NeoData Installation and Technical Guide

Table of Contents

Technical Information

Installation & Setup

Features

Overview

The **NeoData NICU Patient Data System** is a Microsoft Windows-based client-server database application intended for use directly by physicians and other clinical staff in the Neonatal Intensive Care Unit setting. It performs the following general functions:

- Capture of a wide range of clinical patient information.

- Tracking of patient diagnoses and management plans.

- Automatic creation of major patient chart documentation.

- Storage and retrieval of all captured data for current and discharged NICU patients.

- End-user creation of queries and reports.

- Support of various other NICU operations including daily fluid orders, growth charts, high-risk follow-up clinic, and physician billing.

- Support for the Vermont Oxford Network data collection process for participating centers.

Data entry is typically performed by physicians and physician extenders themselves; clerical staff involvement is not required.

Supported Database Engines

NeoData is a conventional client-server data application. It uses Microsoft **ADO.NET** for data access. **Microsoft Data Access Components** (MDAC) Version 2.6 or later must be installed on the workstations. NeoData can operate in two modes:

- **SQL Server Mode:** In SQL Server mode, the system uses a **Microsoft SQL Server** database; SQL Server versions 2008 and above are supported. As part of the installation process, a function within the application itself is used to create the database using a Microsoft Access database file as a template.   Isoprime strongly recommends that you use SQL Server as your database. It is more reliable and more efficient than MS Access. Also, SQL Server is required if you plan to use certain NeoData features such as the HL7 Interface, audit trails, and note signing.

- **Access Mode:** In Access mode, the system uses a **Microsoft Access** database; the Access 2000 file format is supported. Access does not need to be installed on the client or on the server for NeoData to run; all necessary Jet database engine files are available for installation. Isoprime recommends that MS Access only be used if absolutely necessary and that it only be used when running NeoData on a single PC.

Privacy, Security, & HIPAA

NeoData includes an extensive set of features designed to control access to the system and to assist with your compliance with HIPAA regulations. These include:

- Password-protected user logins.

- Grouping of users by role with detailed group-level control of read and write access to all data screens and all program functions.

- Auditing of user logins with configurable lockout of users after failed logins.

- Auditing of access to patient data screens.

- 128-bit strong encryption of all passwords and other security settings.

- Control of minimum and maximum password length and maximum password age (for password expiration).

- Locking of patient data after discharge.

- Automatic logout after a specified idle time.

- Support for various database authentication models.

- Auditing of additions, deletions, and edits of all patient data records, with field-level display of audit entries (SQL Server mode only).

Configuration

System: With a few exceptions, all configuration settings for the database connection (SQL Server or Access) and for the privacy & security features can be made from within the application's configuration screens. NeoData manages user names and passwords for its own logins, but does not create SQL Server user logins.

User Interface: NeoData provides a rich set of configuration options covering many aspects of the application behavior. These allow customization of such things as document layout and content, data entry forms, drop-down lists, and billing support features. In general, these relate to issues that are clinical rather than technical. Since they primarily affect how the NICU team uses the system, they are most often managed by clinical or NICU departmental staff rather than technical support staff. Most of these are global to all users; users should make decisions about these issues as a group.

Defaults are provided for all of these settings. Some need to be adapted for your site prior to live use of the system, but most can be optimized gradually thereafter.

Interfaces

Incoming: An HL7 Interface Module is available as an optional add-on to NeoData. This inteface currently supports inbound ADT, Maternal Data and Labs information via HL7 messages. See the Inbound Transactions topic for more information.

Outgoing: NeoData can produce the major patient documents (admission summaries, progress notes, and discharge summaries) in electronic form for inclusion in an electronic medical record system. Electronic documents can be created in ASCII, RTF, PDF, or HTML format, and can optionally include customizable headers for proper integration with your document management system. The optional HL7 Interface Module can also send the text of the major patient documents to other systems via HL7 messages. It can also send certain clinical data such as immunizations and diagnoses, to third-parties. See the Outbound Transactions topic for more information.

Standard Documents

NeoData produces standard patient documentation including admission summaries, daily progress notes, and discharge summaries. Extensive configuration features are available to customize these documents to the needs of the clinical staff and to meet the medical records requirements at your site. Other documents produced include growth charts, problem lists, procedure notes, interval patient summaries, daily IV orders, and billing charge sheets.

The major documents (admission summaries, progress notes, and discharge summaries) can be created in electronic form for inclusion in an **electronic patient record system** (see **Outgoing Interfaces** above).

Queries & Reports

NeoData includes a **Query & Report Module** which allows clinical users to create a broad range of custom queries and associated printed reports. No experience with other querying tools or with SQL is required; the Query Module is designed specifically with the non-technical user in mind.

Since NeoData uses non-proprietary databases (Microsoft SQL Server or Microsoft Access), other queries and reports beyond the capability of the Query Module can be created with Microsoft Access or any standard Windows database system.

Before Installation

Prerequisites

The following software must be installed on each computer which will be running NeoData:

- Microsoft .NET Framework 3.5 SP1.

- Microsoft Data Access Components (MDAC) 2.6 or later.

- Microsoft Jet 4.0 Database Engine Service Pack 5 or later.

Some or all of these may already be installed on any given computer. For instance, the Microsoft Jet 4.0 Database Engine is included with Windows; MDAC 2.6 is included with Windows. Just in case, the installation packages for the above products are included on the NeoData installation disk. You may install these products either before or after completing the NeoData installation.

Prior Versions of NeoData

If you have an earlier version of NeoData installed, take care to protect existing data that may already be present on your computer. **If done properly, installing a newer version of the program should not overwrite your existing data files, but you should back them up as a safety precaution.**

If you use a **Microsoft Access** database that uses a **file format** earlier than Access 2000, Isoprime recommends that it be converted to Access 2000 format.

Screen Resolution

NeoData is designed to run at a screen resolution of 1024 by 768 pixels or higher. It will run at 640 by 480 resolution, but it will not be fully usable. The installation program itself may be run at **any** screen resolution; if necessary, you may change the workstation's resolution **after** installation.

When running NeoData on a computer with a screen resolution of **1024 by 768 pixels**, the **Taskbar** (typically displayed at the bottom of the screen) may hide part of the NeoData screen. **To hide the Taskbar when NeoData is running:**

1. Click the **Start** button.

2. Select **Settings.**

3. Select **Taskbar & Start Menu.**

4. In the **Taskbar Properties** dialog box, turn off **Always on Top** OR turn on **Auto Hide**.

Power Protection

In order to protect your patient data, Isoprime **STRONGLY** recommends that each workstation running NeoData and any computer acting as a **file server** should be connected to a **uninterruptable power supply**. These devices supply uninterrupted power to the computer in the event of an AC power outage and significantly reduce the likelihood of data file corruption. Additionally, a regular and reliable program of **backing up your critical files** should be instituted. Your institution's computer support staff should be able to assist you with both of these issues.

System Requirements

NeoData Client Workstation

Operating System:
Windows 7 or above

Processor: 1 Ghz processor or above

Memory: 2 GB RAM

Local Disk Storage: 50 MB

Server - SQL Server Database

NOTE: Isoprime strongly recommends using SQL Server as the database management system. If the signing feature will be used, SQL Server must be used.

Network Operating System:
Windows Server 2008 or above

Database System:
SQL Server 2008 or above

Processor: 1.4GHz or above

Memory: 2 GB minimum

Initial Server Disk Space: 20 GB

Typical Database Growth (30-bed NICU): 100 MB/year if the electronic signing feature is not enabled, if the electronic signing feature is enabled there would an additional 900 MB/year for a total of 1,000 MB/year (1 GB/yr)

Server - Access Database

NOTE: Isoprime strongly recommends using SQL Server as the database management system. If the signing feature will be used, SQL Server must be used.

Network Operating System:
Windows Server 2008 or above

Processor: 1.4GHz or above

Memory: 2 GB minimum

Initial Server Disk Storage: 20 GB

Typical Database Growth (30-bed NICU): 100 MB/year

Printers

NeoData is designed to print to any Windows printer visible to the workstation, including faxing software configured as a printer. The program does not require any specific printer or feature. It supports certain features such as duplexing and multiple paper trays when available.

Networking

NeoData does not require any specific network operating system. When used across a network, the application must be able to access the necessary file directories and, when running in **SQL Server Mode**, must be able to connect to an instance of SQL Server.

Recommended minimum bandwidth over the network is **8-10 Mbps**.

NeoData supports the **Universal Naming Convention** for file access; a mapped drive letter is not required.

Installation

Starting The Installation

Note: You can run the installation either from a server or from a workstation. There is no software component of NeoData that runs on the server (unless you use the server as a workstation to run the application itself). You should decide whether client workstations will run the application executable from a local drive or a network drive; the latter is recommended since this simplifies both the initial installation and the process of applying updates.

Close all applications currently running in Windows. To begin the installation, place the disk in your CD-ROM drive. The Setup program should start automatically after a few seconds. If it does not, continue as follows:

1. From the **Start** menu, select **Run.**

2. Assuming that your CD-ROM drive is drive **D:**, type **D:\nd6setup** and press **<Enter>**.

3. Follow the on-screen directions. Please read all information **carefully**.

4. The installation program will display a series of dialog boxes asking how you want to set up NeoData. Some of these options are discussed in the next several sections.

Alternatively, you can copy the **ND6Setup.EXE** installation file to a server on your network and run the installation file directly.

Selecting Features To Install

The installation program will ask you which features you wish to install. The options are:

- **Program & Help Files:** This feature installs the main program file (**NEODATA.EXE**), other code files used by the program (***.DLL**), the help file (**NEODATA.CHM**), and the main dictionary file (**NEODATA.DCT**), among others.

- **Data & Configuration Files:** This feature installs the Access database file (**NEODATA.MDB**) and the configuration file (**NEODATA.CFG**), among others. If you plan to use a SQL Server database, the Access database file will be used by NeoData to create the SQL Server database after you complete the installation; the Data & Configuration File do not need to be installed on the same server as the SQL Server instance.

- **Desktop Shortcut:** This feature installs a shortcut on the desktop which you can use to start the program.

- **Program Menu Shortcuts:** This feature installs several shortcuts in the Program Menu which you can use to start the program and view the Help document.

Please make these selections carefully, based on the following considerations:

- If you are running the installation for the first time (i.e. no previous version is present), then you must install both the **Program Files** and the **Data Files**.

- If you are installing over a **previous installation**, you should normally **not** install the **Data Files**. If you do so, it is possible to overwrite your data or configuration files. NeoData's installation program is designed to protect your existing data, but the safest course of action is always to back up your data files before installation.

- If you are setting up multiple workstations, you only need to install the **Data Files** once (if at all; see above). If you are installing the **Program Files** on a file server (the recommended approach) where it will be shared by all workstations (rather than on each local drive), you also only need to install the Program Files once. In this case, you can still use the installation program to create shortcuts on each workstation; you must enter the Program File Directory and the Data File Directory correctly so that the install program can set up the shortcuts properly (unless you choose to create each workstation's shortcuts manually).

Selecting Directories

Depending on the features you select, the installation program will ask you for the **directories** in which to place the installed files.

The directories for the Program Files (the **Program File Directory**) and for the Data Files (the **Data File Directory**) may be the same or different depending on the choices you make during installation (see below). By default, NeoData will create a directory named **C:\ND4** and use it for both the Program Files and the Data Files; however, in most cases you will want to change this to a network directory. Note that the default directory names do not change for each new version of NeoData. They will typically all be **ND4**. If you want to place the Program and/or Data Files in a location other than the default directory, highlight the desired option and click the **Browse** button to open a file selection dialog box. Type a new directory name in place of the default, or browse the list to find the correct place for the NeoData files.

Select the directory locations based on the following considerations:

- The **Data Files** must be obviously be placed on a network drive so that the files may be shared (unless you are creating a single-user system).

- The **Program Files** may be installed on each workstation's local drive (which may produce slightly better performance if network bandwidth is limited), but the recommended approach is to install them on a network drive and have all workstations share the same files. Placing the program files on a network drive makes for quicker

installation and simplifies the installation of updates. If you use this latter approach, there is usually no reason why the Data Files and Program Files can't be placed in the same directory.

NOTE: NeoData and its HL7 interface cannot be installed into same location (the same directory on the same disk drive). This configuration could cause NeoData and its HL7 interface to behave unexpectedly. If NeoData and its HL7 interface are installed into the same folder, contact Isoprime for assistance in correcting the installation.

After Installation

Check For Updates

Before you begin using NeoData, you should check the Isoprime web site for the most current version of the program:

1. Go to the Isoprime web site (**www.isoprime.com**) and click on the **Program Updates** link in the NeoData section.

2. Download and run the **Update Installation Program** (ND6UPDATE.EXE). Follow the instructions.

3. On the **Select A Function** dialog, select the **Update existing NeoData Version 4, 5 or 6 program files** option.

4. On the **Select Destination Directory** dialog, click **Browse** and enter the path to your existing executable file (NEODATA.EXE). Note that if you have a working NeoData shortcut, its **Target** property will show you the correct executable file path. When you return from the **Browse** dialog, be sure that the **Program File Directory** is entered correctly.

5. Before copying files, the installation program will ask you for a password. If you don't have the current password, contact Isoprime .

Passwords

NeoData uses **Administration Passwords** to limit access to the many configuration options and to certain other operations. Access to these functions by persons who are unfamiliar with the details of the program could adversely affect your operation or result in loss of data. In general, only persons acting as **System Administrator** should know the passwords. The default Administration Password is '**nicu4321**'; this should be changed immediately after installation.

In addition, there is a separate **Billing Password** for the billing functions. Since you may or may not need to limit access to these functions, this password can be disabled altogether. The default Billing Password is '**charges**'.

Getting Started

Help documentation for NeoData can be accessed without running NeoData by opening the **Start Menu** and selecting the **Programs|NeoData|NeoData Help** option. You may also view the **Help document** from within the program itself via the **Help|Contents** menu option. We recommend that new users begin by reading the first two chapters, **Introduction** and **Getting Started**.

This guide may be reviewed at any time by selecting the **Technical Information** topic in the online **Help document**.

New features and software fixes for are documented on the Isoprime web site at http://www.isoprime.com/neodata/documents/nd6update.doc.

At least one person at your institution should familiarize themselves with the Help information and other documentation as fully as possible; your team cannot expect to make full and efficient use of the program otherwise. The System Administrator should pay particular attention to the **Configuration** section of the Help document. This information will allow you to adapt the program as fully as possible to the specific needs of your unit. Some configuration of the system will be necessary before it is used for actual patient care; see the **Initial Configuration** topic.

The database file shipped with NeoData contains no patient data. None of the patient data functions will respond until you have added new patients using the **Procedures|New Admission** menu option.

Initial Configuration

Most system configuration choices can be made as you actually use the program and become more familiar with its features. However, there are some configuration tasks which should be carried out before you begin formal use of the system. These involve editing certain lists in the **List Configuration Form** and reviewing various settings in the **General Configuration Form**. Note that the General Configuration Form is divided into a number of sections and subsections.

NOTE: It is HIGHLY recommended that you convert your MS Access database to SQL Server as part of your initial configuration. Many new features only apply when using SQL Server (such as auditing) and the software is only Meaningful Use certified if you are using it in SQL Server mode. Instructions for creating and using a SQL Server database are included below. For more information on setting up NeoData for Meaningful Use please see the Meaningful Use Overview topic.

To access the configuration forms, select the **Configuration** option from the **Tools** menu, enter your **Administration Password** (the default password is 'nicu4321'), and click the appropriate button (**General** or **List**). Within any configuration form section, you can click the **Help** button to see the documentation for the settings in that section.

Note: The configuration items given here include only those that you are most likely to need to change. There are many other settings that you **might** need change, depending on your needs. If you have questions about how to configure the program for your specific requirements, review the configuration topics in this Help document and/or consult Isoprime.

General Configuration

General > System

- **Institution Name & Location:** These settings are used on various printed documents.

- **Practice Name & Location:** These are used only on certain billing-related documents.

General > Identifiers

- **Admission ID Length, Mask, & Label:** The Admission ID is the admission-specific (or encounter-specific) identifier. The **Mask** is simply a list of allowed characters for this identifier; the order of these characters in the mask is not significant. The **Label** determines how the identifier will be labeled on-screen and in printed documents.

- **Medical Record ID Length, Mask, & Label:** The Medical Record ID is the patient-specific identifier.

Documents > Layouts

- **Layout Settings:** For each document type, there are a number of settings (e.g., margins) that control the exact look of the document.

Security > Administration & Administration Passwords

- **Administration Passwords:** The default administration passwords may serve your needs, but if you would like greater control over access to password-protected features, you may want to change some of these settings and/or create new passwords.

Security > Logins

- **Require NeoData Login:** By default, no authentication into the application is required at startup. However, you will probably want to turn on the login requirement and create user accounts as needed.

- **Login Settings:** You can control user password requirements and other related options in this section. Note that these settings apply to user accounts, not to Administration Passwords.

Security > Users & User Categories

- **User Accounts:** Create user accounts as needed. Note that every user must be assigned to one (and only one) User Category.

- **User Categories:** Each user must belong to a category. Privileges to use the various features of the program are assigned to user categories, not directly to individual users.

Security > Database

- **Convert To SQL Server Mode:** If you plan to use a SQL Server database and you haven't already done so, you should do so now. See the **SQL Server Mode Configuration** topic or the **Create New SQL Server Database** topic (in the NeoData Help document) for details. Once you have created your SQL Server database, you will need to set the relevant configuration options in the **Security > Database > SQL Server** section.

- **Microsoft Access Backup Strategy:** If you are planning to use an Access database, backing up the database frequently and reliably is **very** important. By default, NeoData will automatically back up the database every 6 hours but will issue a notice to the user each time before it does so. You should review this and related settings and make changes if appropriate. If you have another reliable backup strategy, you may wish to turn this feature off altogether. Please see the **Database Backups** section of the **After Installation** topic.

List Configuration

Most of the lists (used for drop-down lists for data entry fields) can be customized gradually with ongoing use of the system. However, the following lists should be edited as part of the initial setup:

- **Attending MD**

- **Cities**

- **CopiesTo**

- **County**

- **Hospital**

- **Insurance**

- **Obstetrician**

- **Pediatrician**

- **PreparedBy**

- **Service**

- **States**

For details on editing lists, see the **List Configuration** topic (in the NeoData Help document).

SQL Server Mode Configuration

NeoData can use either a **Microsoft Access** or a **Microsoft SQL Server** database. An Access database for use by NeoData is automatically installed as part of the Data Files component during installation, and NeoData runs in Access mode by default (i.e., no other configuration changes are required to use NeoData in Access mode). If the system is to be used with a **SQL Server** database, several more steps are required:

SQL Server Setup

Prior to creating the SQL Server database:

- Configure your server as appropriate. Confirm that your workstations can log into the server.

- If not already present, install **SQL Server 2008 or above** on your server. The defaults for the Code Page (1252 - ISO Character Set), Sort Order (dictionary order, case-insensitive), and Unicode settings are adequate for NeoData.

- If not already installed, install **MDAC 2.6** or later on the server. The MDAC installation file is included on the NeoData installation disk.

Database Conversion

To run NeoData in SQL Server Mode, you must create a SQL Server database using the Access database as a template, and you must set the necessary configuration options.

To create the SQL Server database:

- Start NeoData on any workstation in Access mode (the default); make sure that the Access database has been opened successfully. Select the **Configuration** option under the **Tools** menu. Enter the **Administration Password** (the default password is 'nicu4321').

- Open the **General Configuration Form** by clicking the **General** button on the **Configuration Panel**. Navigate to the **Security > Database > SQL Server** configuration section.

- Click the **Create New SQL Server Database** button. In **SQL Server Login** section of the **Create New Database** dialog, enter your SQL Server instance name and your system administrator login name (typically '**sa**') and password. Click the **Try Connection** button to verify that the program can log into SQL Server.

- In the **New Database** section, enter the requested information. Be sure the **Create A New Database And Copy Data** option button is selected. Enter the directories for the new database's data and log files (remember that these paths are relative to the server on which your SQL Server instance resides).

- If you are setting up a new NeoData installation, click the **Config Tables** button to select only the configuration tables from which to copy data. This will insure that your new database will be empty of any patient data. If you are converting an existing NeoData installation to SQL Server, click the **All** button to copy data from all tables.

- Click the **OK** button and follow the prompts. The program will create your new SQL Server database.

NeoData Configuration

To configure NeoData to use the new SQL Server database:

- In the **Security > Database > SQL Server** section of the **General Configuration Form**, turn on the **Start Up In SQL Server Mode** checkbox and enter the remaining connection settings.

- Enter the **Server Name** (the name of the SQL Server instance) and the **Database Name** (usually '**neodata**').

- Choose the **Authentication** method you wish to use (see the NeoData documentation). In most cases, the default setting (SQL Server Authentication - Common Login) is appropriate.

  > **Note**: If you select **SQL Server Authentication - Individual Login** or **SQL Server Authentication - Double Login**, you must turn on the **Require NeoData Login** option on the in the **Security > Logins** section and create the appropriate logins on the **Users** tab.
  >
  > **Note**: Whichever mode you choose for authentication the SQL Server user account(s) MUST be db_owner on the NeoData database

- Exit the General Configuration Form and close NeoData.

SQL Server Configuration

To configure NeoData to use the new SQL Server database:

- Open **SQL Server Management Studio** (SQL Server 2008 and above). Expand the **Security** tree structure and select **Logins**.

- The database conversion should have created a default login ('**neodatauser**') for use as the NeoData **System Login** and **Common Login**. If the login is present, verify that its **Properties** are set correctly; if not, create it using the following properties:

  Name: 'neodatauser'
  Password: 'jabberwocky'
  Authentication: SQL Server authentication
  Default Database: 'neodata' (or your database name)
  Default Language: English
  Server Roles: None
  User Mapping: 'neodata' (or your database name)
  Database Roles for 'neodata': Public and Db_Owner

- Expand the **Databases** tree structure and find the new database (you may need to refresh the display). Right-click on the new database name and select **Properties**. Under **Options**, review and if necessary change any settings based on your preferred approach to database maintenance and recovery.

- Expand the new database and select **Security** then **Users**. There should be a user for the default login. If not, create one by right-clicking on **Users** and selecting **New Database User**. Set both **Login Name** and **User Name** to the default login name. Turn on both the **Public** and **Db_owner** database roles.

- If necessary, create additional SQL Server logins as needed depending on the authentication method you plan to use (see below).

Final Steps

- Reopen NeoData and confirm that it can open the SQL Server database. The SQL Server instance name and the database name should appear in the Caption Bar of the main application form. Even if the application doesn't open the database successfully, you can return to the General Configuration Form to recheck your settings if necessary.

- Establish a reliable backup process for the new database; **SQL Server Agent** is a good option.

Launching NeoData

In most cases NeoData will be launched using a standard Windows **shortcut**, either from the Desktop or the Program Menu. A NeoData shortcut should be configured as follows:

- The **Target** property should point to the executable file (e.g., **C:\Nd4\NeoData.exe** or **F:\Apps\Nd4\NeoData.exe**) followed by any **command-line options** (see below). Note that the executables files may be installed on either a network drive or a local drive.

- The **Start In** property should point to the **Data File Directory** where the database file (Access databases only) and the configuration file (**NEODATA.CFG**) are located.

Command Line Options

The following **command line options** are available for starting the NeoData program. These are useful when using the Start Menu **Run** command and when creating or changing Program Menu or Desktop **shortcuts**:

- **/C:** Starts NeoData in **Clinic** mode (example: **C:\Nd\NeoData.exe /C**).

- **/B:** Starts NeoData in **BloodGas** mode (example: **C:\Nd\NeoData.exe /B**).

- **/NoDb:** Starts NeoData without opening the database.

- **/SQL:** Starts NeoData in **SQL Server** mode.

- **/MDB:** Starts NeoData in **Access** mode.

- **/Local:** The **Run Application Locally** feature is forced **on** (see below).

- **/NoLocal:** The **Run Application Locally** feature is forced **off** (see below).

- **STARTIN=xxxx:** Sets the **startup directory** (Data File Directory) to **xxxx.**

- **MDBFILE=xxxx/yyyy:** Sets the **startup directory** to **xxxx** and opens the **yyyy** Access database (if NeoData is started in Access mode).

- **SERVER=xxxx:** Sets the SQL Server **server name** to **xxxx** (SQL Server mode).

- **DATABASE=xxxx:** Sets the SQL Server **database name** to xxxx (SQL Server mode).

- **NETLIB=xxxx:** Sets the SQL Server **net library** to xxxx (SQL Server mode).

- **TIMEOUT=xxx:** Sets the **connection timeout** value to xxx (0-999) when attempting to connect to SQL Server (SQL Server mode).

Running NeoData From A Local Drive

The recommended method for configuring workstations is to install the executable files in a single, shared network location and to have each workstation run the application from that directory. This simplifies system management, particularly when installing program updates.

However, in rare cases performance may be improved if the application is run from a local directory, with each workstation having its own copy of the executable files. This can of course be accomplished by installing the application on each workstation and pointing the shortcut's **Target** property to the local installation, but this approach requires that any updates be installed separately on each workstation.

An alternative is to use the **Run Application Locally** feature. When this feature is enabled, the program's run-time behavior is as follows:

1. The program is launched using a shortcut whose **Target** property points to a shared network installation of the application.

2. At startup, the program checks to see if there is a copy of the executable files in a specified local directory on the workstation. If not, it creates the local directory (if necessary) and copies the files from the network to the local directory.

3. If the files are already present in the local directory, the program compares versions and updates the local files from the network if appropriate.

4. The first instance of the program closes and simultaneously launches a second instance using the executable files in the local directory.

This feature has the advantage of automatic installation and updating of the application's executable files while the user still benefits from any performance improvement that comes from running the application locally. The only disadvantage is the relatively brief startup delay as the first (network) instance of the application runs, closes, and launches the second (local) instance.

To enable the Run Application Locally feature:

You can enable the Run Application Locally feature in one of two ways:

- Turn on the **Run Application From Local Directory** option in the **Security >> Administration** section of the **General Configuration Form**. This will enable the feature for all workstations. The local directory name is specified by the **Local Application Directory** configuration setting.

- Add the **/Local** command line option to the **Target** property of a workstation's shortcut. This will enable the feature just for that workstation.

To disable the Run Application Locally feature for a workstation.

- Add the **/NoLocal** command line option to the **Target** property of the workstation's shortcut. This will disable the feature even if the **Run Application From Local Directory** configuration option is turned on.

Administration

Updates

All updates for the licensed product are included in the maintenance agreement which is part of the software license; the current build of the application is always available on the Isoprime web site. All updates are cumulative; updates can be skipped if desired. To apply an update:

1. Go to the Isoprime web site (**www.isoprime.com**) and click on the **Program Updates** in the NeoData section of the site.

2. Download and run the **Update Installation Program** (ND6UPDATE.EXE). Follow the instructions.

3. On the **Select A Function** dialog, select the **Update existing NeoData Version program files** option.

4. On the **Select Destination Directory** dialog, click **Browse** and enter the path to your existing executable file (NEODATA.EXE). Note that if you have a working NeoData shortcut, its **Target** property will show you the correct executable file path. When you return from the **Browse** dialog, be sure that the **Program File Directory** is entered correctly.

5. Before copying files, the installation program will ask you for a password. If you don't have the current password, contact Isoprime.

Users or technical staff at the customer site can discuss potential application software changes with Isoprime at any time by email or phone. Such requests are prioritized based on the magnitude of the change and the general applicability of the feature to other sites, among other considerations.

Database Backups

You will obviously want to implement a reliable backup strategy to insure that your database is protected at all times. There are many ways to do this, although the approach will depend on the type of database you have in place.

For **SQL Server databases**, corruption of the database and loss of data are very rare events due to the inherent robustness of the database system. Given this, a single-tier strategy involving just a nightly server backup may be adequate. Nonetheless, more frequent backups (for instance, using SQL Server Agent) may be desirable to minimize the amount of work lost should something happen to the database.

Access databases are much more prone to corruption and possible data loss. While corrupted databases are often repairable, this is by no means always the case. Therefore we recommend a very aggressive backup strategy, preferably with two levels of protection. Even if your database resides on a server which is backed up nightly, we **strongly** recommend that regular backups of the database and configuration files themselves be made on a more frequent basis. Ideally, the clinical users should be involved in deciding how often this occurs, since they are the ones at risk for having to re-enter many hours of data.

NeoData includes a built-in backup function (accessible via the **Tools | Database Administration** submenu) and a reminder system (which is on by default) to remind users to create backups at a definable interval (6 hours by default, although you may wish to be reminded more frequently). You can control various aspects of the reminder logic via the **Security > Database > MS Access** section of the **General Configuration Form**.

Documentation

Full NeoData documentation is provided as an on-line help file; a revised version of the documentation is installed with each major or minor software update. A printed **Quick Start Guide** is also supplied with the system software upon purchase of the software license; a current version of this guide is available as a Microsoft Word document at any time upon request.

Training

A tutorial is included in the **Quick Start Guide** and is also available from within the application (see the **Getting Started Tutorial** option in the **Help** menu). Full documentation of the system features is included in the on-line help file. Assistance with user questions by phone or email is included in the maintenance agreement which is part of the software license.

Most sites begin operations with the system without on-site training, although this is available by arrangement for a daily fee.

Because NICU's often bring in new clinical staff regularly, training for the NeoData system tends to be a continuous process. The process usually works best when one or two users are designated as team resource staff for training, problem-solving, configuration, and administration tasks. Isoprime works closely with such staff by phone and email to assist them in these functions.

We recommend that on-site visits, when necessary, be made after the system has been in use for at least a limited time. Visits tend to be more helpful to users when they are already familiar with the basic concepts and operation of the program and the available time can be used to focus on their specific questions, needs, and problems.

Technical Support

Technical support is included in the maintenance agreement which is part of each software license. Isoprime provides support for the application software only by phone and email.

Users should report a problem by email or by calling the support number (630-737-0964). If a support person is not available, the user should leave a voicemail message. A support person will contact the user as soon as possible.

Other Information

Creating a Test Environment

Isoprime strongly recommends that each customer maintain a test environment separate from their NeoData production environment. This can be used to safely test major upgrades and new features, and can also be used for training without affecting production data. Typically a test environment is created by starting with a copy of the production data. This ensures that the configurations are the same and that realistic data is available.

These instructions are for creating a NeoData test environment on the same server as the production environment. These same steps can be used to create a test environment on a different server. You will just need to change the UNC paths. Similarly, if you would like to use a test SQL Server instead of the production server to host your test database, you will just change the server name at the appropriate time.

Note: These instructions assume that you are using a shared NeoData executable on a server, NOT installing NeoData locally on all the workstations. This is typically the case. If your organization installs NeoData locally instead of running from a shared instance on the server please contact Isoprime technical support. Visit http://www.isoprime.com/contact.htm for contact information.

SQL Server

Create a test database

Option 1: Manually create a test database

1. Using a SQL Server utility (Enterprise Manager, Management Studio, etc) take a backup of your production database.

2. Using the SQL Server utility restore the backup to your server as a different name, such as NeoDataTest

3. Run the following SQL Command against all the database to realign the SQL user account from the transferred database to the new SQL Account on the new server: EXEC sp_change_users_login 'Auto_Fix','neodatauser'   where 'neodatauser' is the SQL account you use for NeoData (see Step 3).

Option 2: Use NeoData to create a test database

1. Log in to your production NeoData

2. Open the **Settings > Configuration** menu option (password required)

3. Navigate to **Security->Database->SQL Server** and click **Create New Database**

4. On the screen that appears:

a. Enter the SQL Server instance name and the sa password

b. Choose **Create A New Database and Copy Data**

c. Enter a **New Database Name** and a path to store the data (Note that the path is relative to the SQL Server, not the Workstation you are using). If you are going to use a different SQL Server also enter the server name.

d. If you would like to copy over all the current patient data click the **All** button next to the **Select Tables From Which to Copy Data** list; if you would only like to copy the configuration data and lists click the **Config Tables** button.

e. Click **OK** to create the new database. Creating the new database does NOT alter the settings for NeoData, so YOUR WORKSTATION STILL POINTS TO PRODUCTION DATA, NOT TEST DATA. We will create a Test NeoData instance in the next step.

Create a test instance of the NeoData software

1. Locate the two NeoData folders – the product (location of the executable NeoData software) folder and the data folder. In many cases these are the same folder. There are <u>two ways</u> to locate these folder(s):

    a. If you have a NeoData shortcut you can right-click it and go to properties. The product folder is in the "Target:" line, and the data folder is the "Start in:" folder. You should make note of both of these folders (again, these may be the same folder)

    b. Search your server for NeoData.exe – the folder that contains this file is the product folder. Search for neodata.cfg – the folder that contains this file is the data folder.

2. If you have a single folder (your product and data in folder are the same) proceed as follows:

    a. Copy the folder to another folder on the server (we recommend including "test" in the folder name). For example, copy \\isoprimeserver\nd4 to \\isoprimeserver\nd4test

    b. Proceed to Step 4

3. If you have two folders (your product and data folders are different) proceed as follows

    a. Copy the product folder (location of the executable NeoData software) to another folder on the server (we recommend including "test" in the folder name). For example, you would copy \\isoprimeserver\nd4 to \\isoprimeserver\nd4test

    b. Copy the data folder to another folder on the server (again, we recommend including "test" in the folder name).

    c. Proceed to Step 4

4. You have now created a test NeoData installation, and you should create a shortcut to the new test instance

    a. Create a shortcut to the test NeoData.exe file that is located in the newly created folder (NOT the original product folder)

    b. Set the "Start in" property of the shortcut to point to the newly created data folder (NOT the original production data folder).

5. The test instance still points to the production data. We now need to point the NeoData test version to the test database.

    a. Launch your newly created shortcut to the test system. At this point you are still pointing to the production data

    b. Open the **Tools > Configuration** menu option (password required)

    c. Click on **General** on the **Configuration Panel** to go to the **General Configuration**

    d. Navigate to **Security->Database->SQL Server** and update the Server Instance and Database Name to point to the test database

6. Update other configuration options to reflect the test server name if you are using the related features. See the following help topics for specifics:

    a. Protocol Configuration

    b. Facsimile Signatures of the Security Configuration

    c. Logo Files in Print Layout Document Configuration

    d. Electronic Document Configuration

    e. Patient Education Resources Configuration

    f. Directories specified on the General Configuration **System Settings** panel

g. Custom dictionary file name in the General Configuration **Spell Checking** panel

h. CCD General Configuration

i. Secure File Transfer Form

j. Third Party HL7 Configuration

k. Nursing Module Link Configuration

Microsoft Access

1. Locate the two NeoData folders – the product (location of the executable NeoData software) folder and the data folder. In many cases these are the same folder. There are two ways to locate these folder(s):

a. If you have a NeoData shortcut you can right-click it and go to properties. The product folder is in the "Target:" line, and the data folder is the "Start in:" folder. You should make note of both of these folders (again, these may be the same folder)

b. Search your server for NeoData.exe – the folder that contains this file is the product folder. Search for neodata.cfg – the folder that contains this file is the data folder.

2. If you have a single folder (your product and data folder are the same) proceed as follows:

a. Copy the folder to another folder on the server (we recommend including "test" in the folder name). For example, copy \\isoprimeserver\nd4 to \\isoprimeserver\nd4test

b. Proceed to Step 4

3. If you have two folders (your product and data folder are different) proceed as follows

a. Copy the product folder (location of the executable NeoData software) to another folder on the server (we recommend including "test" in the folder name). For example, given the above screenshot you should copy \\isoprimeserver\nd4 to \\isoprimeserver\nd4test

b. Copy the data folder to another folder on the server (again, we recommend including "test" in the folder name).

c. Proceed to Step 4

4. You have now created a test NeoData installation, and you should create a shortcut to the new test instance

a. Create a shortcut to the test NeoData.exe file that is located in the newly created folder (NOT the original product folder)

b. Set the "Start in" property of the shortcut to point to the newly created data folder (NOT the original production data folder).

5. Update other configuration options to reflect the test server name if you are using the related features. See the following help topics for specifics:

a. Protocol Configuration

b. Facsimile Signatures of the Security Configuration

c. Logo Files in Print Layout Document Configuration

d. Electronic Document Configuration

e. Patient Education Resources Configuration

f. Directories specified on the General Configuration **System Settings** panel

g. Custom dictionary file name in the General Configuration **Spell Checking** panel

h. CCD General Configuration

i. Secure File Transfer Form

j. Third Party HL7 Configuration

k. Nursing Module Link Configuration

Migrating NeoData To a New Server

These instructions are intended for an IT professional familiar with SQL Server databases and Microsoft Server administration.   Specific questions can be directed to Isoprime support staff.   In addition, Isoprime offers a billable service whereby we assist through the duration of the migration project.

The following can be used to migrate one or more NeoData SQL Server databases along with any related NeoData software or NeoData HL7 Interface software to a new server.

Database

NOTE: This section only applies to SQL Server databases. Skip this section if you are using a Microsoft Access database.

Manual Move

At a high level this involves using a SQL Server utility, such as Enterprise Manager, to create a backup of the database on the old server and restore the database from that backup on the new server.

Isoprime does not typically recommend this method because it creates additional work (and thus additional risk) for the DBA (database administrator). Once the database is on the new server the DBA will need to manually create the SQL User account that NeoData uses and ensure it is db_owner for the NeoData database. Also, there are some issues inherent to moving a SQL Database from one server to another based on the way Microsoft handles SQL Server User Accounts (and internal ids and such) which need to be accommodated

Specifically, the steps are:

1. Create a backup of the database and (optionally) zip the database backup

2. Move the backup to the new server and unzip if needed

3. Create the neodatauser SQL Account (or whatever SQL account you use for NeoData) on the new server

4. Restore the database

5. Run the following SQL Command against all the database to realign the SQL user account from the transferred database to the new SQL Account on the new server: EXEC sp_change_users_login 'Auto_Fix','neodatauser'   where 'neodatauser' is the SQL account you use for NeoData (see Step 3).

Use NeoData Utility

1. Log in to NeoData

2. Open the **Tools > Configuration** menu option (password required)

3. Click on **General** on the **Configuration Panel** to go to the **General Configuration**

4. Navigate to **Security->Database->SQL Server** and click **Create New SQL Server Database**

5. On the screen that appears:

a. Enter the SQL Server instance name and the sa password for the new database

b. Choose **Create A New Database and Copy Data**

c. Enter a **New Database Name** and a path to store the data (Note that the path is relative to the SQL Server, not the Workstation you are using).

d. Click the **All** button next to the **Select Tables From Which to Copy Data** list to copy all patient data

e. Click **OK** to create the new database.

NeoData

1. Locate the two NeoData folders – the product (location of the executable NeoData software) folder and the data folder. In many cases these are the same folder. There are two ways to locate these folder(s):

a. If you have a NeoData shortcut you can right-click it and go to properties. The product folder is in the "Target:" line, and the data folder is the "Start in:" folder. You should make note of both of these folders (again, these may be the same folder)

b. Search your server for NeoData.exe – the folder that contains this file is the product folder. Search for neodata.cfg – the folder that contains this file is the data folder.

2. If you have a single folder (your product and data folder are the same) proceed as follows:

a. Copy the folder to the new server. For example, copy \\oldserver\nd4 to \\newserver\nd4test

b. Proceed to Step 4

3. If you have two folders (your product and data folder are different) proceed as follows

a. Copy the product folder (location of the executable NeoData software) to the new server. For example, copy \\oldserver\nd4 to \\newserver\nd4test

b. Copy the data folder to the new server

c. Proceed to Step 4

4. You have now created a new NeoData installation, and you should create a shortcut to the new instance

a. Create a shortcut to the test NeoData.exe file that is located in the newly created folder (NOT the original production folder)

b. Set the "Start in" property of the shortcut to point to the newly created data folder (NOT the original production data folder).

5. If you are using SQL Server

a. Launch your newly created shortcut that points to the new system.

b. Open the **Tools > Configuration** menu option (password required)

c. Click on **General** on the **Configuration Panel** to go to the **General Configuration**

d. Navigate to **Security->Database->SQL Server** and update the Server Instance and Database Name to point to the new database

6. Update other configuration options to reflect the new server name if you are using the related features. See the following help topics for specifics:

a. Protocol Configuration

b. Facsimile Signatures of the Security Configuration

c. Logo Files in Print Layout Document Configuration

d. Electronic Document Configuration

e. Patient Education Resources Configuration

f. Directories specified on the General Configuration **System Settings** panel

g. Custom dictionary file name in the General Configuration **Spell Checking** panel

h. CCD General Configuration

i. Secure File Transfer Form

j. Third Party HL7 Configuration

k. Nursing Module Link Configuration

7. If the old server is not being shut down, we recommend that you remove the sharing permission ("unshare") from the folders on the old server to ensure that none of the client PCs can point to the old location.

NeoData HL7 Interface

On the Old Server

1. Take screenshots of the Configure Receiver Form and the Configure Transmitter Form for any/all transmitters.

2. Take screenshots of the Interface Hospital Form for all hospitals on the dropdown list.

3. Use the Select Map Form to export the ADT Map, the Labs Map, and (if applicable) the Documents Outbound Map.   ??? translate tables???

On the New Server

NOTE: The server MUST have the .NET Framework 3.5 (it must have this specific version – newer versions do not contain this version – it must be installed side-by-side with newer versions if you have them)

1. Download and install the full interface from:

http://www.isoprime.com/neodata/documents/ndupdtmp/interfacev3setup.exe

2. Launch th eNeoData Interface Manager Form and click **Add Hospital** - enter the license key information from your screenshots of the old server. See the NeoData Interface Manager Form help topic for the default password to use the NeoData Interface Manager Form.   This password will only work if your site has not changed the password.

3. Click **Configure** under **Receiver** and update the information from screenshots on the old server; click **Configure Maps** and import the ADT and Lab maps from the old server

4. If you are using the documents interface you will need to go to **Manage Transmitters** and create a documents oubound interface. Then enter the configuration information from your screenshots

b. Click **Configure Map** and import the documents map from the old server

Database Queries

NeoData uses either a **Microsoft Access** or **Microsoft SQL Server** database. In addition to the built-in **query and reporting** capabilities, the database may be queried outside the NeoData program using Access or most other DBMS programs. The ease with which you can manipulate the database using programs other than Access will vary from

product to product; the following discussion is based on the assumption that you are using Access. Access in particular has querying and reporting features which are both powerful and easy to use.

If you plan to analyze data, you should expect to become familiar with at least one of these programs, with the NeoData **database structure**, and with the process of performing data queries. This topic includes various notes to help you get started.

Database Structure

The NeoData database uses a **relational data model**. Information is organized into **columns** (**fields**) and **rows** (**data records** or table entries) within a set of **data tables**. Each table stores information about a particular **object**, such as a patient, a diagnosis, or a group of related lab results. The structure of these tables can be examined from within Access.

The NeoData data file includes over 130 tables, which can be separated into several groups:

- **List Tables:** A number of tables are used to store **lists** used by **list fields** within NeoData. These **lists tables** have a name ending in '**List**', such as:

  MedicationList
  ProcedureList
  DiagnosisList
  FormulaList

  Note: Don't confuse the list's **table name** (i.e., DiagnosisList) with the list name itself (i.e., Diagnosis).

- **Program Tables:** Other tables are used by NeoData to manage certain program functions, such as:

  - **Addresses:** Stores the names and addresses used in the **Address List** feature.

  - **CPTCodes:** Stores the CPT Codes used by the **Daily Charges** feature.

  - **CurrentPatients:** Stores the list of currently admitted patients as displayed in the **Current Patients List** on the **Select Patient Form**.

  - **DBInfo:** Used to maintain compatibility between the current version of NeoData and the database file itself. It should not be altered.

  - **FieldParams:** Stores a list of all data entry fields used in the NeoData **Data Entry Forms** and the associated field settings, such as field name, data type, minimum and maximum values, etc.

  - **FormParams, GroupParams, and GroupFields:** Store information about the **NeoData Data Entry Forms**.

  - **Logins:** Dynamically stores information about which workstations are logged into the database at any given moment.

  - **StandardLists:** Stores all Standard Lists in a single table.

- **Non-NICU BloodGas Tables:** 3 tables are used only by the **BloodGas Module** for cord blood gas results, blood gas results for patients outside the NICU, and quality control specimens:

  CordGases
  OtherGases
  Controls

- **Clinic Tables:** 3 tables are used only by the **Follow-up Clinic Module**:

  CPatients
  CVisits
  CTests

- **Lab and ADT Interface Tables:** Certain tables are used only by the **Lab/Admission Interface** feature, such as:

  LabData
  LabNames
  ADTData

- **Patient Data Tables:** The remainder of the tables store actual patient-related information, such as:

  Admissions
  BloodGases
  Cultures
  DailyInfo
  Diagnoses
  IVFluids
  LabsHeme
  Medications
  Patients
  Pregnancies
  Procedures
  Radiology

Table Relationships

In a relational database structure, tables whose information is related are **linked by a common field**. This relational approach minimizes the duplication of data in a database. However, some mechanism must be provided to allow matching of related records to each other when necessary; this is accomplished by using a common field in both tables. These common fields must store the same type of information and are usually given the same name. Relationships may be **one to one** (each entry in Table A may be linked to no more than one entry in Table B) or **one to many** (each entry in Table A may be linked to more than one entry in Table B).

For instance, in NeoData a single patient may have multiple admissions. Consequently, information relating to the **patient** (name, birthdate, etc.) is stored in a single record in the **Patients** table, while information related to that patient's **admissions** (admission date, admission hospital, discharge date, etc.) is stored in one or more records in the **Admissions** table. This approach avoids the duplication of storing the patient information in **each** of the admission data records. Matching of each admission data record with the appropriate patient data record is accomplished by using a numeric *PatientID* field in both tables. In the **Patients** table, a *PatientID* entry is **unique**; in the **Admissions** table, more than one record may have the same value entered in the **PatientID** field, since more than one admission may be related to the same patient. This is an example of a **one to many** relationship.

Note that the *PatientID* field is an example of a data table field which does not appear on any of NeoData's **Data Entry Forms**. This field is invisible to NeoData users, but is **critical** to the proper functioning of the program. This field **will** be visible from within Access, however. Care must be taken not to change the entry in such a field unless you know precisely what you are doing.

NeoData uses a fairly large number of relationships between its data tables. Many tables (the lab tables, the **BloodGases** table, etc.) are linked both to the **Patients** table (using the *PatientID* field) and the **Admissions** table (using the *AdmitID* field).

Examples of relationships:

- The **Pregnancies** table is linked to the **Patients** table via the *PregnancyID* field.

- The **Patients** table is linked to all other patient data tables (except the Pregnancies table) via the *PatientID* field.

- The **Admissions** is linked to most of the other patient data tables via the *AdmitID* field.

- The **Diagnoses** table is linked to the **Medications** and **Procedures** tables via the *DiagnosisID* field.

Note that there is no **direct** relationship between the **Pregnancies** table and the **Admissions** table; to link these tables requires a two-step process linking **Pregnancies** to **Patients** and **Patients** to **Admissions**.

Patient Identifiers and Record Identifiers

It is important to keep in mind the distinction between the **patient identifiers** (the *Admission ID* and *Medical Record ID* data entry fields, found on the **New Admission** and **Change Admission Info** forms), and the hidden **record identifier** fields used to maintain linkages between records in the database tables (*PregnancyID*, *PatientID*, *AdmitID*). Although referred to as *Admission ID* and *Medical Record ID* on the data screens and printouts, these patient identifiers have **different names** in the actual data tables. The Admission ID field is named *HospitalID* in the **Admissions** table, and the Medical Record ID field is named *MedRecNum* in the **Patients** table.

Although you could possibly use the patient identifiers to link tables in queries, the record identifiers are specifically designed for this purpose. For instance, the record identifiers are **always** present in each record, whereas the Medical Record ID can be used as an optional data entry field, and even the Admission ID might conceivably be missing.

Storing Queries

Like most DBMS programs, Access allows you to store queries, once created, for later reuse. Since Access (**unlike** many other DBMS programs) stores all objects associated with the database (including tables and queries) in a single file, this raises the problem of **where** to store queries. There are basically two options:

- The queries can be stored in the same file as the data tables (**NEODATA.MDB**).

- The queries can be stored in a separate database file (e.g., **NDQUERY.MDB**), and the data tables from **NEODATA.MDB** can be **attached** to the query database file. Attaching a table from one database file to another allows the second file to manipulate the data in the attached tables without actually storing a copy of the tables in the second file. Refer to the Microsoft Access documentation for more information about attaching tables.

The issues to consider in making this decision are as follows:

- It is safer to create, test, and run queries on a **copy** of the data file rather than on the working data file itself, especially in a multiuser environment where others may be accessing the database simultaneously. It is also desirable when possible to perform these operations on a different computer than the one acting as file server.

- However, if you create and store queries in the copied file, the queries will be overwritten (i.e., lost) the next time you make a copy of the working data file.

For these reasons, you will probably find it preferable to store queries in a separate file and attach the necessary data tables from **NEODATA.MDB**. The attached tables will remain attached to the query database file between sessions, so you will have to perform this step only once for each table.

Creating Queries

Based on the above considerations, the following is one possible sequence for setting up reusable queries:

- Create a copy of the data file (**NEODATA.MDB**) and place it in an appropriate working **subdirectory** on your file server or local workstation;

- Open **Microsoft Access**;

- Create a new database with a name such as **NDQUERY.MDB** and place it in the same working directory as the data file copy (**NEODATA.MDB**);

- Attach whichever tables you wish to query from the data file copy to the query database;

- Create one or more queries using the **Design View** or **SQL View**;

- Save each query in the query database;

- Test and run each query as necessary.

Microsoft Access

When running in Access mode, **NeoData** uses the same **database engine** and the same **data file formats** as Microsoft Access, allowing NeoData files to be used by Access as needed for data queries, file conversions, or other operations. However, the existence of multiple versions of Access can complicate the use of Access with NeoData files. If you plan to use Access with NeoData files, please review the following:

- **NeoData Version 6** can work directly with the 2000 file format and above.

- If you are **upgrading** from a previous version of NeoData, your database file **may** be in the Access 97 format. Isoprime recommends you convert your data file to the Access 2000 format.

- If you are setting up a **new installation** of NeoData, you will automatically be using the Access 2000 file format.

Note: NeoData cannot convert a database file from one file format to another. This operation must be performed using Access.

HL7 Interface Options

Introduction

NeoData allows admission data and certain labs to be loaded from an external system into a staging area within NeoData (an Access database). This data can then be reviewed within NeoData and, if accepted by the clinician, is loaded into the NeoData patient database.

This staging area can be loaded by programs developed internally at the hospital. However, some customers use a third-party program called DT7. This program was written specifically for NeoData users to provide a means for receiving HL7 transactions from external systems and loading them into the NeoData staging area. For more information visit www.dt7.com.

In Version 5 of NeoData, Isoprime has introduced its own tool for sending and receiving HL7 transactions. This allows NeoData users to deal with one vendor for NeoData and the related interfaces, and allows Isoprime to expand the interface capabilities.

DT7 or Other Third-Party Modules

Customers may choose to remain with the DT7 interface. New customers may also choose to go with that solution. The NeoData HL7 module is offered as another option. Also, customers will continue to have the option to load the NeoData staging area themselves if they have the capability.

The tools and structure required to facilitate our customers' use of DT7 (or their own HL7 utility) will remain in place until no longer needed. There will be no enhancement to those capabilities. All future development work will be focused on the NeoData HL7 module.

Documentation on the NeoData features that support the use of DT7 can be found in the **DT7 or Other Third-Party Modules** chapter of the online Help, the first page of which describes the Lab/Admission Interface.

NeoData HL7 Module

The **NeoData HL7 Module** provides a means to receive HL7 admission (ADT), birth data (ORU) and lab messages (ORU) from an external system, validate and parse those messages, and load them into NeoData so they can be reviewed and accepted by an NICU clinician. Once accepted, the data can be included in notes as if entered by a clinician.

It also supports the transmission of all NeoData note types via HL7 MDM messages. Users can select HL7 MDM as a new electronic document type and have their notes automatically transmitted to an external system using the same triggers and procedures as those available for other electronic document formats. Data on patient diagnoses and immunizations can also be transmitted via HL7.

This module is provided as an alternative to DT7 or other third-party products. Tips for moving from DT7 to the HL7 Module can be found in the topic Migrating from DT7 to NeoData HL7 of the online Help.

The **NeoData HL7 Module** is sold and supported by Isoprime as an add-on module for NeoData. Please visit our web site at www.isoprime.com for pricing information or contact the Sales department at 630-737-0963.

Data File Corruptions

Note: The information in this section applies primarily to **Microsoft Access** database files, such as those used by NeoData in Access mode. **SQL Server** databases (used by NeoData in SQL Server mode) rarely become corrupted, and when they do the causes and solutions are generally different than for Access databases.

File Corruptions

Under certain circumstances, it is possible for the NeoData database file to become **corrupted**. File corruptions can be of varying degrees of severity ranging from large amounts of missing or damaged data to a single damaged record in a single table. In some cases, the file cannot be opened; in others, the file can be opened but displaying certain information causes problems.

A corrupted data file can produce the following symptoms:

- When NeoData is started, you may see an error message explicitly stating that the data file is corrupted. NeoData will continue running, but the data file will not be opened.

- When NeoData is started, you may see some other error message; again, NeoData will continue running but the data file will not be opened.

- When certain data screens are displayed for certain patients, the program hangs or exits (with or without an error message).

- Information may appear garbled in certain screens for certain patients (often in one specific screen for one specific patient).

When a file corruption is suspected, it is very important for all users to exit the program and for the data file to be repaired immediately. File corruptions may result in lost data, and the longer a corrupted file is used, the more data may be lost.

Repairing a Corrupted Database File

There are 3 ways to repair NeoData data files:

- **Using NeoData:** NeoData's built-in **Compact/Repair Database** function (found under the **Tools | Database Administration** menu) can repair most corruptions, and can be used even if the program can't open the data file. It is a good idea to run this procedure regularly even if you are not having problems.

- **Using Access:** Microsoft Access can sometimes repair a data file that NeoData can't, but you must use the version of Access (97 or 2000) that matches your data file format. The database cannot be open to use this feature.

- **Using JetComp:** The **JetComp** utility from Microsoft (**JETCOMP.EXE**) can be found in your NeoData directory and can repair some corruptions which neither NeoData nor Access can fix. NeoData should not be running when JetComp is used.

If you have any problems repairing a corrupted data file, please contact Isoprime for assistance.

Possible Causes of File Corruptions

- **NeoData Bug Causing Corruptions Directly:** It is possible for errors in the NeoData program to cause corruptions directly. However, this is fairly unlikely, since **all** database access from NeoData is handled by the **Jet** (Microsoft Access) **database engine** code files. These are widely-used Microsoft files, are reasonably stable, and change only infrequently.

- **NeoData Bug Causing Abnormal Program Termination:** If a NeoData problem causes the program to exit abnormally for any reason, there is always some small risk of a file corruption. NeoData maintains a log of all errors that it recognizes; this log is stored in a file named **NEODATA.LOG** in the NeoData directory. In some cases, the log file can help Isoprime identify problems which might be causing file corruptions.

- **File Corruption Which Has Not Been Fully Repaired:** One possible explanation of **recurrent** file corruptions is that a initial corruption has not been fully repaired. It appears that in rare cases it is possible for the database repair tools provided by Microsoft to fail to completely repair a corrupted data file even though the repair process seems to be successful. Isoprime can help identify such situations.

- **Corrupted Code Files:** A corrupted code file (the NeoData program file itself or any DLL or OCX code file used by NeoData) could potentially cause file corruptions. If the NeoData code file is corrupted and is installed on a server, then problems might appear on any workstation. However, DLL or OCX files are installed on the workstation's local hard drive; if one of these files is corrupted it would lead to problems only on that machine. Reinstalling the program might eliminate such a problem.

- **Abnormal Program or Workstation Shutdown:** Users should **always** exit NeoData before exiting Windows, and they should **always** exit Windows before shutting off the workstation. Turning off the workstation while NeoData is running presents a definite risk of file corruption.

- **AC Power Issues:** In some institutions, brief power dropouts are common events. In some cases even the short time required for the hospital's emergency power to come on-line is enough to cause a workstation to reboot. Isoprime strongly recommends that all workstations be connected to a **UPS** (uninterruptable power supply), which will provide temporary battery power to the workstation.

- **Viruses:** A virus which has infected a workstation could theoretically cause a file corruption, either directly or by causing other problems which make the program or Windows exit abnormally. However, this is probably a **very** unlikely cause of file corruptions under normal circumstances.

- **Running Other Software:** The more often a NeoData workstation is used to run other software, the more opportunity there may be for problems to occur which could corrupt files.

Preventing Corruptions

The following is a list of measures which may help reduce the likelihood of data file corruptions. Not all of these may be practical at your institution, but they are presented for consideration:

- **Upgrade** your version of NeoData regularly to take advantage of the latest bug fixes.

- **Report** all apparent software problems to Isoprime so that bugs can be identified and fixed.

- **Train** all users in the normal procedures for exiting NeoData and Windows and for turning off workstations.

- **Connect** each workstation to a **UPS** (uninterruptable power supply).

- **Scan** workstations regularly for viruses.

- **Avoid** running other software when NeoData is running.

- **Compact and Repair** your data files regularly.

- **Fix** any suspected problems promptly.

Technical Support

Isoprime Web Site

Technical support is included in the maintenance agreement which is part of each software license. Isoprime provides support for the software by phone and email.

Users should report a problem by email or by calling the support phone number. If the support person is not available, the user should leave a voicemail message. If not immediately available, the support person will contact the user as soon as possible.

For further technical support information and for program updates, please visit our Web site at:

http://www.isoprime.com

Contacting Isoprime

If you are having problems with the installation or operation of the NeoData system or if you have questions about the use of the program, you may contact Isoprime at:

Address:	Isoprime Corporation
	505 Warrenville Road, Suite 104
	Lisle, IL 60532
Support Phone:	630-737-0964
Office Fax:	630-737-0971
Email:	isosupport@isoprime.com

Copyright © 1995-2018 Isoprime Corporation.

Index