

Novell exteNd Application Server

5.1

www.novell.com

DATABASE CONFIGURATION GUIDE



Novell®

Legal Notices

Copyright © 2003 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher. This manual, and any portion thereof, may not be copied without the express written permission of Novell, Inc.

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Copyright ©1997, 1998, 1999, 2000, 2001, 2002, 2003 SilverStream Software, LLC. All rights reserved.

SilverStream software products are copyrighted and all rights are reserved by SilverStream Software, LLC

Title to the Software and its documentation, and patents, copyrights and all other property rights applicable thereto, shall at all times remain solely and exclusively with SilverStream and its licensors, and you shall not take any action inconsistent with such title. The Software is protected by copyright laws and international treaty provisions. You shall not remove any copyright notices or other proprietary notices from the Software or its documentation, and you must reproduce such notices on all copies or extracts of the Software or its documentation. You do not acquire any rights of ownership in the Software.

Novell, Inc.
1800 South Novell Place
Provo, UT 85606

www.novell.com

exteNd Application Server *Database Configuration Guide*
December 2003

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

ConsoleOne is a registered trademark of Novell, Inc.
eDirectory is a trademark of Novell, Inc.
GroupWise is a registered trademark of Novell, Inc.
exteNd is a trademark of Novell, Inc.
exteNd Composer is a trademark of Novell, Inc.
exteNd Director is a trademark of Novell, Inc.
iChain is a registered trademark of Novell, Inc.
jBroker is a trademark of Novell, Inc.
NetWare is a registered trademark of Novell, Inc.
Novell is a registered trademark of Novell, Inc.
Novell eGuide is a trademark of Novell, Inc.

SilverStream Trademarks

SilverStream is a registered trademark of SilverStream Software, LLC.

Third-Party Trademarks

Acrobat, Adaptive Server, Adobe, AIX, Autonomy, BEA, Cloudscape, DRE, Dreamweaver, EJB, HP-UX, IBM, Informix, iPlanet, JASS, Java, JavaBeans, JavaMail, JavaServer Pages, JDBC, JNDI, JSP, J2EE, Linux, Macromedia, Microsoft, MySQL, Navigator, Netscape, Netscape Certificate Server, Netscape Directory Server, Oracle, PowerPoint, RSA, RSS, SPARC, SQL, SQL Server, Sun, Sybase, Symantec, UNIX, VeriSign, Windows, Windows NT

All third-party trademarks are the property of their respective owners.

Third-Party Software Legal Notices

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JDOM.JAR

Copyright (C) 2000-2002 Brett McLaughlin & Jason Hunter. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution. 3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@jdom.org. 4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (<http://www.jdom.org/>)." Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jdom.org/images/logos>.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sun

Sun Microsystems, Inc. Sun, Sun Microsystems, the Sun Logo Sun, the Sun logo, Sun Microsystems, JavaBeans, Enterprise JavaBeans, JavaServer Pages, Java Naming and Directory Interface, JDK, JDBC, Java, HotJava, HotJava Views, Visual Java, Solaris, NEO, Joe, Netra, NFS, ONC, ONC+, OpenWindows, PC-NFS, SNM, SunNet Manager, Solaris sunburst design, Solstice, SunCore, SolarNet, SunWeb, Sun Workstation, The Network Is The Computer, ToolTalk, Ultra, Ultracomputing, Ultraserver, Where The Network Is Going, SunWorkShop, XView, Java WorkShop, the Java Coffee Cup logo, Visual Java, and NetBeans are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Indiana University" and "Indiana University Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <http://www.extreme.indiana.edu/>. 5. Products derived from this software may not use "Indiana University" name nor may "Indiana University" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Phaos

This Software is derived in part from the SSLava™ Toolkit, which is Copyright ©1996-1998 by Phaos Technology Corporation. All Rights Reserved. Customer is prohibited from accessing the functionality of the Phaos software.

W3C

W3C® SOFTWARE NOTICE AND LICENSE

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications: 1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. 2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code. 3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Contents

About This Guide	9
1 Configuring Cloudscape Databases	11
Setting up SilverMaster databases	11
Step 1: Install the Cloudscape software	12
Step 2: Create a Cloudscape SilverMaster database	12
Setting up deployment databases	13
Step 1: Create a Cloudscape deployment database	13
Step 2: Add the Cloudscape database or connection pool to the application server	13
Locating Cloudscape databases	14
Cloudscape users, schema, and the application server	15
Managing access to Cloudscape databases	15
2 Configuring DB2 Databases	17
Connecting to DB2 databases	17
Step 1: gather information about your DB2 environment	18
Step 2: prepare DB2 for connection from Universal Database Client	18
Step 3: install the database client	19
Step 4: configure the DB2 client connection	20
Step 5: modify the DB2CLI.INI file	21
Step 6: set environment variables	23
Step 7: use the database with the application server	24
Troubleshooting	24
3 Configuring Informix Databases	25
Connecting to Informix databases	25
Step 1: install the JDBC driver	26
Step 2: create the Informix database and dbspaces	26
Step 3: create the Informix user account	26
Step 4: add the driver path to AGCLASSPATH	26
Step 5: use the database with the application server	26
Troubleshooting	27
4 Configuring Microsoft SQL Server Databases	29
Basic setup for Microsoft SQL Server databases	29
Create the database and configure SQL Server	29
Create the database login	30
Setting up JDBC access	30
Install the JDBC driver	30
Add the driver path to AGCLASSPATH	30
Setting up ODBC access	31
Install the Microsoft SQL Server client	31
Create the ODBC System DSN entry	31
Using the database with the application server	31

5	Configuring MySQL Databases	33
	About MySQL support	33
	Preparing to access MySQL	34
	When MySQL is installed for you	34
	When you install MySQL yourself	34
	Creating MySQL databases	36
	Setting up a SilverMaster database	36
	When SilverMaster is set up for you	37
	When you set up SilverMaster yourself	37
	Setting up a deployment database	37
6	Configuring Oracle Databases	39
	Basic setup for Oracle databases	39
	Create an Oracle tablespace for use with SilverMaster	39
	Create an Oracle user for the application server	40
	Setting up an Oracle driver	40
	Using the Oracle Thin JDBC Driver	40
	Using the Oracle OCI Driver	41
	Using the Novell exteNd Oracle driver	41
	Using the database with the application server	42
	Creating a deployment database on Oracle	43
	Using synonyms	45
	Specifying the driver	45
	Types of synonyms	46
	Enabling and disabling synonym support	46
	Troubleshooting	46
7	Configuring Sybase Adaptive Server Anywhere Databases	49
	Preparing to use Sybase ASA	49
	Installing Sybase ASA	49
	Preparing to use jConnect	50
	Setting up SilverMaster and deployment databases	50
	Step 1: Create a database	50
	Step 2: Create user accounts	51
	Step 3: Run Sybase ASA as a service	51
	Step 4: Make the database available to the application server	51
8	Configuring Sybase Adaptive Server Enterprise Databases	53
	Setting up Sybase Adaptive Server Enterprise databases	53
	Step 1: install Sybase jConnect	54
	Step 2: configure the Sybase server	54
	Step 3: create the SilverMaster database in Sybase	55
	Step 4: create the Sybase login for the application server	56
	Step 5: use the database with the application server	56
	Troubleshooting	57
	Adaptive Server Enterprise troubleshooting	57
	jConnect troubleshooting	59

About This Book

Purpose

This book describes how to configure supported databases for use with the Novell® exteNd™ Application Server.

Audience

This book is for anyone responsible for installing and configuring the Novell exteNd Application Server (for both development and production environments).

Prerequisites

Before you begin database configuration, you'll need an understanding of the overall installation process for the application server and other exteNd products.



See [Installing Novell exteNd](#).

Organization

Read the chapter for your DBMS to learn about the database configuration you must perform:

Chapter	Description
Chapter 1, “Configuring Cloudscape Databases”	How to set up IBM Cloudscape databases for use within the application server environment
Chapter 2, “Configuring DB2 Databases”	How to set up IBM DB2 databases for use within the application server environment
Chapter 3, “Configuring Informix Databases”	How to set up IBM Informix databases for use within the application server environment
Chapter 4, “Configuring Microsoft SQL Server Databases”	How to set up Microsoft SQL Server databases for use within the application server environment

Chapter	Description
Chapter 5, "Configuring MySQL Databases"	How to set up MySQL databases for use within the application server environment
Chapter 6, "Configuring Oracle Databases"	How to set up Oracle databases for use within the application server environment
Chapter 7, "Configuring Sybase Adaptive Server Anywhere Databases"	How to set up Sybase Adaptive Server Anywhere databases for use within the application server environment
Chapter 8, "Configuring Sybase Adaptive Server Enterprise Databases"	How to set up Sybase Adaptive Server Enterprise databases for use within the application server environment

Additional documentation



For the complete set of Novell exteNd documentation, see the [Novell documentation Web site](http://www.novell.com/documentation/exteNd.html) (www.novell.com/documentation/exteNd.html).

1

Configuring Cloudscape Databases

This chapter describes how to configure your IBM Cloudscape database for use as a SilverMaster or deployment database with the Novell exteNd Application Server. It includes these sections:

- ◆ [Setting up SilverMaster databases](#)
- ◆ [Setting up deployment databases](#)
- ◆ [Locating Cloudscape databases](#)
- ◆ [Cloudscape users, schema, and the application server](#)
- ◆ [Managing access to Cloudscape databases](#)



For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Setting up SilverMaster databases

To use a Cloudscape database as the SilverMaster database, you need to:

- 1 Install the Cloudscape software.
- 2 Create a Cloudscape SilverMaster database.

The sections that follow explain these steps.

Step 1: Install the Cloudscape software

To use Cloudscape databases with the Novell exteNd Application Server, you must:

- 1 **Install** the Cloudscape software yourself (it's available from the IBM Cloudscape Web site at www-3.ibm.com/software/data/cloudscape/).
- 2 **Specify the location** of these three Cloudscape JAR files in the AGCLASSPATH environment variable (in the .agprofile file on Linux):

```
cloudscape.jar  
cloudutil.jar  
RmiJdbc.jar
```


On Linux, the application server install will ask for the location of these JARs so it can update the .agprofile file for you.

 For information on editing AGCLASSPATH, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

As with any SilverMaster or deployment database, you must be able to connect to the destination Cloudscape database from the machine that hosts the Novell exteNd Application Server.

Step 2: Create a Cloudscape SilverMaster database

You can create the Cloudscape database for SilverMaster in either of these ways:

- ◆ **Using the Cloudscape tools** (see the Cloudscape documentation at www-3.ibm.com/software/data/cloudscape/)
Once your Cloudscape database is created, you can install the application server and point to that database as the SilverMaster (as described in *Installing Novell exteNd*).
If the Cloudscape database that you use already contains users (not the APP user), then the database must also contain a schema of the same name (see **“Cloudscape users, schema, and the application server” on page 15**).
- ◆ **When you install the application server**
The application server install will ask for the JDBC URL of the Cloudscape database to use as SilverMaster. You can create the database at this time, by appending the string **;`create=true`** to the URL. (This is an easy way to create an unsecured Cloudscape database.)
 For more information on Cloudscape JDBC URLs, see **“Locating Cloudscape databases” on page 14**.

Setting up deployment databases

To set up a Cloudscape database as a deployment database, you need to:

- 1 Create the Cloudscape database.
- 2 Add the Cloudscape database or connection pool to the application server.


The following sections describe these steps.

Step 1: Create a Cloudscape deployment database

You can create the Cloudscape database in either of these ways:

- ◆ **Using the Cloudscape tools** (see the Cloudscape documentation at www-3.ibm.com/software/data/cloudscape/)
- ◆ **When you add the database** to the application server

In this case, you must specify the JDBC URL for the database. You can create the database at the same time that you add it, by appending the string `;create=true` to the URL.

 For more information on Cloudscape JDBC URLs, see “[Locating Cloudscape databases](#)” on page 14.

Step 2: Add the Cloudscape database or connection pool to the application server

You can add a Cloudscape database using either of the following:

- ◆ The Add Database dialog from the SMC
- ◆ The [AddDatabase](#) command of the SilverCmd command-line tool

The examples below show how to create a new database (with `;create=true` on the JDBC URL) while adding it to the application server. You do not need `;create=true` when you’re adding an existing database. If the database already exists, this string is ignored.

Using the Add Database dialog Provide these values:

- ◆ Database name
- ◆ User name: **APP**
- ◆ Password: **password**
By default, Cloudscape databases are created with no security. Use the APP/password combination whenever connecting to an unsecured database.
- ◆ Database platform: **IBM Cloudscape**
- ◆ JDBC URL: **`jdbc:cloudscape:databasepath/databasename;create=true`**

For example:

```
jdbc:cloudscape:c:/mydb/Testdb;create=true
```

Using SilverCmd AddDatabase You can invoke SilverCmd AddDatabase like this:

```
SilverCmd AddDatabase server[:port] -f file [options]
```

The following XML shows the contents of a file that you might specify using the `-f` parameter. This example creates and adds a database named `EStoreDB` and connects as the user `APP`.

```
<DatabaseOptions isObject="true">
  <MainDatabase isObject="true">
    <ConnectionOptions isObject="true">
      <DatabaseName type="String">EStoreDB</DatabaseName>
      <Username type="String">APP</Username>
      <Password type="String">password</Password>
      <LDSKey type="String">CloudscapeEmbeddedDriver</LDSKey>
      <OtherDriver isObject="true">
        <JDBC_URLtype="String">jdbc:cloudscape:c:/mydb/Testdb;create=tr
ue
        </JDBC_URL>
      </OtherDriver>
    </ConnectionOptions>
  </MainDatabase>
</DatabaseOptions>
```

Adding connection pools Another way to access a Cloudscape database is to add a connection pool for it to the application server. For details on adding databases and connection pools, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Locating Cloudscape databases

Cloudscape JDBC URLs take this form:

```
jdbc:cloudscape:dbname
```

The `dbname` parameter can be a fully qualified pathname, a relative pathname, or a database name. The application server uses the `cloudscape.system.home` system property to locate databases (unless they are fully qualified). By default, the application server sets `cloudscape.system.home` to the following directory:

```
AppServerInstallDir\db
```

If the application server is unable to locate the database using the system property, it tries the current directory.

Cloudscape users, schema, and the application server

If the application server connects to the Cloudscape database as a specific user (not APP), it requires the database to contain a schema of the same name. For example, if you want to connect to a database as the user DBADMIN, then the database must contain a schema also called DBADMIN.



For more information on creating users and schemas, see the Cloudscape documentation (at www-3.ibm.com/software/data/cloudscape/).

Managing access to Cloudscape databases

Some versions of the Cloudscape DBMS are designed to support a single database connection. If the Cloudscape JARs are found on your AGCLASSPATH, the Cloudscape database will automatically start in-process with the application server using the single allowed connection.

Keeping the Cloudscape database from starting automatically If you do not want it to start, you can remove the Cloudscape JARs from AGCLASSPATH.

Running Cloudscape in a cluster server environment Cloudscape databases are not sharable between more than one process at a time and are not appropriate for a cluster server environment.

Connecting to Cloudscape from other tools When the application server has a database open, you must use a different JDBC driver and URL to connect to that database from a different source, such as a JDBC program or a Cloudscape tool. To use Cloudscape's ij, you would normally do the following:

```
java COM.cloudscape.tools.ij
ij> connect 'jdbc:cloudscape:d:/MyDbs/MyDatabase';
```

But to connect while the application server is running, you would have to run it like this:

```
java -Dij.driver=COM.cloudscape.core.RmiJdbcDriver
COM.cloudscape.tools.ij
ij> connect 'jdbc:cloudscape:rmi:d:/MyDbs/MyDatabase';
```

In CloudView, before connecting to the database click on the Connection tab, and change the prefix field to jdbc:cloudscape:rmi://localhost:1099/. This automatically changes the driver to COM.cloudscape.core.RmiJdbcDriver.

Allowing access to Cloudscape from external applications When the Novell exteNd Application Server has a connection to a Cloudscape database, other applications are only able to access the database through Cloudscape's remote RMI/JDBC bridge via the application server's **RMI/JDBC listener**. The application server provides the following property (in the **httpd.props** file) to control the startup of the RMI/JDBC listener:

```
http-
server.com.sssw.db.cloudscape.cloudscapeDatabaseRMIJDBCListener=
[true|false]
```

The following table describes the property settings:

Property Value	Description
true	Setting this value to true causes the application server to start the Cloudscape RMI/JDBC listener at server startup. Starting this listener makes it possible for other applications (such as the Novell exteNd Director EJB Wizard) to connect to the Cloudscape databases that the application server is also connecting to (or holding connections to) using Cloudscape's remote RMI/JDBC bridge driver.
false (default)	Setting this value to false prevents the application server from starting the Cloudscape RMI/JDBC listener at server startup. External applications (such as exteNd Director™) will not be able to connect to any Cloudscape database to which the application server is connected. If you do not need external access to Cloudscape, leave this flag set to false. When the Cloudscape RMI/JDBC listener is enabled, the application server is slower to shut down because it must do additional processing. When set to false, this flag prevents the slower than normal server shutdown.

Running multiple application servers on a single host If you want to run multiple instances of the Novell exteNd Application Server on one computer and run Cloudscape, you need to specify a different Cloudscape **RMI/JDBC port number** for each server in its **httpd.props** file (the default is port 1099):

```
http-server.com.sssw.srv.cloudscape.RmiJdbcPort=number
```

Setting this line to a different port for each server avoids a port number conflict. For more information on setting other ports and properties for running multiple servers on a single host, see the chapter on [running the server](#) in the *Administrator's Guide*.

2

Configuring DB2 Databases

This chapter describes how to configure your DB2 database for use with the Novell exteNd Application Server as a SilverMaster or deployment database. It contains the following sections:

- ◆ [Connecting to DB2 databases](#)
- ◆ [Troubleshooting](#)



For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Connecting to DB2 databases

The Novell exteNd Application Server connects to DB2 using a Type 2 JDBC driver written by IBM and included with their Universal Database client. This driver interfaces with the UDB client to establish database connectivity to the back-end DB2 server. The application server officially supports TCP/IP connectivity, but customers have successfully connected using APPC, IBM's native protocol.

Follow the steps outlined below to make a DB2 database available to the application server:

Step 1: gather information about your DB2 environment

Step 2: prepare DB2 for connection from Universal Database Client

Step 3: install the database client

Step 4: configure the DB2 client connection

Step 5: modify the DB2CLI.INI file

Step 6: set environment variables

Step 7: use the database with the application server

Supported database usage DB2 can be used as the SilverMaster database and to store the application server's system tables required for deployment databases, with this exception: DB2 cannot be used as a SilverMaster (or store the system tables) when DB2 is running on AS/400 or System 390.

SilverMaster and deployment databases Note that a deployment database cannot exist in the same physical space as the SilverMaster database, because several of the required tables have identical names.

Step 1: gather information about your DB2 environment

Gather the following information for your DB2 environment.

- ◆ Operating system (for example: OS/390, MVS)
- ◆ Operating system version
- ◆ DB2 version (if different from OS)
- ◆ Is your database set up for TCP/IP support?
- ◆ Server host name
- ◆ Database name
- ◆ TCP/IP port for the database
- ◆ Service name
- ◆ User ID/password with full read access
- ◆ Owner, schema, or library list that contains the desired tables and views

NOTE: If your DB2 version is earlier than V4R2, your database may require extensive configuration to support Universal Database Client.

Step 2: prepare DB2 for connection from Universal Database Client

IBM has extensive documentation available on configuring the database and the client software to support connectivity from the Universal Database Client. You can find documentation in the Information Center that is installed with the Universal Database Client, or on IBM's Web site.

If you have an extremely large database schema, check for additional database configurations that can help reduce the time it takes to scan the schema in the DB2CLI.INI file. In addition, see the following IBM documents:

- ◆ *Preparing MVS/ESA or OS/390 for DB2 Connect*
- ◆ *Preparing DB2 for AS/400 for DB2 Connect*

- ◆ *Connecting DB2 for MVS/ESA in a DRDA Network*
- ◆ *Connecting DB2 for OS/390 in a DRDA Network*
- ◆ *Connecting DB2 for AS/400 in a DRDA Network Using SNA*
- ◆ *Connecting DB2 for AS/400 in a DRDA Network Using TCP/IP*

Step 3: install the database client

The database client must be installed on the machine where the application server will reside.

Using Universal Database Client Version 5.2 (Enterprise Edition)

Windows NT Follow the instructions in the setup program. Do not confuse the Universal Database Client with the Client Application Enablers (CAE). Installing the CAE will not give you all the components you need. You must install the entire Enterprise Connect product.

Solaris Before you can begin the installation of the DB2 database client on Solaris, you must first alter several kernel configuration parameters to accommodate the install. In the /etc/system directory:

```
set msgsys:msginfo_msgmax=65535
set msgsys:msginfo_msgmnb=65536
set msgsys:msginfo_msgseg=8192
set msgsys:msginfo_msgssz=16
```

Install the database client from the CD. You must install and create a DB2 instance during this installation so you can reference the DB2 profile script for the application server to successfully use the DB2 JDBC driver.

Using DB2 Connect Version 6.1 (Enterprise Edition)

Windows NT Follow these instructions for running the setup program:

- 1 Choose the **DB2 Enterprise - Extended Edition** check box.
You can leave the default choice for the information about the server instance.
- 2 Select **Custom**.
- 3 Select the following items from the list:
 - ◆ Java Enablement
 - ◆ DB2 Connect Server Support
 - ◆ Client Configuration Assistant

UNIX Follow the same instructions as shown above for installing Version 5.2 in Solaris, except that you install DB2 Connect - Enterprise Edition Version 6.1.

HP-UX note The following are recommended values for HP-UX kernel configuration parameters when installing DB2 6.1:

- ◆ msgseg: 8192
- ◆ msgmnb: 65535
- ◆ msgmax: 65535
- ◆ msgssz: 16

Using DB2 Connect Version 7 (Enterprise Edition)

Windows NT Follow the instructions in the setup program.

HP-UX and Solaris See the *IBM DB2 Connect Enterprise Edition for UNIX Quick Beginnings* guide on the IBM Web site for information on installing DB2 Connect and for setting HP-UX and Solaris kernel parameters.

Step 4: configure the DB2 client connection


Windows NT The application server connects using DB2 client **aliases**. These aliases are configured using IBM's Client Configuration Assistant, which can be found in the DB2 for Windows NT program folder. In order to run the Client Configuration Assistant, your NT login must have administrator authority and the ID must be no longer than eight characters. You need to be the DB2 administrator or have administrator privileges to complete these steps.

➤ To create a new DB2 client alias:

NOTE: You may have to contact your DB2 Administration staff to obtain some of the information required in these steps.

- 1 Open the DB2 Client Configuration Assistant from the DB2 for Windows NT program folder.
- 2 Click the **Add** button at the top right of the Client Configuration Assistant screen. Select **Next** to continue.
- 3 Choose to manually configure a connection to a DB2 database. Select **Next** to continue.
- 4 Choose the appropriate protocol (default is TCP/IP). Select **Next** to continue.
- 5 Choose the appropriate platform or LAN-based.
- 6 Configure the DB2 server host name, TCP port number, and service name (optional) that your DB2 server is operating on. Select **Next** to continue.
- 7 Enter the target database name. Select **Next** to continue.
- 8 Enter the alias name (you will use this name later to add the database to the application server). Select **Next** to continue.

- 9 (Optional) Register these settings as an ODBC data source. This step is not required but is recommended.
- 10 Finally, test the connection from inside the Client Configuration Assistant.

 For information on installation, see *Most Common DB2 Connect Problems* on the IBM Web site.

Solaris In order to establish a connection to your DB2 database, enter the DB2 command-line processor and enter the following two commands.

```
catalog tcpip node tcpipnodealias remote servername server port#
```


where *tcpipnodealias* is the local alias for your DB2 server (usually the same as the *servername*), *servername* is the name of the DB2 server, and *port#* is the port on the database server that is set to receive TCP/IP calls.

```
catalog database databasename as aliasname at node tcpipnodealias
```

where *databasename* is the actual database name, *aliasname* is the local name of the database (which may be the same as the actual name), and *tcpipnodealias* is the name given in the first catalog statement.

To test the DB2 connection:

```
connect to aliasname user userid using password
```

 For information on installation, see *Most Common DB2 Connect Problems* on the IBM Web site.

Step 5: modify the DB2CLI.INI file

IBM provides a client side .ini file for the purpose of tailoring the client connection to the database. Without modifications, the application server will attempt to add **all** DB2 tables and views into its environment. Most applications need only a subset of the entire list. Addition of a DB2 database without modifications to the .ini file can take an extraordinarily long time.

There are several things that can be done to reduce the time it takes for the application server to read the schema and two main approaches to follow:

- ◆ [Using exteNd tools](#)
- ◆ [Using IBM tools](#)

Using exteNd tools

One option is to specify the subset of tables to make available to the application server when adding the database to the server. For more information, see the chapter on [SilverCmd](#) in the *Facilities Guide* or the section on [adding a database](#) (in the chapter on data source configuration) in the *Administrator's Guide*.

Using IBM tools

GUI approach Alternatively, you can follow these steps to make changes to .ini file parameters from the Client Configuration Assistant:

- 1 View the properties for an alias and then choose to modify the ODBC settings.
- 2 Connect to the database, then choose **advanced**.

The specific settings found to help with application server installs are listed below. The full list can be reviewed in the Call Level Interface document from IBM.

Text approach Optionally, the DB2 client .ini file (which can be found in the sqllib directory) can be read and altered using a text editor. Your initial alias entry (with the name you entered via the Client Configuration Assistant) should look like this:

```
[db2alias]
DBALIAS=db2alias
```

To limit the number of tables, you can specify a particular **schema**. *Schema* here refers to the owner/user of the objects or library in the case of AS/400:

```
SCHEMALIST="userid", "userid2"
```

For extremely large datasets, use of SCHEMALIST may not be enough to reduce scanning time. To greatly reduce the total number of tables that the application server interrogates, you need to complete these steps:

- 1 Create views of all the system tables using an alternate owner (must be performed by DBA). Do not create these views under the same owner ID as the tables you wish to access for your application; create them under a unique owner.
- 2 Reference that owner of those views in the .ini file.

```
SYSSCHEMA="owner"
```

For example, you may create your views to see the objects of three different owners. This sample view syntax was used for DB2 on NT.

```
CREATE VIEW MYSCHEMA.SYSTABLES AS SELECT * FROM SYSIBM.SYSTABLES
WHERE CREATOR IN ('USER1', 'USER2', 'USER3')
```

After views were created for all of the required system tables under the same schema, the entry in the ini file would look like this:

```
SYSSCHEMA="MYSCHEMA"
```

The application server will then scan only those objects. If used in conjunction with SCHEMALIST, the server will scan the objects of the three owners but only add those belonging to the owners referenced in SCHEMALIST.

IMPORTANT: The preceding options are highly recommended to reduce installation and server start times.

Other configuration options

For MVS environments, you may have to set a variable for long data type compatibility. For a complete description of this setting, see the LONGDATACOMPAT section of the Call Level Interface document.

```
LONGDATACOMPAT=1
```

If you can't save records in DB2 after adding the database, you may have to set a transaction variable. The example listed below is for an AS/400 system and eliminates the need for table journaling. For a complete list of options, see the TXNISOLATION section of the Call Level Interface document.

```
TXNISOLATION=32
```

Step 6: set environment variables

Windows NT Connecting the application server to DB2 requires that you add the DB2 libraries to your AGCLASSPATH environment variable. To do this, open Control Panel / System and select the Environment tab. Create a System environment variable called AGCLASSPATH and set its value to C:\sqllib\java\db2java.zip (assuming that you installed your DB2 client onto the C: drive in the default location); if you did not install the DB2 client onto the C: drive, set the AGCLASSPATH variable accordingly.

By default, this sets up the JDBC 1.0 driver. In DB2 7.x, you can use the JDBC 2.0 driver by running this script: sqllib\java12\usejdbc2 (for details, [click here](http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?&fn=1008960): www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?&fn=1008960).

UNIX DB2 requires that you add the DB2 JDBC driver's native libraries and Java classes to your Java classpath and system library path. To do this, you may reference the DB2 instance (created previously) directly in the application server's environment file .agprofile (which is stored in the application server's installation directory). Edit .agprofile by adding the following line:

```
. /export/home/db2inst1/sqllib/db2profile
```

By default, this sets up the JDBC 1.0 driver. In DB2 7.x, you can use the JDBC 2.0 driver by adding the following lines to your .agprofile file (for details, [click here](http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?&fn=1008960): www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?&fn=1008960):

```
. /home/db2inst1/sqllib/db2profile  
. /home/db2inst1/sqllib/java12/usejdbc2
```




For more information on editing AGCLASSPATH and .agprofile, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Step 7: use the database with the application server

Now you're ready to access the DB2 database from the application server. This enables you to do any of the following, as appropriate:

- ◆ Install the application server and point to that database as the **SilverMaster** (as described in *Installing Novell exteNd*)
- ◆ Add that database to the application server for use as a **deployment database**
- ◆ Add a **connection pool** to the application server for accessing that database

 For more information, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Troubleshooting

SQL0551N “user does not have the privilege to perform operation GRANT” If you receive this message from the DB2 client while connecting to DB2 6.1, you may have to bind the DB2 utility programs to the database. You must apply the procedure to each database you are using with the application server. Here is the binding procedure:

- 1 Log in as your DB2 instance user (such as db2inst1), change to the bnd subdirectory of the instance, and start the DB2 utility.

```
db2inst1@mymachine> cd sqllib/bnd
db2inst1@mymachine> db2
```

- 2 Connect to your database as the DB2 administrator:

```
db2 => connect to mydatabase user db2admin using db2admin
```

You will see the following info:

```
Database Connection Information
Database server           = DB2/NT 6.1.0
SQL authorization ID     = DB2ADMIN
Local database alias     = MYDATABASE
```

- 3 Run the first bind command:

```
db2 => bind @db2ubind.lst blocking all grant public
```

- 4 Run the second bind command:

```
db2 => bind @db2cli.lst blocking all grant public
```


3

Configuring Informix Databases

This chapter describes how to configure your IBM Informix database for use with the Novell exteNd Application Server as a SilverMaster or deployment database. It includes these sections:

- ◆ [Connecting to Informix databases](#)
- ◆ [Troubleshooting](#)



For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Connecting to Informix databases

The Novell exteNd Application Server can access Informix databases through the Informix Type 4 JDBC driver. For supported versions, see the application server [Release Notes](#). For details on drivers, see the IBM Informix Web site at www-3.ibm.com/software/data/informix/.

Follow the steps below to make an Informix database available to the application server:

- Step 1: install the JDBC driver**
- Step 2: create the Informix database and dbspaces**
- Step 3: create the Informix user account**
- Step 4: add the driver path to AGCLASSPATH**
- Step 5: use the database with the application server**

The sections that follow explain these steps.

Step 1: install the JDBC driver

You must be able to connect to the destination Informix server from the machine that will host the application server, so first install the Informix JDBC driver on that machine.

Step 2: create the Informix database and dbspaces

Before you can install the application server, you must create the actual Informix database and the underlying dbspaces. You will need to connect to Informix as a user who is a member of the Informix-Admin group. You can create these objects using either Informix Enterprise Command Center (IECC) or a command-line utility such as DBACCESS.

NOTE: When creating an Informix database for use with the application server, make sure you **turn logging on** for that database.

Step 3: create the Informix user account

You need to create an Informix user account that can be used to connect to the application server. This user account must be a member of the Informix-Admin group. See your Informix documentation for more information. The password for this account **cannot** be null.

Step 4: add the driver path to AGCLASSPATH

The location of the Informix JDBC driver (c:\informixdriver\lib\ifxjdbc.jar for example) must be specified in the AGCLASSPATH variable. For Windows, AGCLASSPATH is a system environment variable. For UNIX, you'll find it in the .agprofile file. For NetWare®, you need to use the setenv command to set AGCLASSPATH (`setenv AGCLASSPATH=path1;path2;path3;...`).


 For details, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Step 5: use the database with the application server

Now you're ready to access the Informix database from the application server. This enables you to do any of the following, as appropriate:

- ◆ Install the application server and point to that database as the **SilverMaster** (as described in *Installing Novell exteNd*)
- ◆ Add that database to the application server for use as a **deployment database**

- ◆ Add a **connection pool** to the application server for accessing that database

 For more information, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Troubleshooting


This section may help you if you run into trouble using Informix and the application server.

Determining driver version To determine which version of the Informix JDBC driver you are using, type the following at the command line:

```
java -cp pathToDriver\lib\ifxjdbc.jar com.informix.jdbc.Version
```

Logical log full You encounter the Informix Server error event Logical Log Full. The application server may not allow you to save data or may not start properly when this problem occurs. (You can view this event in the Informix IECC utility's event list.)

To resolve this problem, back up the logical log or enable Informix's Continuous Log backup.

 For more information about resolving this log file problem, see the *Informix Enterprise Command Center User Guide (IECC)* and the *Informix Administrator's Guide*.

Memory capacity You encounter the Informix Server error event Memory Capacity during SilverMasterInit or while the application server is running. (You can view this event in the Informix IECC utility's event list.)

To resolve this problem, increase the BUFFERS parameter in the ONCONFIG file in the Informix /etc directory. See the ONCONFIG environment variable to find this location.


 For more information about the ONCONFIG file and the BUFFERS parameter, see the *Administrator's Guide for Informix Dynamic Server*.

4

Configuring Microsoft SQL Server Databases

This chapter describes how to use Microsoft SQL Server databases with the Novell exteNd Application Server. It includes information about using Microsoft SQL Server as a SilverMaster or deployment database. It contains these sections:

- ◆ [Basic setup for Microsoft SQL Server databases](#)
- ◆ [Setting up JDBC access](#)
- ◆ [Setting up ODBC access](#)
- ◆ [Using the database with the application server](#)

 For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Basic setup for Microsoft SQL Server databases

You'll perform the following setup steps whether your Novell exteNd Application Server will access a Microsoft SQL Server database via JDBC or ODBC:

- 1 Create the database and configure SQL Server.
- 2 Create a database login to be used by the application server.

Create the database and configure SQL Server

Before you can install the application server, you must first create the actual SQL Server database (and underlying database devices). You can create these objects using either ISQL or a graphical tool such as SQL Enterprise Manager.

Create the database login

Create a login (with a name such as SilverUserLogin) to be used by the application server. Note the following:

- ◆ The login cannot have a null password.
- ◆ You must set the default database for this login to the SilverMaster database you just created. If you do not, you receive error messages during the application server's startup stating **The database schema for SilverMaster is out of sync.**
- ◆ For SQL Server 7, create a login using a SQL Server authentication mode and enable the db_owner role for the desired database on the Database Access tab.
- ◆ The login must have the following permissions: Create Table, Create View, Create SP, Create Default, Create Rule, Backup DB, and Backup Log.

Setting up JDBC access

The application server can connect to Microsoft SQL Server using the JDBC driver supplied by Microsoft.

Install the JDBC driver

You must have the Microsoft SQL Server JDBC driver installed on the machine where the application server will be. It must be able to connect to the destination SQL Server database.

Add the driver path to AGCLASSPATH

The location of the Microsoft SQL Server JDBC driver (JAR files msbase.jar, mssqlserver.jar, and msutil.jar) must be specified in the AGCLASSPATH variable. For Windows, AGCLASSPATH is a system environment variable. For UNIX, you'll find it in the .agprofile file. For NetWare, you need to use the setenv command to set AGCLASSPATH (`setenv AGCLASSPATH=path1;path2;path3;...`).

 For details, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Setting up ODBC access

The application server can connect to Microsoft SQL Server using the ODBC driver supplied by Microsoft as part of the SQL Server client. There are different versions of this ODBC driver.

Install the Microsoft SQL Server client

You must have the Microsoft SQL Server client software installed on the machine where the application server will be. Your client must be configured to connect to the destination SQL Server database.

Create the ODBC System DSN entry


Configure an ODBC System DSN entry for the SilverMaster or deployment database.

- **To create the ODBC System DSN for a Version 7 or 2000 database:**
- 1 Add an entry using **SQL Server** as the driver type.
 - 2 Enter the server name for the target SQL Server.
 - 3 Enter the SQL Server user authentication information for the login you created and click **Next**.
 - 4 Select the proper default database name.
 - 5 Deselect **Use ANSI quoted identifiers**.
 - 6 Deselect **Use ANSI nulls, paddings, and warnings** and click **Next**.
 - 7 Accept the proper language and region settings, then click **Finish**.

Using the database with the application server

Now you're ready to access the Microsoft SQL Server database from the application server. This enables you to do any of the following, as appropriate:

- ◆ Install the application server and point to that database as the **SilverMaster** (as described in *Installing Novell exteNd*)
- ◆ Add that database to the application server for use as a **deployment database**
- ◆ Add a **connection pool** to the application server for accessing that database

 For more information, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

5

Configuring MySQL Databases


This chapter describes how to configure a MySQL database for use with the Novell exteNd Application Server as a SilverMaster or deployment database. Topics include:

- ◆ [About MySQL support](#)
- ◆ [Preparing to access MySQL](#)
- ◆ [Creating MySQL databases](#)
- ◆ [Setting up a SilverMaster database](#)
- ◆ [Setting up a deployment database](#)

About MySQL support

When running on **Windows**, **Linux**, or **NetWare**, the Novell exteNd Application Server supports access to MySQL databases for the following uses:

- ◆ As the application server's **SilverMaster database**
- ◆ As a **deployment database** or **connection pool**

 To learn more about these uses, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

DBMS and driver versions The application server accesses MySQL through the Connector/J driver. For details on the supported versions of this DBMS and driver, see the Novell exteNd Application Server [Release Notes](#).

MySQL Web site You can visit www.mysql.com to get additional information about MySQL.

Preparing to access MySQL

Before the application server can start using MySQL databases, you need to make sure that:


- ◆ A supported version of the **MySQL DBMS** is installed on one or more host machines accessible to the application server (often including the application server machine itself)
- ◆ A supported version of the **Connector/J driver** is installed on the application server machine

For details, see the section below for your situation:

- ◆ [When MySQL is installed for you](#)
- ◆ [When you install MySQL yourself](#)

When MySQL is installed for you

On NetWare and for appropriate Novell exteNd configurations on Windows and Linux, the exteNd install provides all of the MySQL software (MySQL DBMS and Connector/J driver) you need for the application server. In these cases, the exteNd install also takes care of all necessary DBMS and driver configuration for you.

 For more information, see [Installing Novell exteNd](#).

When you install MySQL yourself

In some cases, you may want or need to obtain and install the MySQL software yourself:

- ◆ [MySQL DBMS](#)
- ◆ [Connector/J driver](#)

MySQL DBMS

If you don't already have the MySQL servers you need, set them up and make sure they can be accessed by your application server machine. Follow these steps (note that the examples shown are for Linux):

- 1 Obtain the MySQL DBMS and install it on an appropriate host machine.
- 2 Once your MySQL server is installed, set up the **configuration file** for it by using one of the templates in `/usr/share/mysql`:
 - ◆ `my-huge.cnf`
 - ◆ `my-large.cnf`
 - ◆ `my-medium.cnf`
 - ◆ `my-small.cnf`

Choose the appropriate template (based on size of RAM) and copy it to `/etc/my.cnf`.

- 3 Edit the `my.cnf` file to change the `max_allowed_packet` setting from the default of 16M to 8M (required to run `SilverMasterInit`):

```
set-variable = max_allowed_packet=8M
```

- 4 If you want the MySQL server to support **distributed transactions**, uncomment the following lines in the `my.cnf` file:

```
# Uncomment the following if you are using InnoDB tables
innodb_data_home_dir = /var/lib/mysql/
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /var/lib/mysql/
innodb_log_arch_dir = /var/lib/mysql/
```

Note that when you later create tables for use in distributed transactions, they must be specified as **type InnoDB**. For example:

```
CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
species VARCHAR(20), sex CHAR(1), birth DATE, death DATE)
TYPE InnoDB;
```

- 5 Add the following lines to the `my.cnf` file:

```
transaction-isolation = READ-COMMITTED
innodb_flush_log_at_trx_commit = 1
```

- 6 To turn on **logging** (recommended), add this line to the `my.cnf` file:

```
log-bin = filename.ext
```

Connector/J driver

You need to set up the Connector/J driver so the application server can use it when connecting to MySQL. Follow these steps:

- 1 Obtain Connector/J and install it on the application server machine.
- 2 Specify the location of the Connector/J driver's **JAR file** by setting the `AGCLASSPATH` environment variable (in the `.agprofile` file on Linux). The JAR file is named:

```
mysql-connector-java-version-bin.jar
```



For more about setting `AGCLASSPATH`, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Creating MySQL databases

This section provides basic instructions for manually creating a MySQL database that you can then use as a SilverMaster, deployment database, or connection pool.

NOTE: On NetWare and for appropriate Novell exteNd configurations on Windows and Linux, the exteNd install can automatically create the MySQL database for your SilverMaster. In these cases, you aren't required to manually create the MySQL database ahead of time.

➤ To manually create a MySQL database:

- 1 From the command prompt, start the **MySQL monitor** by typing:

```
mysql
```

- 2 To **create your database**, type:

```
create database databasename;
```

For example:

```
create database salesdb;
```

- 3 You must **grant access rights** for this database to the MySQL user through which the application server will be connecting. Type:

```
grant all on databasename.* to username@localhost identified by  
'password';
```

For example:

```
grant all on salesdb.* to appserver@localhost identified by  
'secretpw';
```

- 4 To **exit** from the MySQL monitor, type:

```
quit
```

Setting up a SilverMaster database


The SilverMaster is a database in which the application server maintains its system information. Because the SilverMaster is configured when you install the application server, this database must be available at that time.

You can use a MySQL database as your application server's SilverMaster. For details, see the section below for your situation:

- ◆ [When SilverMaster is set up for you](#)
- ◆ [When you set up SilverMaster yourself](#)



When SilverMaster is set up for you

On NetWare and for appropriate Novell exteNd configurations on Windows and Linux, the exteNd install can automatically create and configure a MySQL database for the SilverMaster when installing your application server.

 For more information, see *Installing Novell exteNd*.


When you set up SilverMaster yourself

In some cases, you may want or need to set up a MySQL database for the SilverMaster yourself:

- 1 **Create** the MySQL database ahead of time.
 For the manual creation steps, see “[Creating MySQL databases](#)” on [page 36](#).
- 2 **Start** the exteNd install and specify that database when you’re prompted for SilverMaster settings.
 For more information, see *Installing Novell exteNd*.

Setting up a deployment database

It’s common to deploy your J2EE applications to the SilverMaster database, but you can also add other databases to the application server anytime and deploy to them instead. To set up a MySQL deployment database, you must:

- 1 **Create** the MySQL database.
 For the manual creation steps, see “[Creating MySQL databases](#)” on [page 36](#).
- 2 **Add** that database to the application server. Use either of the following:
 - ◆ The [Add Database dialog](#) in the SMC (Server Management Console)
 - ◆ The [AddDatabase command](#) of the SilverCmd command-line tool

You’ll supply the database name, user name, and password (as specified when the database was created), and choose the MySQL database platform and driver set. For the JDBC URL, specify the following (replacing `%HOST%` and `%DBNAME%` with the appropriate MySQL host address and database name):

```
jdbc:mysql://%HOST%/%DBNAME%
```

For example:

```
jdbc:mysql://localhost/salesdb
```

If you want to add a **connection pool** to the application server (for accessing data from a MySQL database), the steps are similar.

6

Configuring Oracle Databases

This chapter describes how to configure your Oracle database for use with the Novell exteNd Application Server as a SilverMaster or deployment database. It includes these sections:

- ◆ [Basic setup for Oracle databases](#)
- ◆ [Setting up an Oracle driver](#)
- ◆ [Using the database with the application server](#)
- ◆ [Using synonyms](#)
- ◆ [Troubleshooting](#)



For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Basic setup for Oracle databases

You'll perform the following setup steps no matter which driver your Novell exteNd Application Server will use to access an Oracle database:

- 1 Create an Oracle tablespace for use with SilverMaster.
- 2 Create an Oracle user for the application server.

Create an Oracle tablespace for use with SilverMaster

Creating a new Oracle tablespace for use with SilverMaster ensures optimal performance and ease of installation. To create one, run SQL Plus and connect to Oracle as System or SYSDBA.

For example, this statement creates a tablespace with an initial size of 30 MB:

```
CREATE TABLESPACE SilverTablespace
DATAFILE 'c:\databases\silvertablespace.dbf' SIZE 30M;
```

You can issue this statement from a SQL Plus prompt. If you choose to use an existing tablespace, make sure there is enough space available.

Create an Oracle user for the application server

To connect the application server to Oracle, create a new Oracle user for exclusive use by the application server. This user will own the SilverMaster tables.

Issue the following statements using the SQL Plus utility. These statements create the user and set the user's privileges.

```
CREATE USER smuser IDENTIFIED BY password
DEFAULT TABLESPACE SilverTablespace
QUOTA UNLIMITED ON SilverTablespace;
GRANT CONNECT, RESOURCE to smuser;
REVOKE UNLIMITED TABLESPACE from smuser;
```

Setting up an Oracle driver

The application server connects to Oracle databases using one of the following drivers:

- ◆ Oracle Thin JDBC Driver
- ◆ Oracle OCI Driver
- ◆ A Type 2 JDBC driver written and provided by Novell (this driver interfaces with the Oracle SQL Net Client to achieve connectivity to the Oracle server)

 See the application server's [Release Notes](#) for which versions of Oracle databases you can use with which drivers.

Using the Oracle Thin JDBC Driver

The application server can connect to Oracle databases using the thin JDBC driver supplied by Oracle.

Install the driver

You must have the Oracle Thin JDBC Driver installed on the machine where the application server will be. It must be able to connect to the destination Oracle database.

Add the driver path to AGCLASSPATH

The location of the Oracle Thin JDBC Driver (for example, *DriverPath*/ojdbc14.jar and *DriverPath*/nls_charset12.zip) must be specified in the AGCLASSPATH variable. For Windows, AGCLASSPATH is a system environment variable. For UNIX, you'll find it in the .agprofile file. For NetWare, you need to use the setenv command to set AGCLASSPATH (`setenv AGCLASSPATH=path1;path2;path3;...`).



For details, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Using the Oracle OCI Driver

The application server can connect to Oracle 9i databases using the OCI driver supplied by Oracle.

Install the driver

You must have the Oracle OCI Driver installed on the machine where the application server will be. It must be able to connect to the destination Oracle database.

Add the driver path to AGCLASSPATH

The location of the Oracle OCI Driver (for example, *DriverPath*/ojdbc14.jar and *DriverPath*/nls_charset12.zip) must be specified in the AGCLASSPATH variable. For Windows, AGCLASSPATH is a system environment variable. For UNIX, you'll find it in the .agprofile file.



For details, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Using the Novell exteNd Oracle driver

The application server can connect to Oracle databases using the Type 2 JDBC driver supplied by Novell. Setup involves these steps:

- 1 Install the Oracle SQL Net client onto the machine where the application server will reside.
- 2 Create a SQL Net Alias for use by the application server.

Install Oracle SQL Net Client

If Version 2.3.4.0.0 of the Oracle SQL Net Client is not installed on the application server machine, install it now before you continue. If you do not have this version of SQL Net, obtain it from Oracle Corporation. Using the Oracle CD, follow the steps to install the client for an application user.

If you already have the client installed, you can run Oracle Installer (from the Oracle for Windows NT Program Folder) to determine the version of SQL Net that is installed.

Create an Oracle SQL Net Alias

In order to connect to the Oracle server, you must have a valid Oracle SQL Net Alias that references the target Oracle server.


➤ To create an Oracle SQL Net Alias:

- 1 Run SQL Net Easy Configuration from the Oracle for Windows NT Program Group.
NOTE: On UNIX, run `net8wiz.sh` from the `$ORACLE_HOME/bin/` directory.
- 2 Choose **Add a New Database Alias**.
- 3 Enter the name of the alias (you specify this name during the application server installation).
A suggested name for this alias is **SilverMasterORCL**.
- 4 Choose **TCP/IP** as the protocol and select **OK** to continue.
- 5 Enter the TCP/IP Host Name for the Oracle server and enter the correct Instance name.
- 6 Choose **OK** to continue.
- 7 Choose **Yes** to create the new alias. At this point you can exit the SQL Net Easy Configuration utility.
- 8 Test the newly created SQL Net Alias by using SQL Plus to connect to the Oracle database server.

Using the database with the application server

Now you're ready to access the Oracle database from the application server. This enables you to do any of the following, as appropriate:

- ◆ Install the application server and point to that database as the **SilverMaster** (as described in *Installing Novell exteNd*)
- ◆ Add that database to the application server for use as a **deployment database**
- ◆ Add a **connection pool** to the application server for accessing that database

 For more information, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Creating a deployment database on Oracle

Adding an Oracle database to the application server equates to adding an Oracle user. Adding a database does two things:

- ◆ Adds all the tables and views that the database user has access to
- ◆ Creates the application server's system tables to store metadata

Before adding Oracle databases


Storing system tables separately If you want to store the application server's system tables in a separate database account, create an additional database user as described above and, if using the Novell exteNd Oracle driver, create a new Oracle SQL Net Alias. The system tables may also be stored in any other databases supported by the application server.

Using a different character set You can tell the Novell exteNd Oracle 8 driver to use a different character set than what is specified in the Oracle instance. You would do this when the character set encoding of the data is different than the character set specified by the Oracle instance. A white paper about this topic is available on the Novell exteNd Application Server product support site.

Adding a database

Specifying the database if you are using the OCI or Novell exteNd Oracle driver When adding an Oracle database to the application server, you are asked for the name of the database. If you are using the OCI or Novell exteNd Oracle driver, the database name refers to the Oracle SQLNet Alias or tnsnames.ora entry. The SQLNet Alias points to a specific Oracle Database Instance on a specific server. Typically you have **one** per Oracle database.

The application server requires a unique database name. This means that it requires you to create a unique SQLNet Alias for each Oracle database user you add. So if you have SilverMaster, application tables, and the application server's system tables stored in three different Oracle database users, you need three unique SQLNet Aliases—one for each user. Each SQLNet Alias points to the same Oracle database.

 Optionally, if you do not want to configure multiple SQLNet Aliases, see [“Using one SQLNet Alias” on page 44](#).

Oracle 8i notes

Note the following:

Topic	Description
Using databases containing periods in their names	<p>When using Oracle 8i, you cannot use a database name that contains a period. You can successfully add a database with a period in the name, but later interactions between the application server and the database will result in NullPointerException errors and other unwanted behavior.</p> <p>If you are using the Novell exteNd Oracle driver, the tnsnames.ora entry is of the form <i>myalias.domain.com</i>. On Windows, it is sufficient to specify only <i>myalias</i> for the database name when adding a database to the application server and the connection to the database will be successful. But on most UNIX platforms this truncated specification will not succeed in a match for the entry in tnsnames.ora. When you add the database to the application server, it will result in the error TNS names entry not found.</p>
Verifying your connection	<p>The application server's database connection behavior is consistent with that of SQLPlus. The best way to troubleshoot your connection to Oracle 8i is to first verify the connection through SQLPlus. Be sure to connect only to <i>myalias</i> as that is what is needed for the application server.</p> <p>For example, try to connect using sqlplus <i>login/password@myalias</i> rather than sqlplus <i>login/password@myalias.domain.com</i>.</p> <p>If your connection to <i>myalias</i> does not succeed using SQLPlus, you may need to edit the tnsnames.ora entry and remove the <i>domain.com</i> from the entry.</p>

Using one SQLNet Alias

This is an optional advanced feature if you are using the OCI or Novell exteNd Oracle driver. By default, the Add Database process assumes you are adding an Oracle database with a separate SQLNet Alias (TNS NAME) for each Oracle database user that you use in the application server environment.

The steps that follow are for experienced users. They document how you can install using a single SQLNet Alias.

➤ To install using a single SQLNet Alias:

- 1 When adding a new database to the application server, do not enter the SQLNet Alias (TNS NAME) in the database name box; enter any name you want.
- 2 Now enter the user name and password as before and select **Oracle 7** or **Oracle 8** as the database platform.

- 3 Choose **User-specified JDBC driver** for the driver set.
- 4 Click **Next**.

You can now enter the JDBC connection string.

- 5 For the JDBC URL, enter:

```
jdbc:ssw:oracle:TNSNAME
```

replacing *TNSNAME* with the TNS NAME in your tnsnames.ora file.

You can now connect as many databases as you like using the same TNS NAME. Make sure you store the system tables separately.

What to avoid You can reduce the chance of encountering problems if you follow these rules:

- ◆ Do not put anything else in the same Oracle database user as your SilverMaster.
- ◆ Do not put multiple system tables in the same Oracle database user. These would conflict with each other. When adding multiple databases to the application server, you should keep their system tables separate.

Terminology differences Different DBMSs involve different terminology. The application server uses the term **databases**, but on Oracle these map to **Oracle database users**. The following table describes other differences.

Application server term	Oracle term
SilverMaster database	Oracle database user that owns the SilverMaster tables
Add a database	Add a specific user in an Oracle database

Using synonyms

The application server provides support for Oracle 8 and later synonyms for tables and views.

NOTE: With Oracle, you can also define synonyms for other kinds of objects, including object tables, object views, and stored procedures. The application server supports synonyms only for tables and views.

Specifying the driver

In order to enable synonym support, you need to specify one of the following driver sets when adding the Oracle database to the application server:

- ◆ **Thin Oracle JDBC driver with synonym/mixed case support**—Oracle’s thin JDBC driver with synonym support
- ◆ **OCI Oracle JDBC driver with synonym/mixed case support**—Oracle’s OCI driver with synonym support
- ◆ **Novell exteNd JDBC driver with synonym/mixed case support**—the Novell exteNd Oracle 8 driver with synonym support

Types of synonyms

Synonyms can be either private or public. A **public synonym** is owned by the special user group PUBLIC; every user of a database can access it.

A **private synonym** is in the schema of a specific user, who has control over its availability to others (see Oracle’s *Oracle 8 Concepts* manual). Individual users can see only the public synonyms and their own private synonyms.

Enabling and disabling synonym support

There is a property called **includeSynonyms** in the platforms.dbl file (in the application server’s Resources directory) that specifies whether a driver supports synonyms. The property is in the PROPERTIES section for the driver.

```
DRIVER
/Name ...
/JdbcDriver ...
/DriverSet ...
...

PROPERTIES
/includeSynonyms      "true"
```

If you want to use one of these drivers but want synonym support disabled, set the **includeSynonyms** property to **false**. Similarly, if you have written your own Oracle driver and have implemented synonym support, to enable the support you need to set the **includeSynonyms** property to **true** in the PROPERTIES section for your driver.

Troubleshooting

This section describes the most common errors and workarounds.

ORA-01019 “unable to allocate memory in the user side” If you receive this message while running with a Net8 Oracle Client, free up available memory on the machine where the application server is installed.

ORA-01460 “unimplemented or unreasonable conversion requested” If you receive this message when SilverMasterInit is creating an Oracle SilverMaster or when you are attempting to add an Oracle database to your application server, you must upgrade to at least Version 8.0.4 of the Oracle client software. This problem occurs on versions before 8.0.4.

ORA-01631, 00000, “max # extents (%s) reached in table %s.%s” This message is caused when a table tried to extend past maxextents. To solve this problem, check your INITIAL, NEXT, and PCTINCREASE parameters on the DEFAULT TABLESPACE for the SilverMaster user. You must get your DBA to increase them. For example:

```
SELECT DEFAULT_TABLESPACE
FROM dba_users
WHERE username = 'MYUSER';
SELECT *
FROM dba_tablespaces
WHERE tablespace_name = 'MYTABLESPACE';
ALTER TABLESPACE MYTABLESPACE
DEFAULT STORAGE (INITIAL 10K, NEXT 10K, PCTINCREASE 50);
```

ORA-01950, 00000, “no privileges on tablespace %s” This message is caused when a user does not have privileges to allocate an extent in the specified tablespace. To solve this problem, grant QUOTA to the user on the mentioned database. For example:

```
ALTER USER MYUSER
QUOTA UNLIMITED ON MYTABLESPACE;
```


TNS-12154 "TNS: could not resolve service name" If you receive this message, you must verify that you have correctly created the SQL Net Alias as described in [“Create an Oracle SQL Net Alias” on page 42](#).

Tablespaces for Oracle databases The installation will use the default tablespace of the user name/password specified for your SilverMaster database. You can usually set the default tablespace with the following SQL statement:

```
ALTER USER username DEFAULT TABLESPACE mytablespace ;
```

Adding large numbers of tables If you are planning to connect the application server to a large Oracle tablespace, open the Server Management Console (SMC) and change the value of the timeout for server requests to 30,000 seconds.

You must modify this value before attempting to add any large Oracle application. Once the application has been added successfully, reset the value back to the default value (3,000). The application server must be restarted for these changes to be applied.

 For more information, see the section on [setting performance parameters](#) in the *Administrator’s Guide*.

Using different database types with the same application server If you install with a non-Oracle SilverMaster and want to add an Oracle database, you need to have \$ORACLE_HOME set and have \$ORACLE_HOME/lib in your SHLIB_PATH (for HP-UX), LD_LIBRARY_PATH (for Solaris and Linux), or LIBPATH (for AIX).

Adding an Oracle 8i database on Solaris On Solaris, if you are not using an Oracle 8i database as your SilverMaster but are using Oracle 8i client software and want to add an Oracle 8i database to your server, you must copy a library file in the \$EXTENDAPPSERVERROOT/bin directory. By default, the installed Oracle driver library file is appropriate for Oracle 8.

The following procedure will allow you to use the Oracle 8i library file instead:

- 1 Back up the libagora8_jni.so file:

```
# cd $EXTENDAPPSERVERROOT/bin
# cp libagora8_jni.so libagora8_jni.so.orig
```
- 2 Copy the Oracle 8i driver library file:

```
# cp libagora8_jni.so.8i libagora8_jni.so
```

Connecting to Oracle from HP-UX or AIX If you are using the Novell exteNd Oracle driver, in order to connect to an Oracle database from HP-UX or AIX you must have at least the following Oracle components installed on the HP-UX or AIX machine:

- ◆ Net8
- ◆ Net8 External Naming Adapters
- ◆ Net8 Protocol Adapters
- ◆ Oracle Intelligent Agent
- ◆ PL/SQL
- ◆ SQL*Plus

In addition, during the Oracle installation process you will be asked if you want to relink Oracle product executables. You must answer **yes**. By default, the relinking will not occur.

If the Oracle installation is not sufficient for the application server, you will get the following error during SilverMasterInit:

```
/usr/lib/dld.sl: Unresolved symbol: opinit (code) from
/home/username/exteNdAppServer/bin/libagora_jni.sl
```


If you encounter this error or another type of connection error, verify that you have installed the Oracle components listed above and can connect to your database using SQL*Plus.

7

Configuring Sybase Adaptive Server Anywhere Databases

This chapter describes how to configure your Sybase Adaptive Server Anywhere (ASA) database for use with the Novell exteNd Application Server as a SilverMaster or deployment database. It includes these sections:

- ◆ [Preparing to use Sybase ASA](#)
- ◆ [Setting up SilverMaster and deployment databases](#)

 For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).


Preparing to use Sybase ASA

Before you can set up Sybase ASA databases for use with the application server, you must:

- ◆ Install Sybase ASA
- ◆ Prepare to use the jConnect driver

Installing Sybase ASA

If you don't have a supported version of the Sybase ASA software already installed, you need to obtain it from Sybase and install it yourself.

 For more information, go to www.sybase.com.

Preparing to use jConnect

The application server connects to Sybase ASA using **jConnect** (Sybase's JDBC driver). jConnect is included as part of the Sybase ASA installation.

When you install ASA using the Sybase wizard, the driver is installed in the Sybase **Shared** directory. To use jConnect, you need to do this one-time setup:

- ◆ **Set AGCLASSPATH** (environment variable for the server's classpath) on the machine where your application server is running to include the path to the jConnect 5.5 driver:

```
AGCLASSPATH=SybaseInstallDir\Shared\jConnect-  
5_5\classes\jconn2.jar
```

For example, if you installed ASA in the default location, you would specify this in AGCLASSPATH:

```
AGCLASSPATH=c:\Program Files\Sybase\Shared\jConnect-  
5_5\classes\jconn2.jar
```



For more information on listing JDBC driver JARs via AGCLASSPATH, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Setting up SilverMaster and deployment databases

To set up a Sybase ASA database as a SilverMaster or deployment database, do the following:

- 1 Create a database.
- 2 Create a user account to be used by the application server.
- 3 Configure Sybase ASA to run as a service.
- 4 Make the database available to the application server.

The following sections describe these steps.

Step 1: Create a database

To create your ASA database, use **Sybase Central**. If this will be the SilverMaster database, you must create it before installing the application server.




For more information on creating ASA databases, see the Sybase Central online help.

Step 2: Create user accounts


Sybase ASA provides a default user **dba** with the password **sql**. You can use this user name and password pair for your SilverMaster database or any other Sybase ASA database.

However, it is strongly recommended that you set up separate accounts for your SilverMaster and for each deployment database; this way you can easily tell which applications are hitting your database when and how often, and you can more easily troubleshoot performance problems. Make sure these accounts have at least Resource and Connect privileges.

 For more information about setting up user accounts, see the Sybase Central online help.

Step 3: Run Sybase ASA as a service

It is recommended that you run your Sybase ASA databases as services to make database use and maintenance easier.

 For information about running Sybase ASA as a service, see the Sybase ASA online help. (In that help, select *ASA Database Administration Guide*. Go to “Running the server outside the current session” in Chapter 1.)

Step 4: Make the database available to the application server

If you’re using ASA for the SilverMaster database, you can install the application server and point to your ASA database as SilverMaster (as described in [Installing Novell exteNd](#)).

If you’re using an ASA database as a deployment database, you can make that database available by **adding it to the application server**:

- 1 Make sure the database is **running**. If it is not running as a service, you can start it using Sybase Central.
- 2 Using the **SMC**, add the database to the server.


Specify the name of the database, the user name and password, and the JDBC URL for the database, which has the following format:

```
jdbc:sybase:Tds:hostname:port?SERVICENAME=dbname
```

For example, if your database is named MyDB and you are running it locally at the default ASA port of 2638, your JDBC URL would be:

```
jdbc:sybase:Tds:localhost:2638?SERVICENAME=MyDB
```

Another way to make a database available to the application server is by **defining a JDBC connection pool** for it.

 For more information on adding databases to the server and defining connection pools, see the chapter on [data source configuration](#) in the *Administrator's Guide*.


8

Configuring Sybase Adaptive Server Enterprise Databases

This chapter describes how to use Sybase Adaptive Server Enterprise databases with the Novell exteNd Application Server. It includes information about using Adaptive Server Enterprise as a SilverMaster or deployment database. It contains these sections:

- ◆ [Setting up Sybase Adaptive Server Enterprise databases](#)
- ◆ [Troubleshooting](#)

NOTE: Sybase Adaptive Server Enterprise was formerly known as Sybase SQL Server.

 For the latest information on supported database versions, see the Novell exteNd Application Server [Release Notes](#).

Setting up Sybase Adaptive Server Enterprise databases

This section describes how to create a SilverMaster database (and deployment databases) using Sybase Adaptive Server Enterprise.

First, **install the Sybase client** onto the machine where the Novell exteNd Application Server will reside. Then perform these steps:

- 1 Install the Sybase jConnect classes and stored procedures.
- 2 Configure the Sybase server.
- 3 Create database devices and the actual SilverMaster database as a Sybase database.
- 4 Create a database login to be used by the application server.
- 5 Use the database with the application server.


Step 1: install Sybase jConnect

You must install and configure the jConnect driver on the machine where you will be installing the application server. This driver must be configured before you install the application server.

NOTE: It's recommended that you use jConnect 5.x.

➤ To install Sybase jConnect:

- 1 Install the jConnect **driver**.

 For more information about Sybase jConnect, see www.sybase.com.

- 2 Make sure the location of the jConnect driver is specified in the **AGCLASSPATH** variable on your application server machine. For Windows, **AGCLASSPATH** is a system environment variable. For UNIX, you'll find it in the `.agprofile` file. For NetWare, you need to use the `setenv` command to set **AGCLASSPATH** (`setenv AGCLASSPATH=path1;path2;path3;...`).

If you are using jConnect 4:

```
AGCLASSPATH=installDir\jConnect-version\classes
```


If you are using jConnect 5:

```
AGCLASSPATH=installDir\jConnect-version\classes\jconn2.jar
```

 For more information on listing JDBC driver files via **AGCLASSPATH**, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

NOTE: You do not need to configure the **CLASSPATH** or **JDBC_HOME** variables mentioned in the Sybase installation notes if you are using the driver exclusively with the application server.

- 3 Install the jConnect **stored procedures** onto the Sybase server.

 For more information, see the Sybase *jConnect for JDBC Installation Guide* or www.sybase.com.

Step 2: configure the Sybase server

You must configure the Sybase server to support the application server. These configuration changes involve sizing your Sybase **TEMPDB** database appropriately and configuring the correct amount of Sybase procedure cache to support the jConnect stored procedures.

➤ To configure the Sybase server:

- 1 From Sybase Central, connect to the destination Sybase server and select **Databases**.

You need to connect to the Sybase server as **SA** or an equivalent.

- 2 Right-click the **TEMPDB** database and select **Properties**.

- 3 From the Usage tab, check the **Total Allocated value**.
If the TEMPDB Total Allocated value is less than 30 MB:
 - ◆ Create a new 30 MB database device (from within the Database Devices folder) called **TempExtend**.
 - ◆ When this process completes, right-click the **TEMPDB** database (from within the Databases folder).
 - ◆ From the **Devices** tab, add the new TempExtend device as a DATA device.
- 4 Right-click the Sybase server and select **Configure**.
- 5 Scroll down until you see the value **Total Memory**.
The value for total memory is in 2K pages, so the numeric value shown times 2 equals the amount of memory (in kilobytes) currently allocated to Sybase. Take note of this value and then find the value **Procedure Cache Percent**. This value should be the percentage equivalent to 13 MB of memory.
To find this value, divide 13,000 by the Total Memory amount (in kilobytes). This will give you the percentage that you should enter as the Procedure Cache Percent value.
For example, the default amount of memory allocated to Sybase Adaptive Server Enterprise 11.5 is 9000 pages (equivalent to 18,000 kilobytes). If you divide 13,000 by 18,000, the value for Procedure Cache Percent equals .72 or 72%. Enter 72 as the new value for the Procedure Cache Percent.
- 6 **Restart** the Sybase server for the change to take effect.

Step 3: create the SilverMaster database in Sybase

Before you can install the application server, you must first create the actual Sybase database (and its underlying database devices). You can create these objects using either ISQL or a graphical tool such as Sybase Central. The following procedure describes how to use Sybase Central to create the objects.

➤ To create the SilverMaster database:

- 1 Connect to Sybase using SA or an account with SA equivalence.
- 2 From Sybase Central, connect to the destination Sybase server and select **Database Devices**.
- 3 Create a new 30 MB database device called **SilverMasterData**.
- 4 When this process completes, create another new 30 MB device called **SilverMasterLog**.
- 5 From the Databases folder in Sybase Central, create a new database called **silvermaster** (this is case sensitive).
- 6 Create the database using the two devices that you created above, using all 30 MBs of each respective device for data and for log.

- 7 Once the database has been created, right-click the database and select **Properties**.
- 8 From the Options tab, select **Truncate Log on Checkpoint**.

Step 4: create the Sybase login for the application server

Once the jConnect driver is installed and the SilverMaster database has been created, you must create a Sybase login for use by the application server.

➤ To create the Sybase login:


- 1 From Sybase Central, connect to Sybase using SA or an account with SA equivalence.
- 2 From the Login folder, add a new Login (for example, you could call it **smuser**). You must assign this account a password of at least six characters.
- 3 Set the default database for this login to **silvermaster** and complete the login creation process.
- 4 From the Databases folder, right-click the **silvermaster** database and select **Properties**.
- 5 Click the **Change** button to change the owner of the database from SA to the **smuser** login.

You also need to define logins for each deployment database you will use. You must specify CREATE TABLE permission if you are going to store the application server's system tables in the deployment database (this is the default behavior for deployment databases that you add to the server).

Step 5: use the database with the application server

Now you're ready to access the Sybase Adaptive Server Enterprise database from the application server. This enables you to do any of the following, as appropriate:

- ◆ Install the application server and point to that database as the **SilverMaster** (as described in *Installing Novell exteNd*)
- ◆ Add a database to the application server for use as a **deployment database**
- ◆ Add a **connection pool** to the application server for accessing a database

 For more information, see the chapter on [data source configuration](#) in the *Administrator's Guide*.

Troubleshooting

Adaptive Server Enterprise troubleshooting

This section describes the known problems and workarounds for Sybase Adaptive Server Enterprise databases with the Novell exteNd Application Server.

JZOC0 errors

You may receive error messages from jConnect with an error code of JZOC0 and the error message **Connection already closed**. If you receive this error, verify that you are running Sybase server Version 11.0.3.2. If your Sybase version is not 11.0.3.2, obtain the updates through Sybase technical support.

If you cannot update your Sybase server version, separate the application server's system tables from the Sybase database to work around the problem.

Logins, users, and permissions

The easiest way to avoid potential problems with logins, users, and permissions is to use SA. If problems occur while adding a database, Novell exteNd Technical Support may ask you to try adding the database as SA to determine if the problem is with logins, users, or permissions.

However, in many environments it is not practical to obtain the SA password. The following SQL examples show you how to create a login and user with the correct permissions for the application server.

- ◆ **Login** The login defines access to an Adaptive Server Enterprise database, as well as the default database and password. Use this SQL script to create a login called **SilverUser**:

```
USE master
go
EXEC sp_addlogin SilverUser', 'PASSWORD', 'mydatabase',
'us_english', "SilverUser"
go
```

- ◆ **User** A user defines a set of permissions for a particular database and is associated with a login. Use this SQL script to create a user called **SilverUser**:

```
EXEC sp_adduser 'SilverUser', 'SilverUser', 'public'
go
GRANT CREATE TABLE TO SilverUser
go
```

Primary keys and foreign keys

The primary and foreign keys of older databases (from Sybase System 10 and earlier) may still be defined using system stored procedures named **sp_primarykey** and **sp_foreignkey**. Sybase no longer recommends creating primary and foreign keys in this manner. In fact, newer Sybase products, including jConnect, do not recognize keys created that way. You can determine whether your database uses these stored procedures by examining the DDL used to create the database.

Primary keys

This is what the DDL will look like with an older-style primary key:

```
CREATE TABLE dbo.U_LBX_ADVICE
(
  advice_key smallint NOT NULL,
  <PAYMENT_KEY file_key smallint NOT NULL,
  batch_key smallint NOT NULL,
  line_no varchar(20) NULL,
  item_paid_id varchar(20) NULL,
  item_paid_amount money NULL
)
go
EXEC sp_primarykey 'dbo.U_LBX_ADVICE',
  advice_key,payment_key,file_key,batch_key
```

The newer Sybase Adaptive Server Enterprise format for creating primary keys is part of the CREATE TABLE syntax:

```
create table U_LBX_ADVICE
(
  advice_key smallint not null,
  payment_key smallint not null,
  file_key smallint not null,
  batch_key smallint not null,
  line_no varchar(20) null,
  item_paid_id varchar(20) null,
  item_paid_amount money null,
  constraint PK_U_LBX_ADVICE primary key
  (advice_key, payment_key, file_key, batch_key)
)
```

If your table already exists and you cannot recreate the table with the CREATE TABLE statement, then use ALTER TABLE as shown below:

```
ALTER TABLE dbo.U_LBX_ADVICE ADD CONSTRAINT newprimarykey PRIMARY
KEY NONCLUSTERED (advice_key,payment_key,file_key,batch_key)
```

Foreign keys

Foreign keys define the relationship between tables. The problem with foreign keys is very similar to the one with primary keys described above: Sybase's jConnect does not recognize foreign keys created in the older System 10 format using `sp_foreignkey`. This is the older style DDL that would be used to create foreign keys:

```
exec sp_foreignkey dbo.U_LBX_ADVICE, dbo.U_LBX_BATCH,  
file_key,  
batch_key
```

The syntax for adding a similarly defined foreign key is:

```
ALTER TABLE dbo.U_LBX_ADVICE ADD CONSTRAINT myforeignkey FOREIGN  
KEY (file_key,batch_key)  
REFERENCES sqlfinar1.dbo.U_LBX_BATCH (file_key,batch_key)  
go
```

You can use the Sybase PowerDesigner tool to update the DDL that creates a database to the newer format. To use this tool, follow these guidelines after backing up your database:

- 1 From PowerDesigner's menu, choose **File/Reverse Engineering**.
- 2 Enter either the file containing your database's DDL or the name of an ODBC driver to connect directly to your database.
- 3 Choose **Database/Generate Database** to create a new DDL that will define primary and foreign keys with the correct syntax.
- 4 As with any tool, double-check and test the new syntax for any possible omissions.

jConnect troubleshooting

If the jConnect stored procedures are not properly installed and the problem is not detected, you can have problems adding a database.

One way

Do this to verify that jConnect is properly installed:

- ◆ Open a DOS window and verify that your **AGCLASSPATH** environment variable points to your **jConnect\classes** directory.

Another way

Another way to determine if jConnect is successfully installed is to try running the **sample jConnect applications** (make sure you have installed the samples: check that a **classes\sample** directory exists). Type the following lines at a DOS prompt.

- ◆ If you are using jConnect 4:

```
set CLASSPATH=installDir\jConnect-version\classes
```

```
java sample.SybSample Validate -U sa -P password -S  
jdbc:sybase:Tds:servername:port
```

- ◆ If you are using jConnect 5:

```
set CLASSPATH=installDir\jConnect-version\classes\jconn2.jar
```

```
java sample2.SybSample Validate -U sa -P password -S  
jdbc:sybase:Tds:servername:port
```

These commands will start an application that can launch a variety of jConnect samples (click **File/Open Sample**).

Index

A

- Adaptive Server Anywhere
 - see Sybase Adaptive Server Anywhere
- Adaptive Server Enterprise
 - see Sybase Adaptive Server Enterprise

C

- Cloudscape
 - see IBM Cloudscape

D

- DB2
 - see IBM DB2

I

- IBM Cloudscape
 - configuring databases for Novell exteNd Application Server 11
- IBM DB2
 - configuring databases for Novell exteNd Application Server 17
- IBM Informix
 - configuring databases for Novell exteNd Application Server 25
- Informix
 - see IBM Informix

M

- Microsoft SQL Server
 - configuring databases for Novell exteNd Application Server 29

MySQL

- configuring databases for Novell exteNd Application Server 33
- creating databases 36
- deployment databases 37
- installing on Linux 34
- installing on NetWare 34
- installing on Windows 34
- SilverMaster use on Linux 36
- SilverMaster use on NetWare 36
- SilverMaster use on Windows 36

O

- Oracle
 - configuring databases for Novell exteNd Application Server 39
 - synonyms for tables and views 45

S

- SQL Server, Microsoft
 - see Microsoft SQL Server
- SQL Server, Sybase
 - see Sybase Adaptive Server Enterprise
- Sybase Adaptive Server Anywhere
 - configuring databases for Novell exteNd Application Server 49
- Sybase Adaptive Server Enterprise
 - configuring databases for Novell exteNd Application Server 53
- Sybase SQL Server
 - see Sybase Adaptive Server Enterprise
- synonyms, Oracle
 - support for 45

U

- UNIX
 - using different databases with same application server 48

