IBM® DB2® for Linux®, UNIX®, and Windows®

Best practices
Deploying IBM DB2 products

Andrew Hilden
Integrated Systems Management Archiitect

Darin McBride
DB2 Installation and Up & Running Developer

Jason Shayer
DB2 Information Development

Executive summary

Deploying IBM DB2 products, including IBM DB2 and DB2 Connect™, on Linux, UNIX and Windows platforms for dozens, hundreds, or thousands of clients and servers might seem like a daunting task. However, DB2 9.7 has features and tools to help you deploy servers and clients quickly and seamlessly across your enterprise.

You can deploy DB2 9.7 using deployment technologies such as Microsoft® Systems Management Server (SMS), or you can use an easy and effective manual deployment method. The type of deployment best suited for your organization depends on your distribution and maintenance requirements, as well as on the hardware, software, and infrastructure of your existing computing environment.

You can run the DB2 installer as a GUI installer which provides an interactive type of installation or as a response file where input is passed to the installer through a customized file. The latter method, which is referred to as an unattended or a response file installation, does not display a GUI and does not require any user input during the installation.

An unattended installation, which you can use to deploy software to a large number of users, provides more functionality and flexibility than a GUI installation. Using the unattended installation method, you can:

- Create more than one instance
- Catalog and create databases
- Set database manager configuration parameters and profile registry variables
- Import instance profiles generated from the **db2cfexp** command (the connectivity configuration export tool)
- Embed DB2 software within applications
- Use the same response file to roll out identical copies of your DB2 products across multiple computers using the same components, registry variable settings, and instance configuration settings.

The easiest way to deploy your DB2 product is to perform multiple unattended installations using a response file. The unattended installation eliminates the need for user input at the target computers.

As an up-front preparation cost, the customization of the response file (specifying components, registry variable settings, and instance configuration settings) is done once and the actual deployment can then typically be performed quickly for hundreds or thousands of computers. Additional computers may be added to the deployment later with almost no additional preparation required.

Introduction

This document describes the best practices for deploying the DB2 9.7 family of products on Linux, UNIX, and Windows platforms across multiple computers. There are many DB2 9.7 products, so choosing the right one might be challenging if you are not familiar with the DB2 product line.

Although the steps to install and deploy DB2 9.7 differ by platform, the overall steps are:

1. Plan your deployment
2. Create a location for the installation image
3. Create a response file
4. Create a deployment script or definition
5. Run the deployment script or definition on all computers

For simplicity, examples for Linux and UNIX platforms will use Bourne shell syntax. If you use other shells, you will need to adapt the instructions as appropriate.

For Windows platforms, this paper also describes how to deploy software using the Microsoft® Systems Management Server (SMS), but the general concepts in this paper apply to most software deployment tools.

You can use these instructions to install clients, servers, parallel servers, and fix packs.

Planning your deployment

What should I install?

To help you determine which DB2 products to install on the computers in your environment, read the following product overviews. These overviews are not intended to describe the licensing terms of each product, but rather to summarize the features and usage scenarios for each product.

You can get more details on what products might be right for you at: http://www.ibm.com/developerworks/db2/library/techarticle/dm-0611zikopoulos/.

DB2 server products

Several DB2 server products are available depending on your applications, users' functions, and workload characteristics.

All DB2 servers are licensed products and some features require additional feature licenses. More information about licenses is available in the "Applying DB2 license files to images you will distribute" section.

DB2 Personal Edition is designed for an individual's use and is a good fit when users need a single user at a time data store (and there could be many within an organization). It includes SQL replication, so that users can synchronize their data with the enterprise. It does not permit connections from remote clients.

DB2 Express-C Edition is designed for small businesses and multi-branch companies, as well developers and business partners who serve these clients. DB2 Express-C can be setup quickly, is easy-to-use, and includes self-managing capabilities. It also embodies all of the core features of the more scalable DB2 editions, including the revolutionary pureXML technology for powering a new breed of Web 2.0 and SOA based solutions.

DB2 Express Edition is designed for small and medium businesses with a small number of clients. It has all of the functionality of DB2 Personal Edition, but also allows remote connections.

DB2 Workgroup Server Edition is designed for larger-scale servers. It has all of the functionality of the DB2 Express Edition, but with more extensive licensing.

DB2 Connect Enterprise Edition is designed to connect LAN-based systems and desktop applications to your mainframe and minicomputer host databases, such as DB2 for z/OS. This connectivity solution can consolidate your host access through a gateway and deploy web and multi-tier applications.

DB2 Enterprise Server Edition is designed for large and very large databases. It is functionally equivalent to the combination of DB2 Workgroup Edition and DB2 Connect Enterprise Edition. With Database Partitioning Feature (DPF), Enterprise Server Edition also enables multiple computers to work together as a single shared-nothing cluster, simulating a single database server. The DPF feature is available as part of InfoSphere Warehouse 9.7, but deployed as part of a DB2 Enterprise Server Edition installation.

DB2 Advanced Enterprise Server Edition is based on DB2 Enterprise Server Edition and is an easy and cost-effective way for organizations to assemble a top-flight team of data management software. It bundles IBM DB2 database software with a comprehensive range of optimization, development, and management tools.

DB2 drivers

Depending on the architecture of your application, you should install a DB2 client or driver on each computer that your application is installed on. This setup enables your application to communicate with the DB2 server while the application is running.

Understanding the different clients and drivers that are available and selecting the appropriate one is critical for configuring DB2 connectivity and reducing the footprint of the DB2 files. Because each server has the full functionality of all clients, including the GUI tools and application development tools, there is no reason to install a client on a computer that already has a server.

You can install:
- The **IBM Data Server Driver Package**
- The **IBM Data Server Runtime Client**
- The **IBM Data Server Client**
- The **IBM Data Server Driver for ODBC, CLI, and .NET**
- The **IBM Data Server Driver for JDBC and SQLJ**

As a best practice, you should use the IBM Data Server Driver Package. IBM Data Server Driver Package is a lightweight deployment solution that provides runtime support for applications using ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ without the need of installing Data Server Runtime Client or Data Server Client. This driver has a small footprint and is designed to be redistributed by independent software vendors (ISVs), and to be used for application distribution in mass deployment scenarios typical of large enterprises.

The IBM Data Server Driver Package capabilities include:

- The DB2 Command Line Processor Plus (CLPPlus) for dynamically creating, editing, and running SQL statements and scripts.
- Support for applications that use ODBC, CLI, PHP, or Ruby to access databases.
- Support for client applications and applets that are written in Java using JDBC, and for embedded SQL for Java (SQLJ).
- IBM Informix® support.
- Support for running embedded SQL applications. No precompiler or bind capabilities are provided.
- Application header files to rebuild the PHP, Ruby, Python, and Perl drivers. The Python and Perl drivers are not available in IBM Data Server Driver Package; however, you can download and build these drivers using the header files.

- Support for DB2 Interactive Call Level Interface (db2cli).
- Support for DRDA® traces (db2drdat).
- On Windows operating systems, IBM Data Server Driver Package also provides support for applications that use .NET or OLE DB to access databases. In addition, the IBM Data Server Driver Package is available as an installable image, and merge modules are available to allow you to easily embed the driver in a Windows Installer-based installation.

Redistributing clients and drivers

You can freely redistribute all clients and drivers, except the IBM Data Server Client, with your application. You can download these files from the DB2 Application Development web site (www.ibm.com/software/data/db2/ad) at no charge. Read the license files to understand the terms for use and redistribution.

Installation methods

You can install DB2 Data Server products non-interactively in a number of ways. Which method you choose depends on your preference, platform, and in some cases, product. You can use all methods interchangeably. You can use one installation method now and then later use a different installation method on the same computer. Choosing one method does not commit you to the continued use of that method.

Prior to any deployment on production computers, test your installation on a test computer. Also, familiarize yourself with the GUI installer prior to designing a non-interactive DB2 deployment. The GUI installer can help you understand terms and keywords that are present in the non-interactive installation modes.

The following sections describe the various ways to deploy DB2 products.

Unattended installation using a response file

On Windows platforms, use a response file for all mass deployments using an installation. On Linux and UNIX platforms, you should also use a response file.

This method involves creating a response file and then running the **db2setup** or **setup** command using that response file. Refer to the “Creating a response file” section for more information about response files. This method offers many benefits:

- Syntax checking of the response file up front means that if **db2setup** accepts the response file, the installation is more likely to work.
- It enables you to select what to install with the same or better granularity than the GUI installer does.
- It provides you with more detailed configuration options than the GUI installer does.
- You can set up database manager configuration (dbm cfg) and DB2 profile registry parameters during the installation.
- All output and log files are fully translated.

Unattended installation using the db2_install script (Linux and UNIX)

The **db2_install** script is a thin, coarse wrapper that calls the response file installer to install the entire product. Compared to a response file installation, this method has the following limitations:

- The script performs a coarse installation, meaning that all optional components are installed; you cannot select specific components to install.

- You can input a minimal set of values into the script. To run the script without prompting, you must specify the –b and -p parameters which will input the installation path and the product name respectively. For example, **db2_install -b /opt/IBM/db2/V9.7 -p ESE**

- The script does not create or configure any instances or configure the DB2 Administration Server (DAS). To create and configure instances or configure the DAS, you should use the **db2icrt** or **dascrt** command. Alternatively, to create instances in the unattended mode of installation, use the **db2setup** command instead of the **db2_install** command.

However, the advantage of the **db2_install** method is that it uses a default, built-in response file, so you do not have to set up a response files.

Manual installation (Linux and UNIX)

This method involves uncompressing and untarring the DB2 payload directly.

Due to the advanced nature of this installation method, do not do this unless you have very advanced needs. This paper does not address this method of installation. If you feel comfortable with this installation method, you can extrapolate from the other installation methods to this one.

Reducing the size of your installation image

You can use the **db2iprune** utility to reduce the size of most installation images. The IBM Data Server Driver is already reduced in size and this utility is not available for that image. This utility will remove any features or languages from the install image that you do not require. The **db2iprune** utility reduces the size of the installation images and might reduce the amount of network traffic as you roll out the images to computers. More information is available in the "Pruning a DB2 product installation image using the db2iprune utility" section.

Deploying using software distribution software (Windows)

Several products are available to help you to deploy software to a large enterprise. In this paper, we cover deploying software using SMS on Windows platforms. You can apply these concepts to other technologies to achieve a similar result. See the "Creating a deployment script or package" or "Running the deployment script or definition on all computers" section for details on deploying software using SMS.

Applying DB2 license files to images that you distribute

To use DB2 server products for some IBM DB2 Data Server Client features, you must obtain the appropriate licenses for the server products for each computer you will deploy to. You might also have to obtain a license for a particular feature that you wish to use.

For more information about determining which features you need to purchase, contact your IBM Sales Representative or refer to the DB2 Information Center.

Typically, you can apply the appropriate license file after the installation has completed, but doing this on all computers that have been distributed to across your enterprise is inefficient. To solve this problem, you can automatically install licenses that you have purchased.

To enable this feature you need to acquire the necessary licenses for the product and features that you purchased. The licenses are typically provided as a separate download when you purchase your product and features and are accessible through Passport Advantage.

Although you must purchase a license for each computer that you install on, the key that is distributed to each computer is identical. Copy the license files to the `db2/license` directory of the image that you will deploy to your enterprise.

When the image is installed, the licenses are automatically installed. To verify that a license has been installed, you can run the **db2licm -l** command and review the output. You can periodically check your license compliance by running the **db2licm -g** command to generate a license compliance report.

Alternatively, you can apply the license after installation by running the **db2licm –a** command.

Prerequisite: **T**o use the DB2 Database Partitioning Feature (DPF) in DB2 9.7, you must purchase the InfoSphere Warehouse 9.7 product. You much include the license for that product on your installation media or apply the license post installation as detailed previously.

Fix pack installations

On Linux and UNIX platforms, you can use the **installFixPack** command to update a DB2 data server product. This script is non-interactive and is available for large-scale deployments.

On Windows platforms you can apply a fix pack using a response file installation in the same way that you installed the original product.

You can also install a fix pack in a new location, run the **db2iupdt** command against each instance, and then uninstall the old copy. Use this method if you want to test a new level of the DB2 product on a system before moving your entire production system to the new fix pack level.

Frequently, the first deployment of DB2 products requires the installation of a particular fix pack level instead of the initially released level of the DB2 product. Instead of performing a two-step deployment (installation of the initially released level of the DB2 product and application of the required fix pack), you can perform one installation using the fix pack image, which is a full DB2 product image.

Validating your installation

With DB2 9.7, you can use the **db2val** command to validate your installation and core functionality of your DB2 copy. You can run this command anytime after the product installation or fix pack update to ensure that the basic DB2 functions are still intact. More information is available in the "Validating your DB2 copy" section.

Creating a location for the installation image

Physically moving a DVD from system to system is not a good use of any administrator's time. Physically mounting the DVD on a single computer and then using a network file system such as Samba, NFS, DFS, or AFS slows down all deployments as the network responds to requests.

Instead, use a code server to speed up deployment. By copying the image to a hard disk that is network accessible, you can take advantage of the improved seek times and improved buffers that most file system drivers provide.

Linux and UNIX platforms

To copy the image from a DVD, issue the following command as root:

```
# (cd /dvdrom; tar cf - *) | (cd /bigshareddisk; tar xf -)
```

where:

- *dvdrom* represents the mount point of the DB2 product DVD
- *bigshareddisk* represents the directory from which you want to share the DB2 product image throughout your network.

The primary purpose of using the tar command is that not all platforms have cp commands that can preserve symbolic links (or symlinks). Alternatively, any equivalent command that can preserve symlinks, ownerships, and permissions will work as well.

To uncompress an image downloaded from Passport Advantage, issue the following command:

```
# (cd /downloads; gunzip -c *.tar.gz) | (cd /bigshareddisk; tar xf -)
```

where:

- *downloads* represents the directory into which you downloaded the DB2 product image
- *bigshareddisk* represents the directory from which you want to share the DB2 product image throughout your network.

Alternately, on Linux platforms, you can do this by using the **z** option:

```
# cd /bigshareddisk; tar xzf /downloads/*.tar.gz
```

Do not use a Microsoft® Windows® file server for this purpose. Windows file servers lose file permissions and are unable to handle symbolic links.

The next step is to export the */bigshareddisk* directory through a network file system. The details of this step are outside the scope of this paper. Although you cannot install the DB2 product on an NFS partition, you can install it from an image on a NFS file system.

Windows platforms

Put the image on a file server that is accessible to all computers on which you will perform the installation. This method ensures that if the product must repair itself, it can automatically find the installation image. Use a different location for each DB2 version that you deploy.

To copy the image from a DVD, issue the following command:

```
xcopy /e /q dvdrom:\ \\Server\path
```

Ensure that the path you specify is accessible in read-only mode by all computers that you want to distribute the software to. Use UNC paths when referring to this image because it requires the mounting of the drive on a local computer.

Creating a response file

A response file tells the installation process what to install and how to configure it. On Windows platforms, you must use a response file. This process can simplify the installation script considerably by reducing the number of commands that you must run after the installation.

Depending on what environment you are targeting, the response file can be significantly different. This paper focuses on three environments:

- Clients
- Single-computer servers
- Multiple computer servers that use DPF

For computers that are being deployed as a combination of these, for example, a database server that has the Information Integrator Relational Wrapper for DB2 Data Sources installed and thus must also act as a client to another database, these environments may be combined. However, for the sake of simplicity, these combinations are not discussed in this paper.

Creating response files using the DB2 Setup wizard

The easiest way to create a response file is using the DB2 Setup wizard. You can use this GUI wizard to go through the installation and then save the response file. Use this method to easily select which components you want to install and what configuration options you want. You can also tailor the response file by copying additional entries from the sample response files available on the DB2 product media.

Using the DB2 response file generator

After installing your DB2 product, you can use the DB2 response file generator utility, **db2rspgn, to take** a snapshot of your product installation and configuration settings. This utility saves this information in a response file and in a configuration profile. This utility is not available for the Data Server Driver.

Use this method if you plan to configure your product after installation, such as setting configuration parameters or cataloging remote nodes and databases. All of this information will be saved in the output response file, eliminating the need to manually add it yourself.

The **db2rspgn** utility automatically creates an instance configuration profile for each instance on the system or specified instances. The resulting configuration profile is saved to the same directory as the generated response file.

The INS file is the configuration profile file that contains configuration and connectivity settings for the instance. By default, the generated response file is configured to import all INS files at installation time by setting the **DB2.CLIENT_IMPORT_PROFILE** keyword in the generated response file to [INSTNAME_NAME].INS.

To use the **db2rspgn** utility from the CLP, use the following syntax:

```
db2rspgn -d destination_directory -i instance
```

where:

- *destination_directory* represents the destination directory for the generated response file and any instance configuration files.

- *instance* is an optional parameter that represents the instances for which you want to create a profile. If not specified, a configuration profile will be generated for every instance in the DB2 copy. You can specify this parameter multiple times to input more than one instance.

For example, to create a response file and profile in the C:\myfiles directory, use the following command:

```
db2rspgn -d C:\myfiles
```

The C:\myfiles directory contains a response file and an INS file for each instance associated with this DB2 copy.

Manually creating a response file using the sample response files

You can also create a response file by copying one of the sample response files from the DVD and modifying it. Samples are typically available in the db2/platform/samples directory, where platform is the platform identifier.

For example, db2/linux/samples/db2client.rsp is the IBM Data Server Client sample response file for 32-bit Linux on the DB2 for 32-bit Linux DVD.

Modifying a response file

You can use the information in this section to modify generated response files. A response file can be easy to use, although you should test it prior to deploying it across a large enterprise.

Any portion of a line that begins with the * or # character is ignored as a comment:

```
* This is a comment.
```

Any portion of a line beginning with a ** or ## sequence is ignored as a comment to the end of that line.

```
PROD = ENTERPRISE_SERVER_EDITION ** This is a comment.
```

All data in a response file is presented in the format KEYWORD=VALUE. In the above example, the keyword is **PROD**, and the value given is ENTERPRISE_SERVER_EDITION.

Each keyword is considered to be a token and is left untranslated. Most values are also tokens. For example, in the following line both the **INSTALL_TYPE** keyword and the TYPICAL value are tokens and should not be considered to be English words:

```
INSTALL_TYPE = TYPICAL
```

The response file installer recognizes these tokens, and the tokens that it recognizes are the same, regardless of language.

Entries in the sample response file that are not commented out are considered mandatory and must be filled out for an installation to proceed.

Many keywords map directly to questions that you might encounter during a GUI installation, so a passing familiarity with the GUI installation can help in understanding the response file.

The **PROD** keyword specifies the product to be installed. The value of this keyword is preset in each sample response. There is no need to change this.

This **INSTALL_TYPE** keyword specifies the type of installation: COMPACT, TYPICAL, or CUSTOM.

If you set the **INSTALL_TYPE keyword to** CUSTOM, you can select extra components by removing the comment characters from the names of those components. For comparison, response files separate out components to show which ones would be selected during a typical installation. If the installation type is TYPICAL or COMPACT, the **COMP** keyword is ignored.

The **LANG** keyword specifies the language. If you want a language other than English, specify it using this keyword. Unlike the GUI installer, the **LANG** keyword is set to ENGLISH.

If you set the **SELECT_INSTALLED_LANGS** keyword to YES, the installer will automatically select all languages that DB2 is already installed in. This keyword is useful when you install a second product in the same installation path and want to keep the old language setting for the new product.

The **LIC_AGREEMENT** keyword tells the installer that you have read and agreed to the licensing terms that are included on the DB2 media. By default, the value of this keyword is DECLINE. You must change this to ACCEPT for the installation to proceed. You can find the licenses in the `db2/license/locale.encoding` directory.

Instance creation settings

These settings are available to create an instance as part of the installation. You might create more than one instance in a single installation by using multiple INSTANCE names.

For example, you can specify INSTANCE=instance1 and INSTANCE=instance2. For each instance, you would create the rest of the settings that you want with instance1 and instance2 prefixes.

Most of these settings are either database manager configuration settings, or DB2 profile registry settings. All such configurable settings are available in the response file.

Other settings allow you to create new users, including passwords. If you choose to create new users with the response file, it is recommended that you use proper file system security to ensure that only root on each computer that you install to can read the response file because the password is in plain text. Alternatively, you may want to create the user account, with password, prior to using a response file, or you can generate it using the GUI, in which case it will be encrypted.

Instances are used to store configuration for DB2. This configuration information includes local databases for servers and remote databases for both clients and servers.

You do not need to create a new instance if you are upgrading an instance from a previous version, or updating from another copy. DB2 needs an instance to do anything, whether it's hosting a local database or connecting to a remote database.

You can use the **CLIENT_IMPORT_PROFILE** parameter to import a configuration profile for an instance. To use this feature, install a DB2 product on a test computer and take a snapshot of the configuration information using the **db2cfexp** utility. Refer to the section called "Creating a DB2 configuration profile" in this paper for more information about how to create a configuration profile.

DAS settings

You must have a DAS running if you want to use tools such as the Configuration Assistant, the Control Center, or the Development Center. Using the sample response file, you can create a DAS using the applicable section under products that must have a DAS.

You need to create a DAS only if you are installing DB2 9.7 for the first time. For example, if you have a DB2 9.1 DAS and you want to install DB2 9.7 for the first time, you do not need to create another DAS. In this case, you can comment out the DAS settings. Only one DAS is supported on each machine.

The user associated with the DAS must not be the same user as is used for any instances.

Client deployment considerations

Exporting your client configuration from a working client can ease the work required to catalog nodes and databases from the deployed clients.

Single-server deployment considerations

You cannot use a response file installations to create local databases. You must create these databases using a deployment script.

Parallel-server deployment considerations (Linux and UNIX)

In a parallel environment, you have to create a sever instance on only one computer. Typically, you create the server instance on a local disk of the first computer that you install the product on. This computer is referred to as the instance owning computer. The rest of the computers in the cluster, which do not require instances, are referred to as nodes.

You must perform two types of installations. The instance-owning computer needs a response file that installs DB2 Enterprise Server Edition and creates the local instance. Each node needs a response file that installs DB2 Enterprise Server Edition without creating an instance. This second response file is easy to create: remove any part of the response file related to instances, or keep only the **PRODUCT, INSTALL TYPE, COMP, LANG, SELECT INSTALLED LANGS,** and **LIC AGREEMENT** keywords.

Another way to create these response files is to install the instance-owning node using the GUI installer. An option during the GUI installation is to save a response file for use with all nodes. For example, with DB2 Enterprise Server Edition, the default response files are `PROD_ESE.RSP` and `PROD_ESE_addpart.RSP`. If you are deploying a single cluster, using this approach is often simpler.

Response file installations in parallel environments cannot set up the .rhosts file for rsh access between nodes or set up the db2nodes.cfg file. Your deployment script must include these steps.

Creating a DB2 configuration profile (optional)

The DB2 configuration profile specifies instance configuration settings and node and database catalog information found in the system node and database directories of your installed DB2 product.

The **db2cfexp** utility takes a snapshot of the instance profile and connectivity information to create the configuration profile. This file enables you to distribute a DB2 data server product with identical configuration information across multiple computers. The configuration profile contains database information (including DCS and ODBC information), node information, protocol information, database manager configuration settings, registry settings, and common ODBC/CLI settings.

If you use the **db2rspgn** utility to create a response file, you do not need to do any additional work to create the configuration profile because the **db2rspgn** utility automatically creates it for you. If you used the DB2 Setup wizard or a sample response file to generate your response file, you might want to create a configuration profile using the **db2cfexp** utility.

Open up the DB2 CLP and issue the **db2cfexp** command to create a configuration profile:

```
db2cfexp filename [ template | backup | maintain ]
```

where:

- *filename* represents the path or filename of the configuration profile
- **template** creates a configuration profile that will be used as a template for other instances of the same instance type
- **backup** creates a configuration profile for backup purposes
- **maintain** creates a configuration profile that contains only database and node information for maintaining other instances

For example, the following command creates a configuration profile called db2ese_config in the C:\myfiles directory:

```
db2cfexp C:\myfiles\db2ese_config
```

After you create the configuration profile, specify its path name in the DB2 response file by setting the **DB2.CLIENT_IMPORT_PROFILE** keyword.

Creating a deployment package or script

To deploy DB2 products to another computer, you must create a package using a software deployment tool or creating a script that you can run on the computer. This section describes how to create a package using SMS and how to create a deployment script on Linux and UNIX platforms

Packaging IBM Data Server products using SMS 2003

Prerequisites

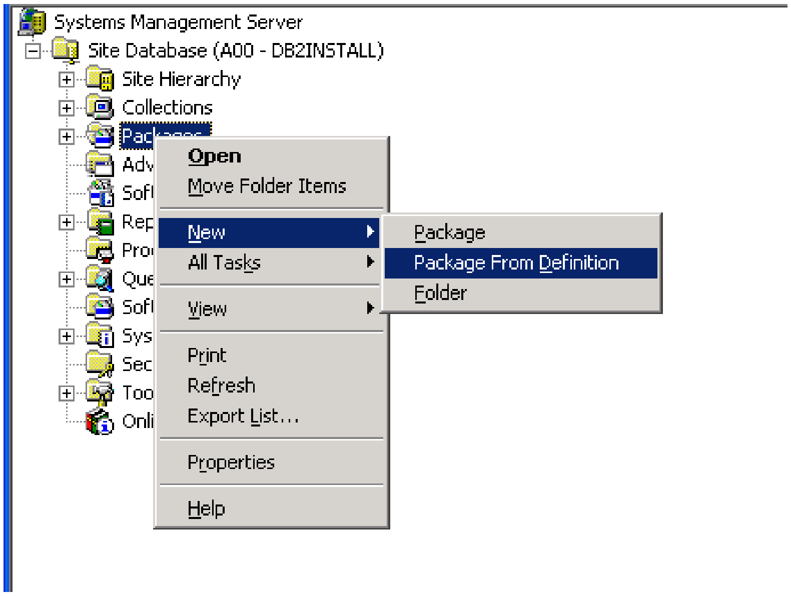
You must have SMS 2003 installed and configured on your network for both your SMS 2003 server and your SMS 2003 client workstations. For details on the following areas, see *the Systems Management Server Concepts, Planning, and Deployment Guide* and *Systems Management Server Operations Guide* at http://www.microsoft.com/technet/downloads/sms.mspx:

- SMS 2003 platforms and prerequisites
- SMS 2003 setup (including the setup of the primary and the secondary sites)
- Addition of clients to the SMS 2003 system

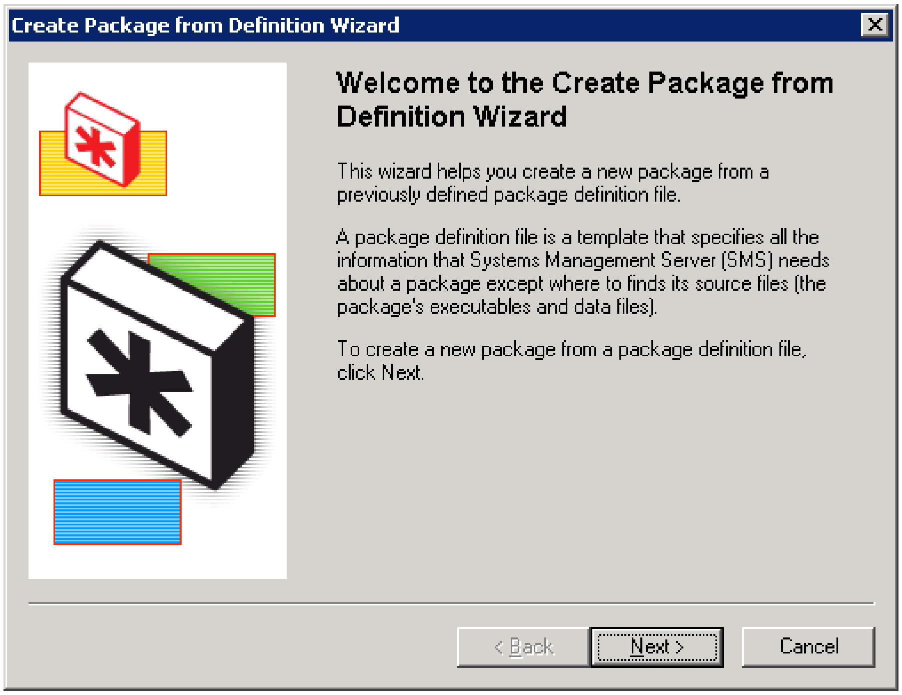
Procedure:

To package a DB2 product using SMS:

1. Copy the product installation image to a location where you can edit its contents.
2. Create a DB2 response file. Refer to the section "Create a response file" in this paper for more details.
3. On an SMS 2003 distribution point server, from the **Start Menu select Programs > Systems Management Server > SMS Administrator Console.**
4. On the SMS Administrator Console, open the Site Database object tree, and right-click Packages. Select **New > Package From Definition**.



5. On the Welcome to the Create Package from Definition Wizard, click **Next.**



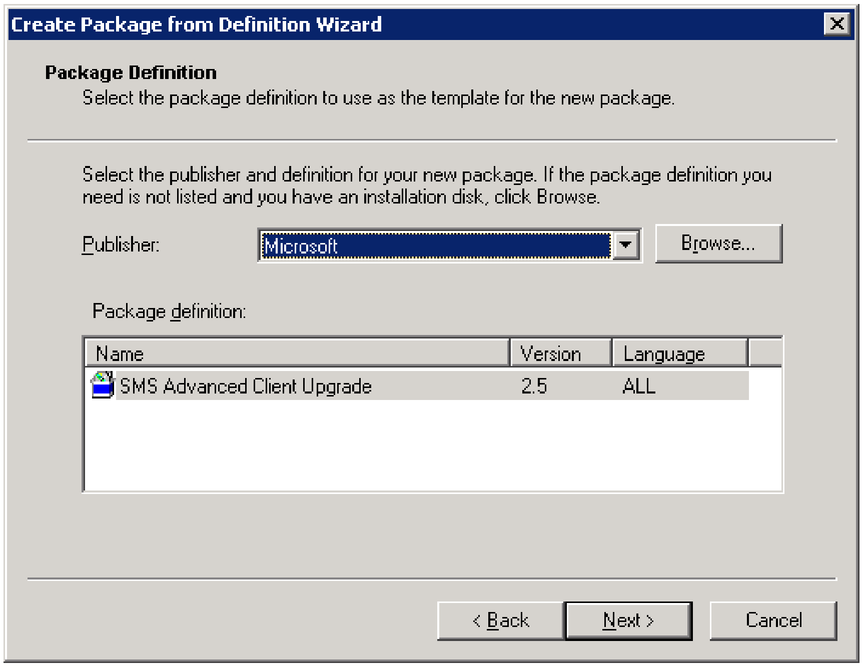
Welcome to the Create Package from Definition Wizard

This wizard helps you create a new package from a previously defined package definition file.

A package definition file is a template that specifies all the information that Systems Management Server (SMS) needs about a package except where to finds its source files (the package's executables and data files).

To create a new package from a package definition file, click Next.

6. On the Package Definition page, to search for a DB2 package definition file, click **Browse.**



Create Package from Definition Wizard

Package Definition

Select the package definition to use as the template for the new package.

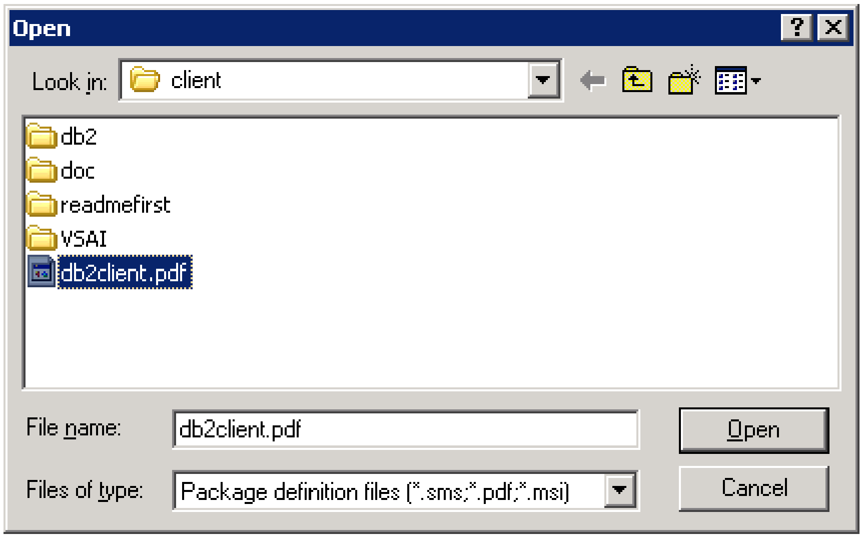
Select the publisher and definition for your new package. If the package definition you need is not listed and you have an installation disk, click Browse.

Publisher: Microsoft

Package definition:

Name	Version	Language
SMS Advanced Client Upgrade	2.5	ALL

7. From the list shown in the Browse window, select the db2*.pdf file from the list shown in the Browse window. By default this file is located in the `db2\Windows\samples\` directory of the installation image. Click **Open.**



Open

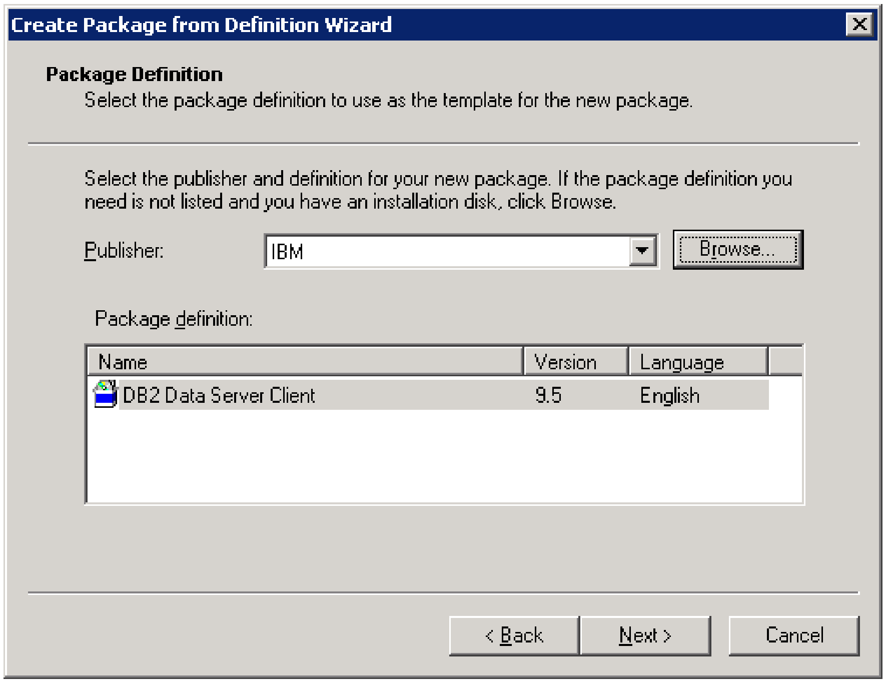
Look in: client

- db2
- doc
- readmefirst
- VSAI
- db2client.pdf

File name: db2client.pdf

Files of type: Package definition files (*.sms;*.pdf;*.msi)

8. On the Package Definition page, click **Next**.



Create Package from Definition Wizard

Package Definition

Select the package definition to use as the template for the new package.

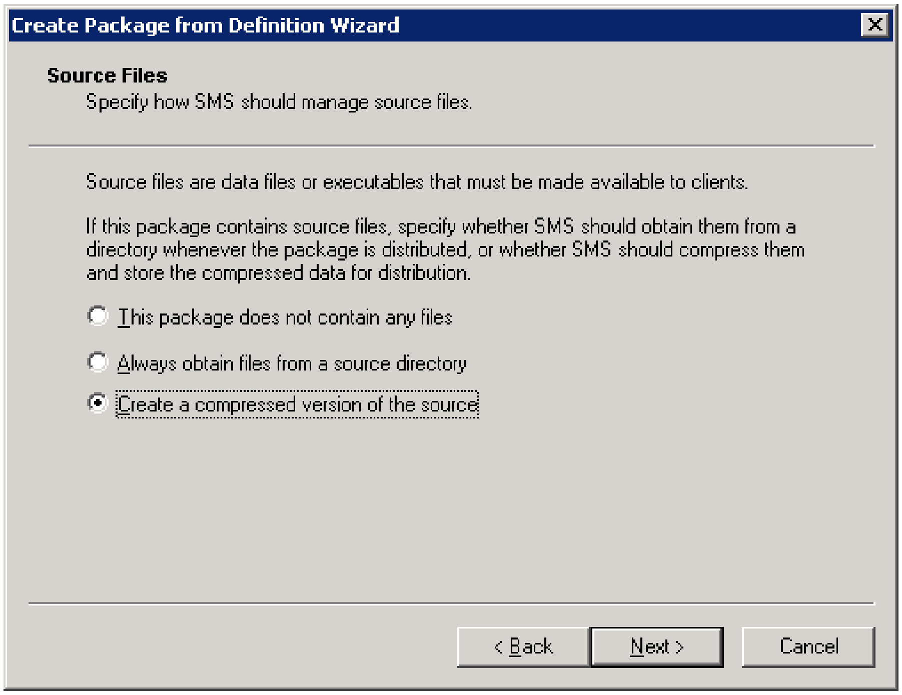
Select the publisher and definition for your new package. If the package definition you need is not listed and you have an installation disk, click Browse.

Publisher: IBM

Package definition:

Name	Version	Language
DB2 Data Server Client	9.5	English

9. On the Source Files page, click **Create a compressed version of the source** and click **Next.**



Create Package from Definition Wizard

Source Files

Specify how SMS should manage source files.

Source files are data files or executables that must be made available to clients.

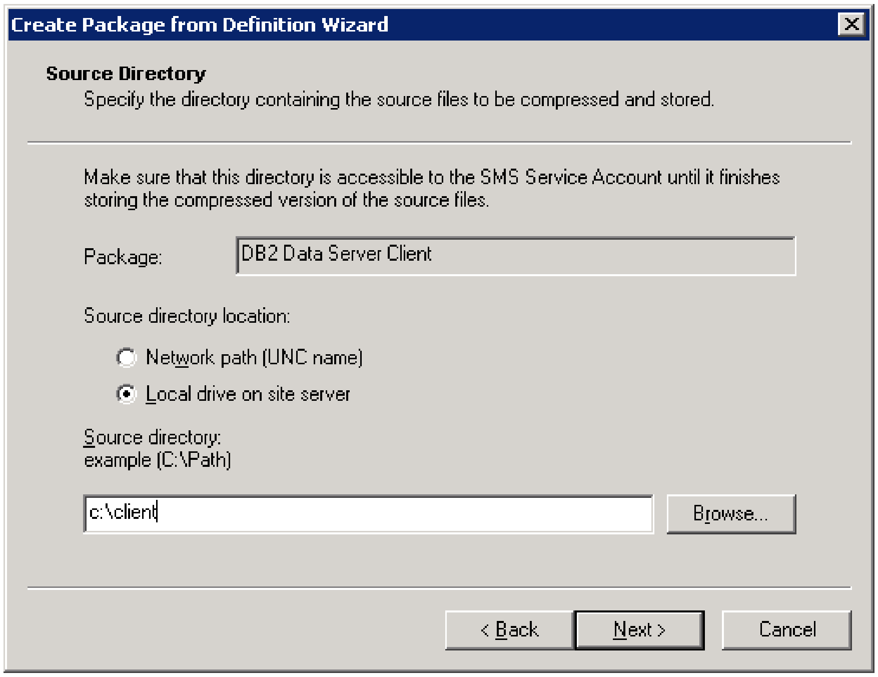
If this package contains source files, specify whether SMS should obtain them from a directory whenever the package is distributed, or whether SMS should compress them and store the compressed data for distribution.

○ This package does not contain any files

○ Always obtain files from a source directory

● Create a compressed version of the source

10. On the Source Directory page, click **Network path (UNC name)** to access the source files from a remote network drive or click **Local drive on the site server** to access the source files from a local drive. In the **Source directory** field, type the name of the directory where the DB2 product installation image is located. This is the directory containing the setup.exe file. Click **Next.**



Create Package from Definition Wizard

Source Directory

Specify the directory containing the source files to be compressed and stored.

Make sure that this directory is accessible to the SMS Service Account until it finishes storing the compressed version of the source files.

Package: DB2 Data Server Client

Source directory location:

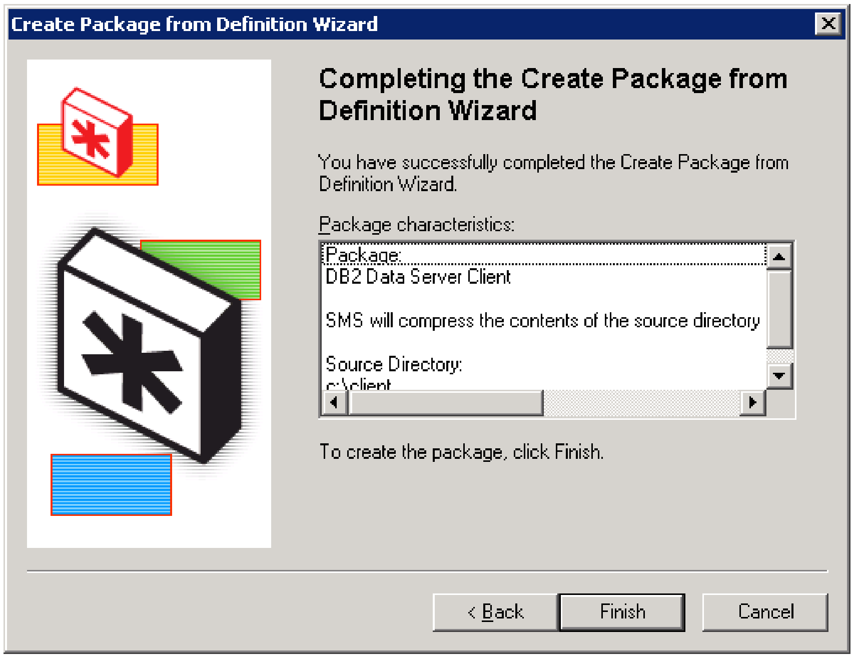
○ Network path (UNC name)

● Local drive on site server

Source directory:
example (C:\Path)

c:\client

11. On the Completing the Create Package from Definition Wizard page, click **Finish**.



Create Package from Definition Wizard

Completing the Create Package from Definition Wizard

You have successfully completed the Create Package from Definition Wizard.

Package characteristics:

Package:
DB2 Data Server Client

SMS will compress the contents of the source directory

Source Directory:
c:\client

To create the package, click Finish.

Creating a deployment script

A deployment script is usually written as a shell script, but you can use any language including Perl, Java™, or C or you can use batch files. You can think of this script as a wrapper to handle aspects of DB2 installation and configuration that the DB2 installation procedure does not handle. You run the script on the target computer as part of the deployment process.

Creating a deployment script might be simple if you created a complete response file, or it might be complex, especially if you are not using a response file to perform the installation.

When reading the examples, assume that /share is the remote file system, which is mounted as /share on all computers that need to be deployed. If no shared file system is available, you can remotely mount a file system or to write a script to automatically mount and unmount a file system. How to mount a file system is outside the scope of this paper.

On Linux and UNIX platforms, the file system can be NFS, AFS, DFS, or any number of non-local file systems. However, do not use a Samba file system for reasons that are given in the “Creating a location where the installation image will be located” section. On Windows platforms, a Samba file system or any network-accessible drive is supported.

The IBM Data Server Client is used as the example product, but the steps are the same for all products packaged with an installation image.

Deploying a response file installation

Deploying a response file installation is the simplest case. Because the product definition, the selection of components, and the setting up of instances are almost completely encapsulated by a response file, you perform the bulk of the work by running that response file:

On Linux and UNIX platforms:

```
/share/client/db2setup -u /share/client/v95GA/db2client.rsp
```

On Windows platforms:

```
\\server\share\setup -u \\server\share\client\db2client.rsp
```

In these examples, the customized response file is in the /share/client or \share\client directory with the product.

Post-installation tasks

To use the DB2 Data Server Client, you typically must catalog remote servers and databases. However, you can catalog these connections automatically using the **db2cfimp** command and then use the **db2cfexp** command to export these connections, or you can catalog these connections manually.

For example, on Linux and UNIX platforms, you might have generated a configuration file for an instance, db2inst1, by using the **db2cfexp** utility. You can use the **db2cfimp** utility to replicate that configuration as follows:

```
su – db2inst1 -c ". sqllib/db2profile;

                  db2cfimp /share/rtcl/v9GA/db2inst.exp"
```

To manually catalog with the same created instance:

```
su – db2inst1 -c ". sqllib/db2profile;

                  db2 catalog tcpip node ..."
```

In the latter case, it is better to put the catalog commands in another file and then allow the CLP to read that file:

```
su – db2inst1 -c ". sqllib/db2profile;

                  db2 -tvf /share/rtcl/v9r7GA/catalog.db2"
```

You can then use this other file to perform other DB2 setup task, as shown in the following examples:

```
catalog tcpip node server remote serverhostname;

catalog database sample at node server;
```

On Windows platforms, the commands are similar. However, to set the DB2 instance that you want to use, use the **DB2INSTANCE** environment variable and issue the **db2cmd** /C command to run DB2 commands:

```
set DB2INSTANCE=DB2

“C:\Program Files\IBM\SQLLIB\BIN\DB2CMD.EXE“ /C “DB2 CATALOG
DATABASE sample AT NODE server”
```

Server considerations

In a server deployment, you typically require extra commands to create and set up databases, table spaces, tables, for example. As with the client, creating a single script and using the **db2 –tvf** command to run it can simplify your deployment.

In a parallel environment on UNIX platforms, the deployment script must be able to update the .rhosts and db2nodes.cfg files as required.

Deploying an installation using db2_install (Linux and UNIX)

You can use the **db2_install** command to deploy an installation. This method bypasses the response file creation step and all of the conveniences of the response file installation.

The installation phase is still relatively simple:

```
/share/client/v95GA/db2_install -p CLIENT -n -b /install/path
```

The product identifier is unique to each product. To see a list of the available product identifies, run the **db2_install** command without any parameters. To specify additional languages, use the **-L** option. For more information about the **db2_install** command, see the DB2 Information Center.

You can use the deployment script to create users, instances, and the DAS. Creating users is a platform-specific function, so consult the operating system manuals for information. Instance and DAS creation is described in the DB2 Information Center.

For further post-installation tasks, see the "Deploying a response file installation" section.

Deploying a fix pack

Upgrading existing installations of DB2 products is just as important, if not more important, as performing the initial installation. Fortunately, upgrading is generally much easier to do.

You can download fix packs from
http://www.ibm.com/software/data/db2/udb/support.html.

Always read the fix pack readme and the list of fixed APARs. You can find this document in the same directory as the fix pack on the FTP site or as a link from the download web page. The readme provides information on how to install the fix pack and specific information about any known problems or workarounds. Consult the readme for any changes or additions that might occur as later fix packs are released.

Linux and UNIX platforms

Rolling out an in-place upgrade of your clients can be as simple as performing the following steps:

1. Download the new client.

2. Put the client image in the `/share` directory, for example, in the following subdirectory:

   ```
   /share/rtcl/v9r7FP5.
   ```

3. Add the actions that the fix pack requires to your deployment script, as shown in the following example:

```
# code to ensure your applications are stopped

/share/rtcl/v9r7FP5/installFixPack -b /install/path

/install/path/instance/db2iupdt -e

# any post-fixpack steps you might require
```

Beginning with DB2 9.5, the **db2iupdt command** step is done automatically by the **installFixPack command**. If you are applying the fix pack to a server, you will likely have to stop all instances first, and restart them afterward. In a parallel environment, you must do this only on the instance-owning node.

If you have multiple products installed, for example WebSphere Information Integrator Relational Wrappers installed on DB2 Enterprise Server Edition, use the Universal fix pack instead of fix packs for the individual products.

The new method of upgrading involves installing to a new location, testing there, and then updating instances to the new copy. This deployment can be rolled out in stages where required and is much more important on servers where downtime and unknowns kept to a minimum.

In this scenario, you should keep your old response files from the initial installation as they can serve as the basis for the new installation. Remove any portion dealing with instances or the DAS, and specify a new installation path through the FILE keyword. Deploy this installation as previously described, and then use the **db2iupdt** command to move your instances to the new copy.

This method allows you to deploy the code without taking down production applications. You must take down only those applications such as the DB2 server, only for as long as it takes to run the **db2iupdt** command.

Windows platforms

The first step is to obtain the fix packs that you need. You must obtain a fix pack for any product that you have deployed to your computers. This is an important and mandatory action if you have more than one DB2 product installed on a particular computer.

If you have multiple products installed, for example WebSphere Information Integrator Relational Wrappers installed on DB2 Enterprise Server Edition, use the Universal fix pack instead of fix packs for the individual products. The Run-Time Client Lite and DSDriver are not included as part of the Universal fix pack as they cannot co-exist with any other DB2 products in the same DB2 copy.

All of the fix packs that are installed on the computer must be available and uncompressed at the time of the fix pack installation because all of the DB2 products in the copy being updated must remain at the same level. If the DB2 Setup wizard does not have access to all of the images installed on the computer it cannot continue

Store all the fix packs in the same location, which allows the installer to find the fix packs for the products that it requires at the time of installation. All images must be uncompressed to subdirectories under the same parent directory.

The fix packs are in self extracting zip files. Uncompress them into the same directory. The resulting directory structure looks something like this example:

```
\\SERVER\FIXPAK \
├───CLIENT
├───SERVER
├───IIRW
└───QP
```

After you extract the fix packs to the same directory, you can deploy them in a similar manner to that of the initial product installation using a response file.

If you do not require any changes to the initial installation, you can likely use the same response file that you used to deploy your initial installation. Otherwise, to construct a response file, start with the sample response file. You can find the sample response file in the `\db2\windows\`samples directory.

In many cases the only change you need to make in the response file is to accept the license terms and conditions by specifying LIC_AGREEMENT=ACCEPT. If you want the installation to set additional database manager configuration parameters or profile registry variables, you can specify them in the response file.

Unless you are sure that there are no DB2 libraries in use, set KILL_PROCESSES to YES. The KILL_PROCESSES keyword stops any DB2 instances, and will immediately stop any applications that may be accessing DB2 libraries. If you have applications that you do not want to have stopped, ensure that these applications are not running at the time of installation.

Alternatively, you can also specify the **–f** parameter for **setup.exe**. However, you should keep as many options as possible in the response file so that you can reproduce the installation as consistently as possible.

Running the deployment script or definition on all computers

Previous sections have provided you with information about actions to take to prepare for the deployment. You can perform these actions once, incurring the upfront preparation costs, and then deploy to hundreds or thousands of computers. You can add extra computers later with almost no additional preparation.

There are two basic styles of deployment: *push* and *pull*.

- In a push deployment, a centralized location initiates the contact with each computer. This is usually suitable for cases where a centralized IT department wants to set up many computers without any user action. However, to do this, an automated service, such as rshd, sshd, SMS, or other deployment software, must already be installed and running on each computer to be "pushed" to, with appropriate access available to the IT department.

- In a **pull** deployment, each location in which a DB2 product is being installed initiates its own contact with the server to start the installation. Often, this operation is done using a link on a web page or in an email. This style is suitable for cases where very few computers require a DB2 product installation or no automated service is available to initiate a push installation.

Distributing IBM Data Server products using SMS

With SMS, you can set up the installation of DB2 or IBM Data Server products from a central location, and install it in an efficient and reliable manner across a computer network. SMS provides tools that automate deployment tasks, thereby reducing the amount of work that your users must perform. This installation method is suitable for a mass deployment of DB2 or IBM Data Server products on a large number of client computers that are all based on the same setup.

Prerequisites
You must have SMS 2003 installed and configured on your network for both your SMS 2003 server and your SMS 2003 client workstations.

For details on the following items, see *Microsoft's Systems Management Server Concepts, Planning, and Deployment Guide* and *Microsoft's Systems Management Server Operations Guide* at http://www.microsoft.com/technet/downloads/sms.mspx :

- SMS 2003 platform and prerequisite information

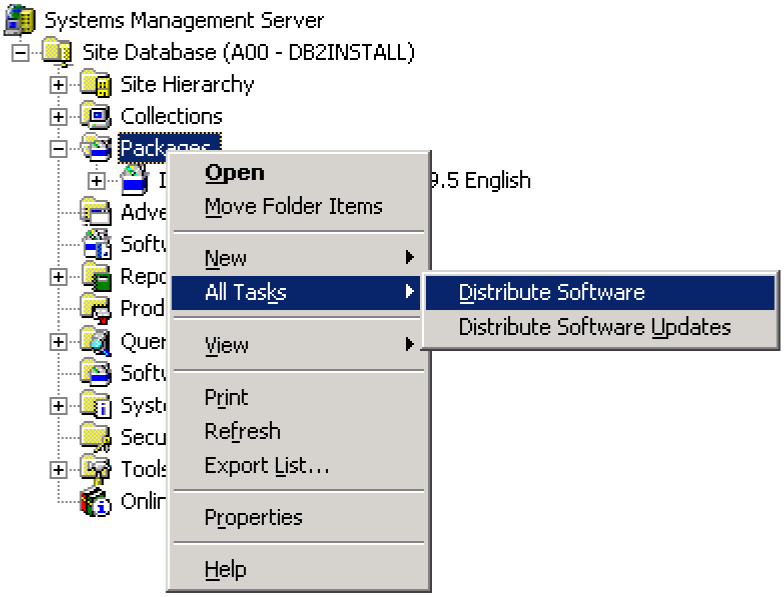
- Setting up SMS (including setting up the primary and the secondary sites)

- Adding clients to the SMS 2003 system

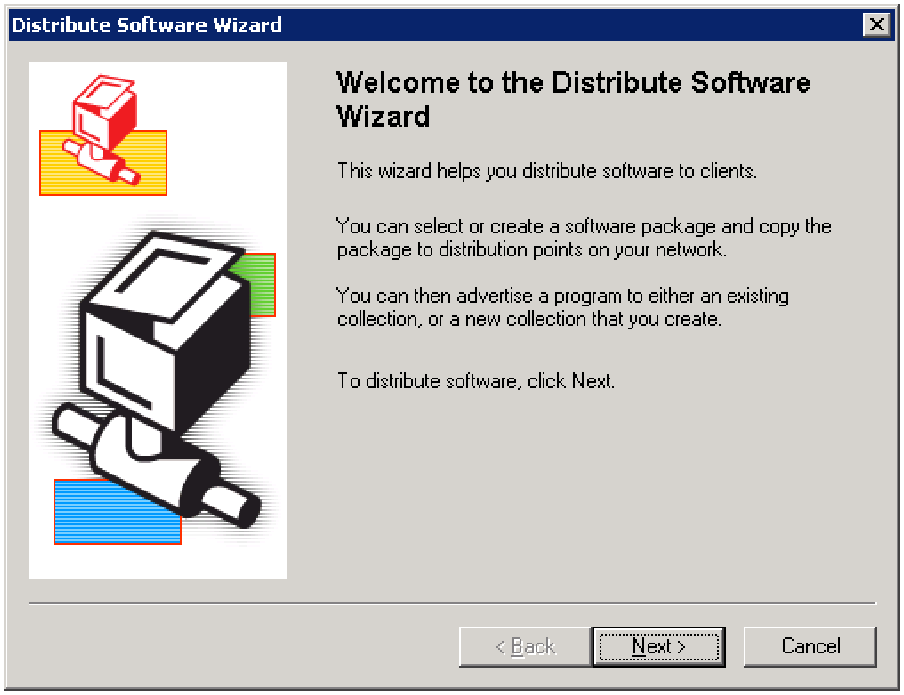
Procedure

To distribute DB2 or IBM Data Server products using SMS:

1. After creating a package, on the SMS Administrator Console, open the Site Database object tree and right-click Packages. Select **All Tasks** > **Distribute Software.**



2. On the Welcome to the Distribute Software Wizard page, click **Next**.



Welcome to the Distribute Software Wizard

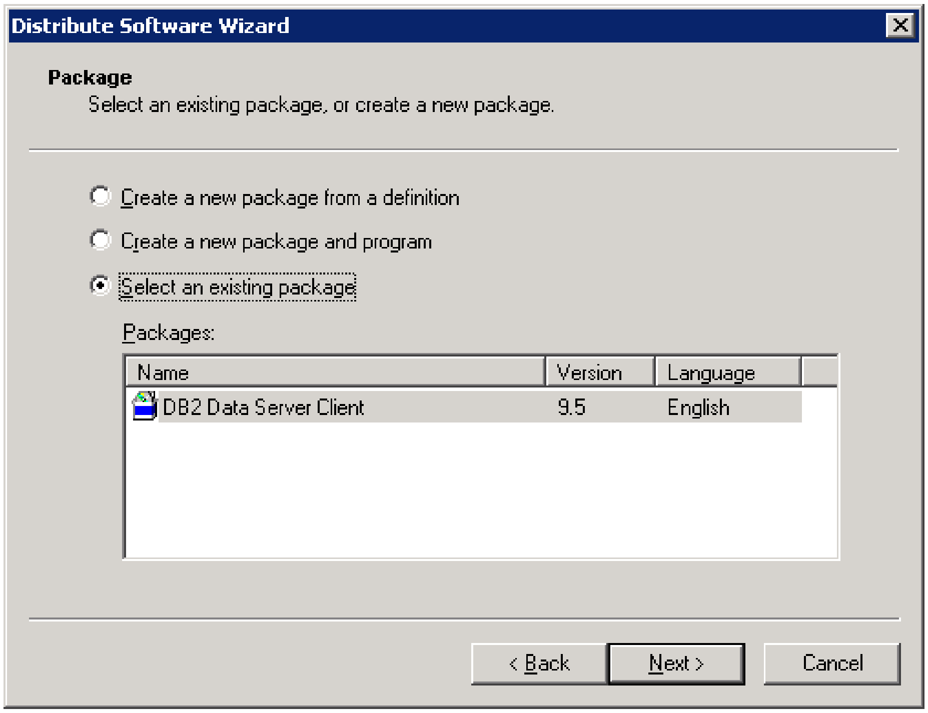
This wizard helps you distribute software to clients.

You can select or create a software package and copy the package to distribution points on your network.

You can then advertise a program to either an existing collection, or a new collection that you create.

To distribute software, click Next.

3. On the Package page, click **Select an existing package** and select the package you want to distribute. Click **Next.**



Distribute Software Wizard

Package

Select an existing package, or create a new package.

○ Create a new package from a definition

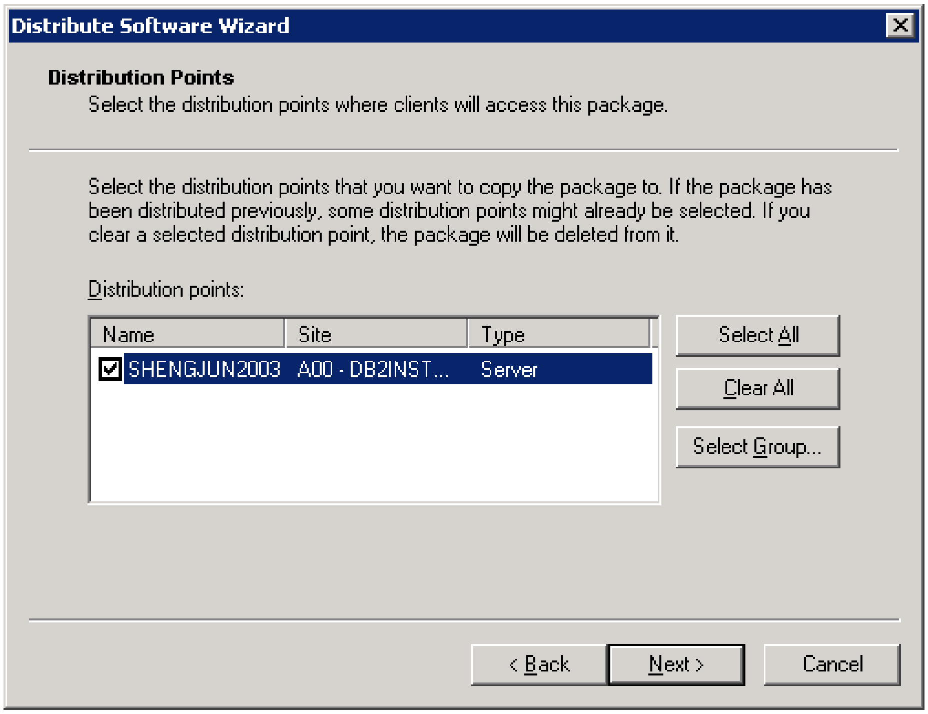
○ Create a new package and program

● Select an existing package

Packages:

Name	Version	Language
DB2 Data Server Client	9.5	English

4. On the Distribution Points page, select the distribution points for the package and click **Next**.



Distribute Software Wizard

Distribution Points

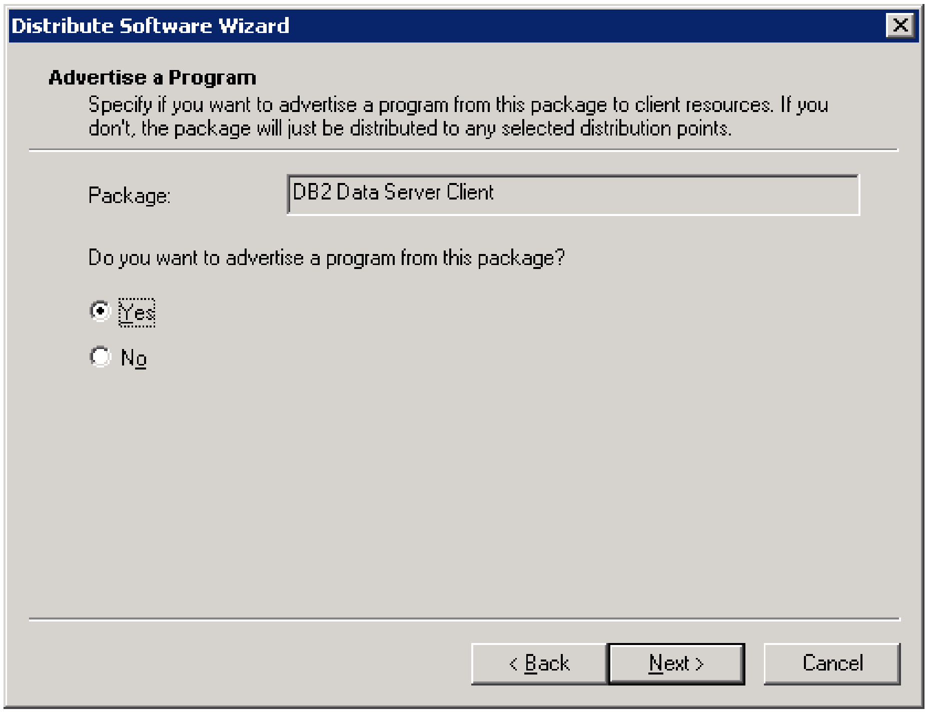
Select the distribution points where clients will access this package.

Select the distribution points that you want to copy the package to. If the package has been distributed previously, some distribution points might already be selected. If you clear a selected distribution point, the package will be deleted from it.

Distribution points:

Name	Site	Type
☑ SHENGJUN2003	A00 - DB2INST...	Server

5. On the Advertise a Program page, click **Yes**. Click **Next**.



Distribute Software Wizard

Advertise a Program

Specify if you want to advertise a program from this package to client resources. If you don't, the package will just be distributed to any selected distribution points.

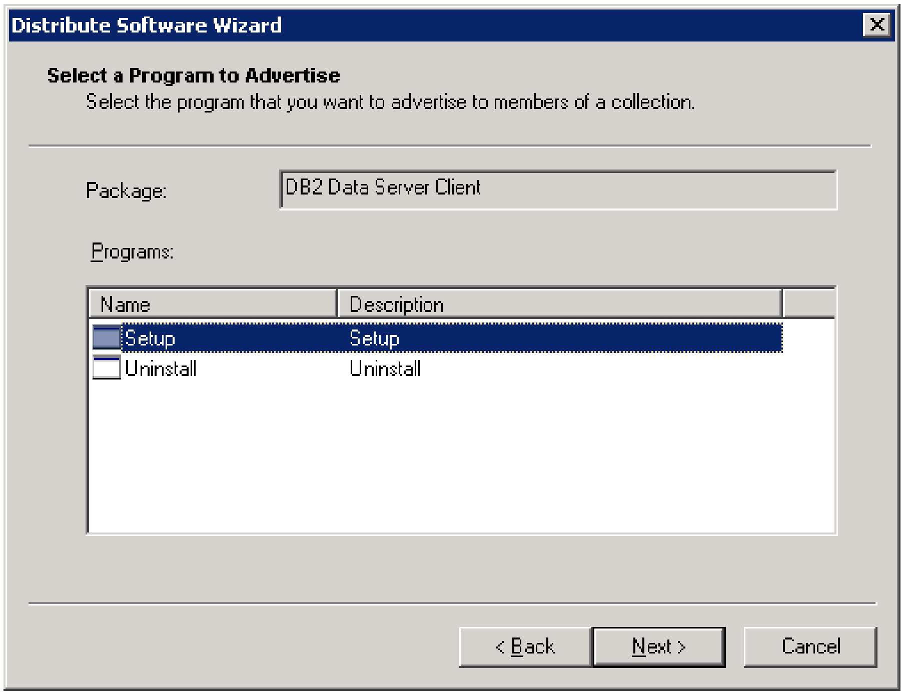
Package: DB2 Data Server Client

Do you want to advertise a program from this package?

● Yes

○ No

6. On the Select a Program to Advertise page, select the program to be advertised to the members of your distribution collection and click **Next**.



Distribute Software Wizard

Select a Program to Advertise

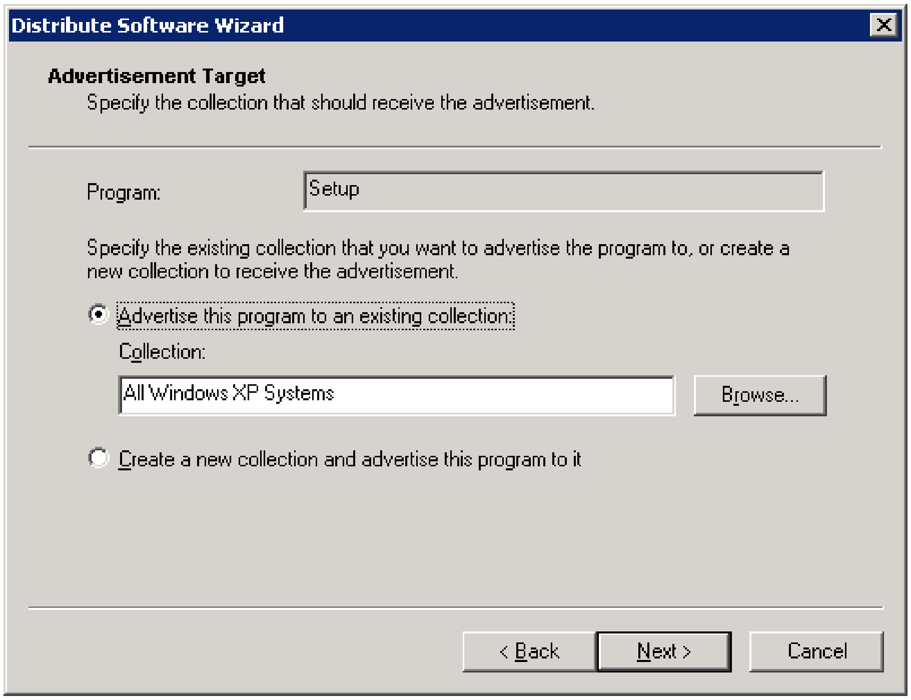
Select the program that you want to advertise to members of a collection.

Package: DB2 Data Server Client

Programs:

Name	Description
Setup	Setup
Uninstall	Uninstall

7. On the Advertisement Target page, click **Advertise this program to an existing collection** and specify a collection name or **Create a new collection and advertise this program to it**. Click **Next**.



Distribute Software Wizard

Advertisement Target

Specify the collection that should receive the advertisement.

Program: Setup

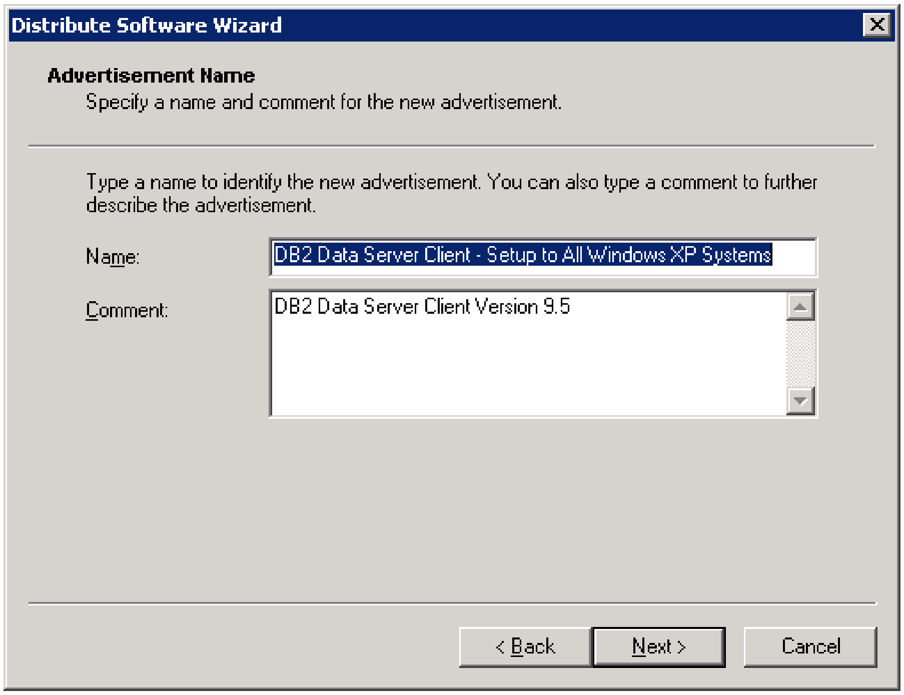
Specify the existing collection that you want to advertise the program to, or create a new collection to receive the advertisement.

● Advertise this program to an existing collection:

Collection: All Windows XP Systems

○ Create a new collection and advertise this program to it

8. On the Advertisement Name page, in the **Name** field, type a name to identify the advertisement and, optionally, add comments. Click **Next**.



Distribute Software Wizard

Advertisement Name

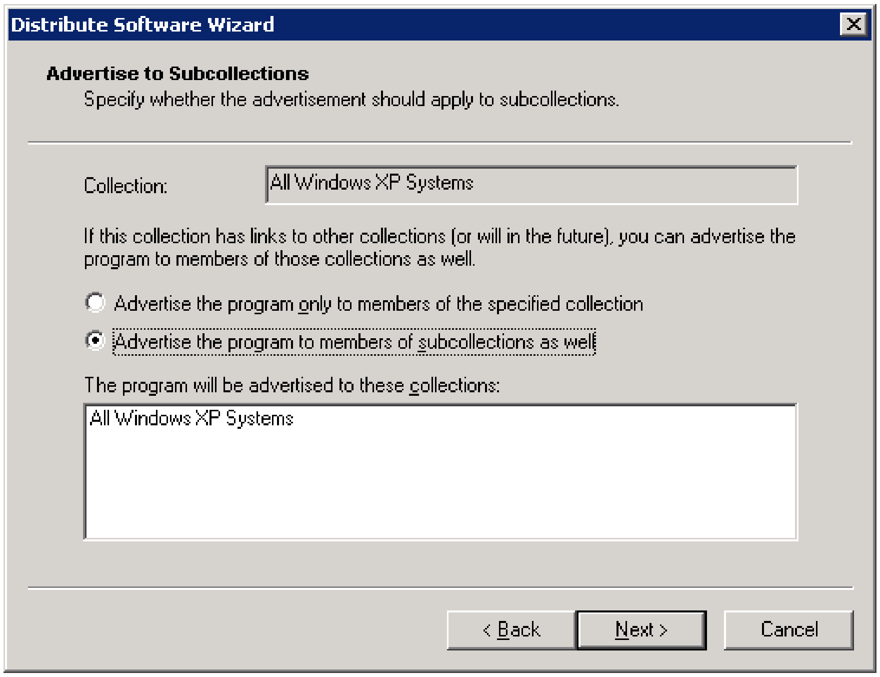
Specify a name and comment for the new advertisement.

Type a name to identify the new advertisement. You can also type a comment to further describe the advertisement.

Name: DB2 Data Server Client - Setup to All Windows XP Systems

Comment: DB2 Data Server Client Version 9.5

9. On the Advertise to Subcollections **page, click Advertise the program only to members of the specified collection** or **Advertise the program to members of the subcollection as well.** Click **Next**.



Distribute Software Wizard

Advertise to Subcollections

Specify whether the advertisement should apply to subcollections.

Collection: All Windows XP Systems

If this collection has links to other collections (or will in the future), you can advertise the program to members of those collections as well.

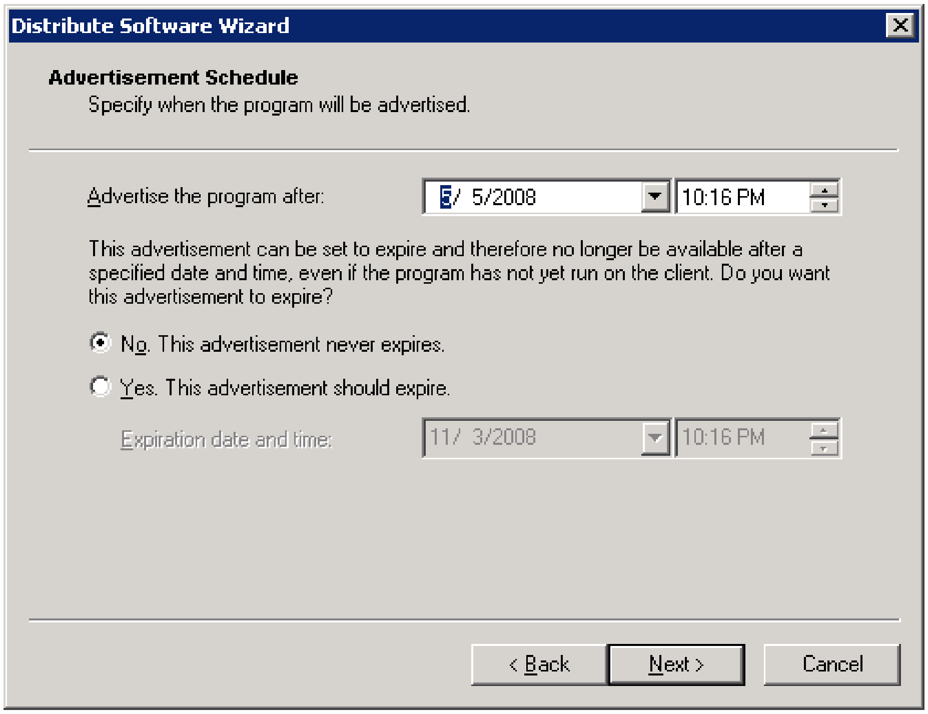
○ Advertise the program only to members of the specified collection

● Advertise the program to members of subcollections as well

The program will be advertised to these collections:

All Windows XP Systems

10. On the Advertisement Schedule page, specify when you want the program to be advertised to your SMS clients. You also have the option to set an expiry date for this advertisement. Click **Next**.



Distribute Software Wizard

Advertisement Schedule

Specify when the program will be advertised.

Advertise the program after: 5/ 5/2008 10:16 PM

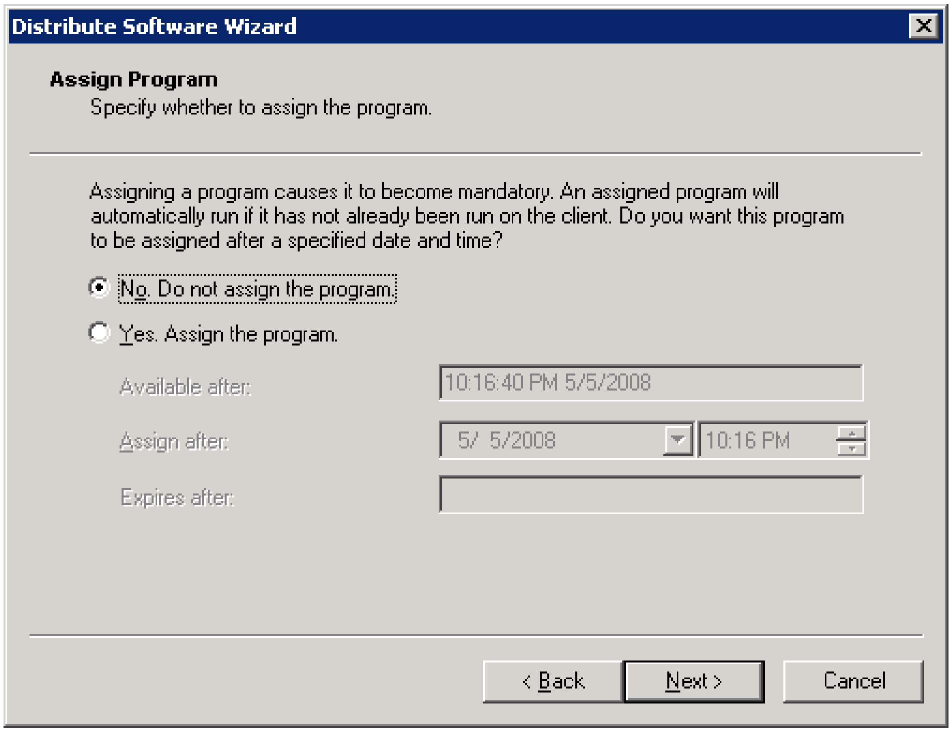
This advertisement can be set to expire and therefore no longer be available after a specified date and time, even if the program has not yet run on the client. Do you want this advertisement to expire?

● No. This advertisement never expires.

○ Yes. This advertisement should expire.

Expiration date and time: 11/ 3/2008 10:16 PM

11. On the Assign Program page, specify whether you want to assign the program which will make it mandatory for your SMS clients. Click **Next**.



Distribute Software Wizard

Assign Program
Specify whether to assign the program.

Assigning a program causes it to become mandatory. An assigned program will automatically run if it has not already been run on the client. Do you want this program to be assigned after a specified date and time?

● No. Do not assign the program.

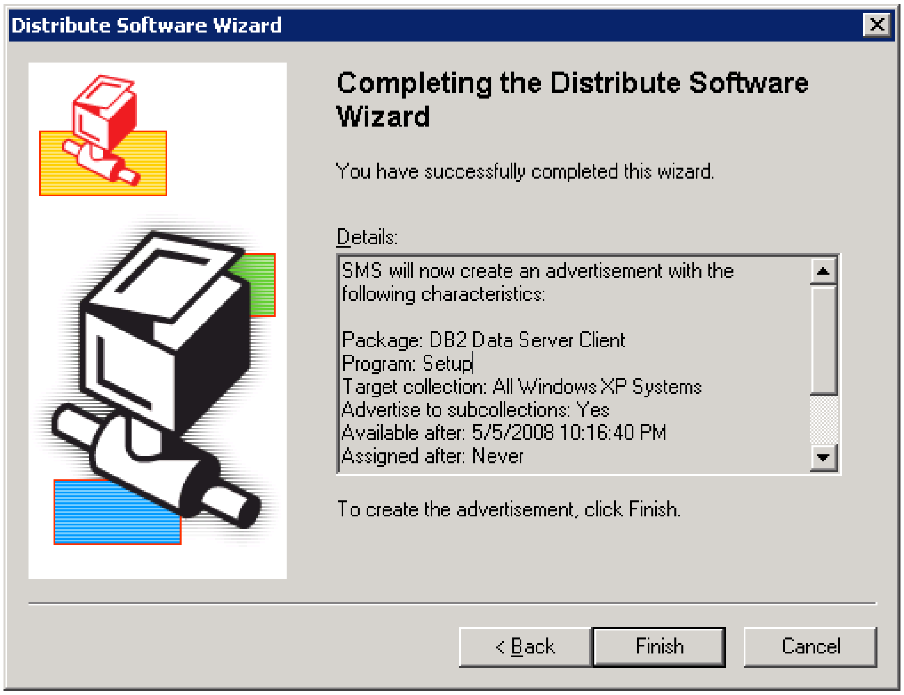
○ Yes. Assign the program.

Available after: 10:16:40 PM 5/5/2008

Assign after: 5/ 5/2008 10:16 PM

Expires after:

12. On the Completing the Distribute Package Wizard page, click **Finish** to advertise the program to your SMS clients. After you distribute a DB2 or IBM Data Server product using SMS, the product is automatically installed on the clients on your computer network.



Distribute Software Wizard

Completing the Distribute Software Wizard

You have successfully completed this wizard.

Details:

SMS will now create an advertisement with the following characteristics:

Package: DB2 Data Server Client
Program: Setup
Target collection: All Windows XP Systems
Advertise to subcollections: Yes
Available after: 5/5/2008 10:16:40 PM
Assigned after: Never

To create the advertisement, click Finish.

Example of a push installation using a script (UNIX)

The following example shows the code that you might use in a shell script, for example, db2deployment.sh, to push out a DB2 installation:

```
#! /bin/sh

dst_computers="wrkstn1 wrkstn2 wrkstn3"

for comp in $dst_computers; do

  ssh -l root $comp "mkdir /share;

    mount -t nfs -o ro fileserver:/bigshareddisk /share;

    /share/rtcl/deployment.sh;

    umount /share; rmdir /share" >> /var/log/deploy.$comp &

done
```

The idea is to put this code, or code substantially similar to it, in a shell script, for example, db2deployment.sh, which you can use to push out the DB2 installation.

Since most of the work is done by the deployment script, there is little left to be done by the push installation script.

Two of the assumptions made for this example are that ssh is set up on all of the workstations and that the user running it is authorized to access root on each workstation without using a password.

Example of a pull installation using a script (UNIX)

The following script is almost identical to the script in the push example. Again, the deployment script does most of the work, and you must set up your environment so that the user can run the script as root.

```
#! /bin/sh

(mkdir /share

mount -t nfs -o ro fileserver:/bigshareddisk /share

/share/rtcl/deployment.sh >> /share/logs/`hostname`.out

umount /share; rmdir /share) >> /var/log/deploy.$$
```

Reducing the size of a DB2 product installation image

DB2 data server products provide flexibility by including a large number of installable features in their installation images. However, the large footprint size of the installation images might cause a problem. You can avoid this potential problem on Windows platforms using the **db2iprune** command line utility. The **db2iprune** utility reduces the size of your installation image.

The **db2iprune** utility is located in the `db2/`*`platform`*`/utilities/db2iprune` directory, where *platform* is the platform identifier. The utility consists of an input file and a **db2iprune** executable file.

Use the input file (.prn file) to specify which features and components you want to remove from the installation image. The **db2iprune** executable file removes the DB2 payload or the cabinet (.cab) files associated with those features and languages (payload or cabinet files are a number of related files that are packaged together for deployment purposes), and makes it impossible to remove a component from the installation image if a different feature in the image requires that component to be present.

Only the unspecified features, and any components that the unspecified features require, are included in the pruned installation image. When you install the pruned image, features that you removed are not displayed in the installation GUI panels. After a pruned image is installed, the result is the same as if the user had selected only that subset of features during an installation from a full image.

The result of using the **db2iprune** utility is a new, smaller installation image that you can install using the regular product installation methods.

Pruning a DB2 product installation image using the db2iprune utility

Use the input file (.prn file), located in the `db2/platform/utilities/db2iprune` directory, to uncomment the products, features, and languages that you want removed from the installation image.

Comments are denoted by an asterisk (*). The **db2iprune** input file includes the following keywords:

- The **PRUNE_PROD** keyword identifies the DB2 database product to remove. If the installation image contains only 1 product, you cannot remove it using this keyword.

- The **PRUNE_LANG** keyword specifies which languages to remove from the product installation image. To remove the payload or .cab files for a language, as well as the doc files in the installation image for this language, uncomment the equivalent PRUNE_LANG keyword.

- The **PRUNE_COMP** keyword identifies which components to remove from the DB2 product installation image. To remove the payload or .cab files for a feature, simply uncomment the equivalent COMP keyword.

For a complete list of the keywords for the **db2iprune** utility, search the DB2 Information Center using terms as “db2iprune” or “Reducing the size of your DB2 product installation image”.

You can uncomment several PRUNE_PROD**,** PRUNE_COMP, or PRUNE_LANG keywords in the same input file to remove several products, features, or languages, as shown in the following example:

```
*REMOVE_PROD                              = CLIENT
*REMOVE_COMP                              = DB2_UPDATE_SERVICE
*REMOVE_COMP                              = CONTROL_CENTER
*REMOVE_LANG                              = FR
*REMOVE_LANG                              = CN
```

Uncommenting these lines removes the following items:

- The IBM Data Server Client product. It will not be selectable for product selection, and files specifically for this product will be removed.

- DB2 Update Service and Control Center features. They will not be selectable for component selection, and any files associated with these features will be removed.

- The language files for French and simplified Chinese.

From the command line, issue the **db2iprune command** using the following syntax:

- On UNIX and Linux platforms:

```
db2iprune -r input_file_path -o destination_directory
```

- On Windows platforms:

```
db2iprune -r input_file_path -p root_directory -o
destination_directory
```

where:

- *input_file_path* represents the full path to the **db2iprune** input file.

- *root_directory* (Windows only) represents the full path to the root directory of the source installation image. This directory contains the setup.exe file and is the root directory of the product installation DVD.

- *destination_directory* represents the full path to the location where the new pruned product image will be copied. Ensure you have write access to this directory.

Restriction: You must run the **db2iprune** command directly from the product installation image on UNIX and Linux.

Use any of the regular installation methods to install and maintain the pruned DB2 product installation image. If you select the typical installation type, the installation process installs the typical set of components without the components you pruned from the image.

The products and components removed by the **db2iprune** utility are not displayed as options during the install. However, the removed languages will still be displayed in the language selection panel. Therefore, ensure that you do not select a language that you have removed from the image using the **db2iprune** utility. If you select a language that you have removed, you will receive an error message.

If you use the **db2iprune** utility with a product fix pack image, ensure that the fix pack image contains all of the features and components that were initially installed. If the fix pack image does not contain all the installed features or components, you will receive an error about missing payload or .cab files when you attempt the fix pack application.

Validating your DB2 copy

The db2val command verifies the core functionality of a DB2 copy by validating the installation, instances, database creation, connections to the database and health of partitioned database environments. You can run this command at any time after your product installation.

After a fix pack update, you can use the **db2val** command to check whether your existing DB2 instances in the DB2 copy are good to use.

This validation can also be helpful if you manually deployed a DB2 copy on Linux and UNIX operating systems by using tar.gz files. The **db2val** command can quickly ensure that you correctly configured the copy and that the copy is what you expect it to be.

Validate your DB2 copy by entering the following command:

```
installpath/bin/db2val
```

where *installpath* represents the installation location of your DB2 copy.

Using a shared DB2 copy in a virtualized environment

Starting in DB2 V9.7, you can install a DB2 copy in the AIX global environment and share that copy with workload partitions (WPARs) on the same AIX system.

You can install a DB2 copy on the AIX global environment and use it not only from the global environment, but also from all WPARs if the DB2 product is installed under `/opt` or `/usr` directory shared among the global environment and WPARs. This does not need write permission to install path from WPARs. On each WPAR that has read permission on the installation path, you can perform instance creation, update, upgrade and drop actions.

When the DB2 copy is updated on the global environment, all the instances on active WPAR that are using the global copy will be automatically updated, assuming the instances were stopped on all the related WPARs.

Uninstalling the shared DB2 copy from the global environment will check if there are any instance exists on the active WPARs to ensure that no instances that are using the shared DB2 copy will be broken by the uninstall. This action is not recommended in production system, but it is useful in testing and development environment.

Example
Assume that an AIX server has two WPARs: wpar1 and wpar2.

1. As root user, install a DB2 copy in the /opt/IBM/db2/V9.7 directory on the global system.

```
imgDir/db2_install –p ese –b /opt/IBM/db2/V9.7
```

2. As root user, log in to wpar1 and create a DB2 instance by running the **./db2icrt** command from the /opt/IBM/db2/V9.7/instance directory. Then log in to wpar2 and create a DB2 instance by running **the ./db2icrt** command from the /opt/IBM/db2/V9.7/instance directory.

```
clogin wpar1

cd /opt/IBM/db2/V9.7/instance

./db2icrt –u db2fenc1 db2inst1

exit

clogin wpar2

cd /opt/IBM/db2/V9.7/instance

./db2icrt –u db2fenc1 db2inst1

exit
```

3. To update the DB2 copy in the /opt/IBM/db2/V9.7 directory, perform these steps:

   - Stop all instances on wpar1 and wpar2.

   - In the global environment, issue the following command:

```
fixpackImgDir/installFixPack –b /opt/IBM/db2/V9.7
```

After the installFixPack command completes, the db2inst1 instances on wpar1 and wpar2 are updated if they were active when you ran the **installFixPack** command.

You can also similarly share a Solaris zones or NFS installation. You should manually stop all instances that are using the shared DB2 copy under Solaris zones or before you apply a fix pack and updated the DB2 copy by running the db2iupdt command. Before uninstalling the shared DB2 copy, you have to make sure that all instances are dropped as there is no automatic checking for shared DB2 copies on Solaris zones or NFS.

Best Practices

- Use a response file for all mass deployments
- Use the latest the DB2 fix pack product images for your deployment instead of installing and applying a DB2 fix pack in two steps
- Use a code server to speed up deployment
- Use the DB2 Setup wizard to create your base response file, especially for deployments in a DPF environment
- Use the **db2rspgn** utility to capture the settings from an existing installation
- Use the **db2cfexp** and **db2cfimp** utilities to distribute connectivity information
- If disk space and network bandwidth are issues, use the **db2iprune** utility to reduce the size of the DB2 product image

Summary

Deploying hundreds or even thousands of DB2 clients or servers on Windows, Linux, or UNIX platforms does not have to be significantly more difficult than deploying a single client or server. With some planning and testing, mass deployment of DB2 products can become a regular administrative task.

Using the best practices outlined in this document can significantly improve the ease of mass deployment across your computing environment. The unattended installation method described in this paper is typically the simplest and most flexible method, saving you valuable installation and configuration time.

Further Reading

- DB2 Best Practices - http://www.ibm.com/developerworks/db2/bestpractices/

Contributors

Glen Sheffield
DB2 Technical Evangelist

Susan M Althoff
IBM Certified Consulting I/T Specialist

Aimin Wu
System Optimization Competency Center

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.