21

    supported design objects 46

    synchronization 21

    unsupported design objects 48

## R

rich text

    field validation 43

RIM Wireless Handheld with M-Business Connect 35

## S

sample code 77–??

sample database

    configuration information 81

    usage 87

sample databases

    descriptions 82

    setup 86

sample Lotus Notes database files 80

security

    ECC encryption 38

server

    Admin Server 32

    Sync Server 32

    UI Server 32

server/device communication 35

settings

    M-Business Connect 35

smart client applications 38

support design objects 46

supported design objects

    ACL 46

    agents 46

    form field labels 46

    form fields 46

    forms 46

    formulas 46

    JavaScript 46

    views 46

Sync Server 32

syncing multiple views and forms 40

## U

UI Server 32

ui server 32

## V

validation

    rich text field 43

view customization 44

view name and database object name match 74

